TÉCNICO LISBOA

Predicting the Probability of Collision of a Satellite with Space Debris: A Bayesian Machine Learning Approach

João Francisco Simões Catulo

Thesis to obtain the Master of Science Degree in

Aerospace Engineering

Supervisors: Prof. Cláudia Alexandra Magalhães Soares
Prof. Francisco André Corrêa Alegria

Examination Committee

Chairperson: Prof. José Fernando Alves da Silva
Supervisor: Prof. Cláudia Alexandra Magalhães Soares
Members of the Committee: Prof. Mário Alexandre Teles de Figueiredo
Eng. Marta Isabel Vaz Guimarães Pinto de Melo

December 2022

Acknowledgments

First of all, I would like to express my sincere gratitude to my supervisors Professor Cláudia Soares and Professor Francisco Alegria for all the ideas, patience, and guidance throughout this thesis. Without your support, this work would not have been possible.

I would also like to give a very special thanks to the engineer Marta Guimarães, not only for your support, effort, patience, dedication, and domain knowledge but also for the strong attitude and positive mindset that you have shown when faced with difficult challenges and obstacles. It is truly an inspiration that I will carry with me throughout my professional path.

I would like to extend my gratitude to Neuraspace, for providing the data, and to everyone that works there, for the sympathy and availability during this project.

I want to thank all my friends for being with me throughout my academic path and for making the experience of being a student unforgettable.

Finally, my deepest thanks to my family, parents, sisters, and girlfriend for sharing my stress and for always believing in me. In particular, a very special thanks to my grandparents for your love, support, wisdom, and patience in taking care of me. You are truly an example that I will always follow and this thesis marks one great achievement (of many more to come!) that I want to share with you. I hope this work makes you proud!

Declaration

I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.

Resumo

Os operadores de satélites necessitam cada vez mais de considerar procedimentos de prevenção de colisões entre satélites e detritos espaciais. Atualmente, este processo baseia-se na análise humana de avisos de colisões, o que poderá ser uma tarefa impossível no futuro, devido ao grande aumento do número de objetos no espaço. Em 2019, a Agência Espacial Europeia lançou uma competição de aprendizagem automático para a previsão do risco de colisões, cujos resultados mostraram que as previsões simplistas são adequadas para este problema, sugerindo que os avisos de colisão podem seguir a propriedade de Markov. Este trabalho investiga essa teoria testando a implementação de modelos ocultos de Markov, usando duas abordagens: na primeira, a sequência do risco contido nos avisos de colisão é diretamente modelada e prevista; na segunda, as incertezas associadas à posição são previstas e o risco é calculado a partir das previsões. Neste trabalho, os modelos são treinados usando inferência Bayesiana, de modo a se inferir a distribuição conjunta dos parâmetros dos modelos implementados em vez de se obter estimativas pontuais. Isto permite a criação de modelos probabilísticos que podem incorporar conhecimentos/crenças prévias sobre o problema e fornecer intervalos de previsão. Apesar da segunda abordagem produzir resultados piores que o modelo base, a primeira abordagem supera a solução simplista, mesmo tendo sido analisada apenas uma característica do conjunto de dados, fortalecendo a ideia de que os avisos de colisão podem seguir a propriedade de Markov e indicando que este é um método promissor que deve ser devidamente explorado.

Palavras-chave: modelos ocultos de Markov, inferência Bayesiana, estimação do risco de colisão, aprendizagem automática

Abstract

Collision avoidance procedures have become essential to all satellite operators. Current procedures rely on the analysis of multiple collision warnings by human analysts, but, with the continuous growth of the space population, this manual approach may be an unfeasible task in the future. In 2019, the European Space Agency launched a machine learning competition for the collision risk prediction problem and the results showed that the naive forecast (considering the risk value at cut-off day) is a strong predictor for this problem, suggesting that the collision warnings may follow the Markov property. This work investigates this theory by benchmarking the use of hidden Markov models, using two different approaches. Firstly, the time series of the risk contained in the conjunction data messages is directly modeled and predicted and, in the second approach, the position uncertainties are predicted and the risk is computed from the predictions. To infer a joint distribution for the parameters of each implemented model instead of obtaining point-wise estimates, Bayesian inference is used, allowing the development of probabilistic models that can incorporate prior knowledge/beliefs about the problem and can provide prediction intervals. Although the second approach yields poor results, the first approach outperforms the baseline solution, despite the fact that only one feature of the dataset was analyzed, which further adds to the idea that the probability of collision may follow the Markov property and suggests that this is a powerful method that should be further explored.

Keywords: hidden Markov models, Bayesian inference, collision risk estimation, machine learning

Contents

List of Tables

List of Figures

Nomenclature

Greek symbols

$\boldsymbol{\mu}$	Set of mean values of the emission distributions of the Hidden Markov Models.
$\boldsymbol{\pi}$	Initial distribution of the states of the underlying Markov Chain in Hidden Markov Models.
$\boldsymbol{\sigma}$	Set of mean values of standard deviations of the Hidden Markov Models.
δ	Uncertainty in position vector of object.
$\mathcal{IG}$	Inverse Gamma distribution.
$\mathcal{M}_K(a, b)$	Set of K evenly spaced numbers between a and b.
$\mathcal{N}$	Truncated Normal distribution.
Σ	3×3 covariance matrix of position error of a space object.
Σ^*	Covariance matrix of the relative position vector projected onto the encounter plane.
σ_N	Positional standard deviation in the normal component of the RTN frame of the object
σ_R	Positional standard deviation in the radial component of the RTN frame of the object
σ_T	Positional standard deviation in the along-track component of the RTN frame of the object

Roman symbols

$\mathbf{A}$	Transition matrix of the underlying Markov Chain in Hidden Markov Models.
$\mathbf{r}$	Position vector of a space object.
$\mathbf{r}_{\text{rel}}$	Relative position vector between the chaser and target objects.
$\mathbf{r}^*_{\text{rel}}$	Relative position vector between target and chaser projected onto the encounter plane.
$\mathbf{v}$	Velocity vector of a space object.
$\mathbf{X}$	Sequence of observations of a Hidden Markov Model.
$\mathbf{x}_n$	n-th observation of the observation sequence of a Hidden Markov Model.
$\mathbf{Z}$	Sequence of states of the underlying Markov Chain in a Hidden Markov Model.

$\mathbf{z}_n$ n-th state of the underlying Markov Chain in a Hidden Markov Model.

HBR Sum of the radii of chaser and target objects.

$\bar{\mathbf{r}}$ Nominal position vector of a space object.

$\bar{\mathbf{r}}_{\text{rel}}$ Nominal relative position vector between target and chaser.

$\bar{\mathbf{r}}^*_{\text{rel}}$ Nominal relative position vector between target and chaser projected onto the encounter plane.

K Number of hidden states of a Hidden Markov Model.

$M_{\text{RTN,x}}$ Transformation matrix that rotates the covariance matrix from the ECI to the RTN reference frame of the object x, where x can be C (chaser) or T (target).

N Number of observations.

P_c Probability of collision.

R Radii of space object

T Projection matrix to the conjunction encounter plane reference frame.

Approach A Method followed in this work in which the risk of collision contained in the CDMs is directly modeled and predicted.

Approach B Method followed in this work in which the position errors of the chaser, contained in the collision warnings, are modeled and predicted, and the risk of collision is computed with the predictions.

Subscripts

ECI Quantity represented in the ECI reference frame.

RTN,x Quantity represented in the RTN reference frame of the object x, where x can be C (chaser) or T (target).

Superscripts

-1 Inverse.

chaser Quantity referred to the chaser object.

combined Combined quantity between target and chaser

T Transpose.

target Quantity referred to the target object.

Glossary

18 SPCS 18th Space Control Squadron.

AI Artificial Intelligence.

CAC Collision Avoidance Challenge.

CDM Conjunction Data Message.

ECI Radial Transverse Normal.

EDA Exploratory Data Analysis.

ESA European Space Agency.

FN False Negatives.

FP False Positives.

GEO Geostationary Earth Orbit.

GPS Global Positioning System.

HAC High Accuracy Catalog.

HBR Hard Body Radius.

HDI Highest Density Interval.

HMC Hamiltonian Monte Carlo.

HMM Hidden Markov Model.

LEO Low Earth Orbit.

LKJ Lewandowski-Kurowicka-Joe.

MAE Mean Absolute Error.

MCMC Markov Chain Monte Carlo.

MH Metropolis Hastings.

ML Machine Learning.

NASA National Aeronautics and Space Administration.

NUTS No-U-Turn Sampler.

O/O Owner/Operator.

RMSE Root Mean Squared Error.

RTN Radial Transverse Normal.

SDO Space Debris Office.

SSA Space Situational Awareness.

SSN Space Surveillance Network.

TCA Time of Closest Approach.

TN True Negatives.

TP True Positives.

US United States.

USSPACECOM Unites States Space Command.

Chapter 1

Introduction

1.1 Motivation

1.1.1 Current Space Debris environment

Since the beginning of the space age, with the launch of Sputnik-1 in 1957, a large number of rockets, spacecraft, satellites, and other instruments have been launched into space. The amount of resident space objects in the orbit of Earth has been steadily increasing since then and so has the number of on-orbit explosions (due to residual fuel or other energy sources that remain on board once a spacecraft has been discarded) and involuntary collisions between them. As of estimations done in August 2022, about 6 250 rockets have been launched into space, placing approximately 13 630 satellites into Earth's orbit, since the start of the space age. At the moment, around 8 850 satellites are still in space and 6 400 are still functioning [1]. Figure 1.1 shows the evolution of the number of objects in geocentric orbit by object class.

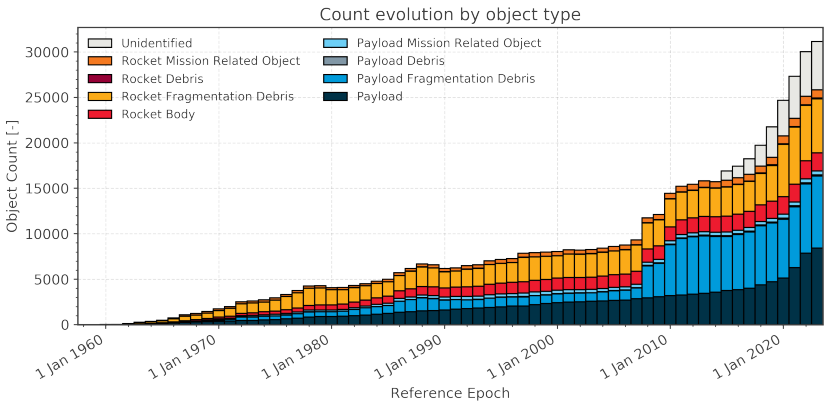


Figure 1.1: Count evolution of objects in space, by object type [2]. It can be seen that the number of objects in space has been continuously increasing since the beginning of the space age, with a sharp growth in the last decade.

The space environment is becoming increasingly crowded (Figure 1.1). Until 2015, the evolution of the number of objects in space was approximately linear with some sharp steps. For example, in 2009

there was an abrupt increase in the object count explained by the accidental collision of an operational United States (US) communications satellite — Iridium 33 — with a defunct Russian communications satellite — Cosmos 2251. This event marked the first time two satellites collided in orbit, generating thousands of pieces of debris and putting many other satellites in similar orbits at increased risk [3]. From 2015 forward, the number of objects in space has almost an exponential evolution. This is due to the fact that space traffic is undergoing notable changes fuelled by the development of commercial and private space activities and deployment of large constellations [4], especially in the Low Earth Orbit (LEO) region. A LEO is defined as the orbit of a spacecraft or other resident space object around the Earth with an altitude of 2 000 km or less. Figure 1.2 shows the evolution of the payload launch traffic into orbital altitudes that range from 200 km to 1 750 km (inside LEO region) per mission funding.

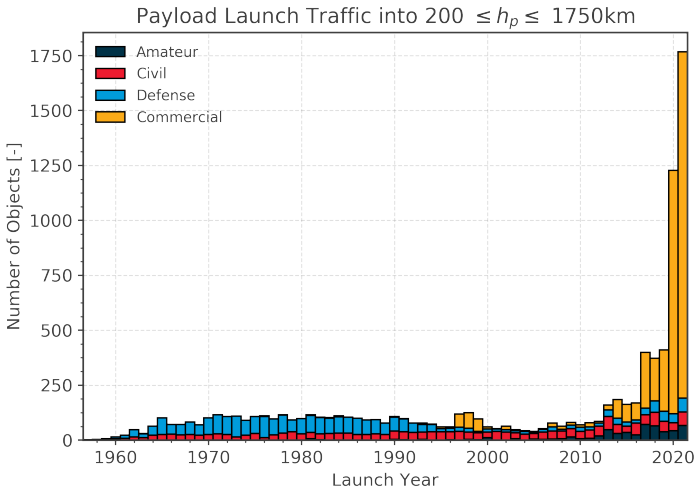


Figure 1.2: Yearly payload launch traffic into orbital altitudes (h_p) that range from 200 km to 1 750 km per mission funding [2]. It can be concluded that the number of objects launched to the LEO region has had a sharp increase in recent years and there has been a shift towards commercial space activities, in the last decade.

The number of objects launched into LEO has suffered a tremendous increase over the last years, especially since the last decade, with a shift towards commercial operators (Figure 1.2). The financing of applications for space-based initiatives is growing and the cost of launch systems, especially in LEO, is decreasing [5], which further adds to the growth of space traffic and the increase of space population. This continuous growth of the number of objects in space can pose a great danger to all operational satellites and other spacecraft since collisions between these objects create a large number of fragments that are further released into orbit.

However, space debris does not consist only of these fragments: discarded launch vehicles, parts of a spacecraft, defunct satellites, or any non-functional instruments orbiting Earth are at risk of being involved in catastrophic collisions. Therefore, space debris can be defined as *"any human-made object in orbit about the Earth that no longer serves a useful function"* [6].

There are multiple sources of space debris [7] such as explosions of spacecraft and rocket bodies due to residual fuel that remains onboard, space collisions, design flaws in satellites, or anti-satellite tests (e.g. the Chinese FengYun-1C event in 2007, which, at that time, increased the trackable space

population by 20% [8]). Most of the space debris have origin in spacecraft, so the highest spatial density of these objects will be at the orbital altitudes where the satellites and other instruments typically operate. Therefore, space debris is mostly located in the LEO region, which extends to 2 000 km above the surface of Earth. In particular, there is a higher density of space debris in the orbital altitudes between 600 km and 1 000 km (Figure 1.3).

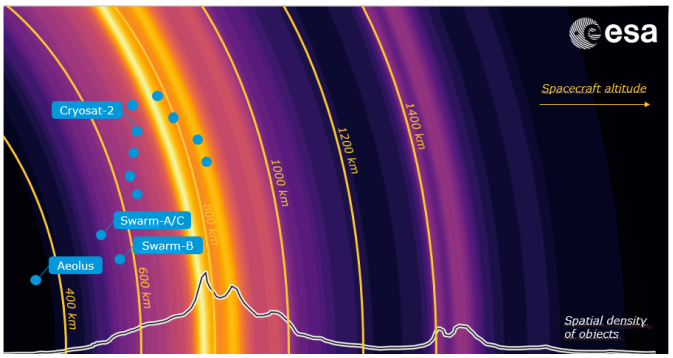


Figure 1.3: Spatial density of objects by orbital altitude [9]. There is a higher density of objects in the orbital altitudes between 600 km and 1 000 km.

It is estimated that more than 130 million space debris objects with a size of less than 1 cm are orbiting Earth, while there are 1 million pieces of debris with a size between 1 cm and 10 cm and 36 500 objects larger than 10 cm [1]. With the constant growth in space traffic, these numbers tend to increase day by day: the number of objects in orbit is directly related to how many collisions are expected to happen in the future and, as things stand, collisions between debris and active satellites are predicted to be the most dominant source of debris [10]. These collisions create hundreds of thousands of shards of dangerous debris that can collide with other objects at speeds that can reach 14 km/s [11], causing significant damage to a satellite or a spacecraft. The collisions between two resident space objects give rise to more debris and lead to more collisions, in a chain reaction, leading to an exponential increase in the number of artificial objects in space. This effect in which several major collisions cause a sequence of new collisions, due to the high density of objects in orbit, is known as the “Kessler Syndrome” and was first introduced by Donald Kessler in 1978 [12]. The NASA space debris expert concluded that, once past a certain critical mass, the total amount of space debris will keep on increasing [13], posing a great danger to active spacecraft.

The fragmentation events create numerous debris that is spread in different directions at different velocities. The resulting cloud of debris first starts by affecting only the immediate environment near the orbit where the fragmentation occurred [14]. Then, over the period of a few orbital revolutions, the cloud gets separated and further collisions may happen between the fragments, creating more dispersion in the distribution of the debris. Over time, the orbit perturbations (like the air drag and gravity field) create more dispersion and alter the orbital plane of the objects, leading to a gradual pollution of a vast volume of space. The dispersion of the cloud of debris over time can be visualized in Figure 1.4.

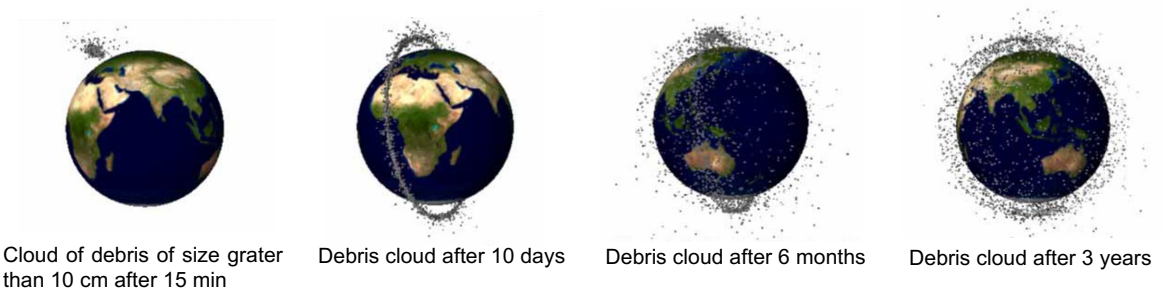


Figure 1.4: Illustration of the evolution of a fragment cloud over time, from the moment of fragmentation or collision [15].

As things stand, the density of objects in space, especially in LEO region, will be continuously rising, increasing the risk of a cascade of collisions. This scenario poses a great danger since, with such a large amount of fragmentation events, entire altitude shells could become entirely inhospitable, affecting the whole space population and creating numerous consequences. Earth is surrounded by spacecraft that carry out relevant missions such as delivering navigation services and global communication and, nowadays, scientific research and related operational applications in many areas depend on the data provided by these satellites. Societies are dependent on space infrastructures, so it is extremely important to avoid collisions between them. Additionally, in view of the fragmentation events that occurred during the last decades (such as the Iridium-33/Cosmos-2251 collision), the need to consider collision avoidance as part of routine operations is evident to all operators.

In the next subsection, the current collision avoidance process in European Space Agency (ESA) will be briefly explained.

1.1.2 Collision avoidance in ESA

ESA started its collision avoidance operations in 2006 with the formation of the Space Debris Office (SDO) which, currently, is responsible for the major research activities regarding space debris at ESA and routinely performs collision analysis for several satellites in the LEO region [16].

Collision avoidance generally requires the analysis and propagation of orbit information of resident space objects which is screened/compared against the propagated orbits of other objects, in order to identify close approaches. ESA has good knowledge about the location of their satellites, but relies on surveillance data from, mainly, US entities [16], that track and propagate the position of the other objects in space and send collision warnings in case of a close encounter. Sections 2.3.1 and 2.3.2 cover a brief overview of the conjunction assessment process and identification of close encounters, respectively. When the propagated orbits of a pair of resident space objects meet the conditions for a close encounter, collision warnings are issued to the owner/operator (O/O) in the form of Conjunction Data Messages (CDMs) that contain multiple information about the close encounter, such as the positions of both objects, the time of closest approach (TCA) and a self-computed risk of collision. All the parameters contained in the CDMs have their values propagated to the TCA. In collision avoidance, the close approach event between a pair of objects is known as *conjunction event*, the monitored space object is commonly

referred to as *target* and the other object that can potentially collide with the primary spacecraft is known as *chaser*. After the initial detection of a conjunction event, the target and chaser are routinely tracked, leading to the creation of more CDMs that contain updated and refined information about the conjunction: the estimation errors of the position of the objects grow with the propagation of the orbit, so the closer to the TCA the CDMs are generated, the shorter the time-span of the propagation and the more precise the information within the warning messages. The decision regarding an avoidance maneuver is then made based on the analysis of these collision warnings.

Typically, in each close encounter in LEO, a sequence of CDMs that covers seven days before the TCA is released, in which the last CDM is considered the best knowledge about the outcome of the given event [17]. However, the decision about the performance of an avoidance maneuver, generally, has to be made one day before the TCA [18], which means that satellite operators cannot wait for the last data message that usually arrives after this time threshold and, consequently, the decision has to be made based on preliminary knowledge. Hence, it can be extremely advantageous to forecast the collision risk of a given event, after this time threshold, since it allows spacecraft operators to make informed decisions about the need for potential avoidance maneuvers.

Additionally, the current collision avoidance process relies on the analysis of CDMs by human experts/analysts, that, even after automated processing and filtering of these messages, have on average two conjunction alerts per spacecraft per week that need to be further examined [19]. With the continuous growth of the space population, this manual approach may be an unfeasible task in the future, highlighting the importance of automation in risk assessment and estimation.

1.2 AI in collision avoidance and Related work

The major challenge in the analysis of the outcome of close encounters is the assessment of the information, or more precisely, the collision probability that future observations/CDMs will contain. An approach to overcome the problems described in Section 1.1.2 is to analyze and learn the data patterns of previous events using AI methods and use that knowledge to predict the information in future observations of new events, which can give better insights into their outcome.

In 2019, ESA launched the *Collision Avoidance Challenge* (CAC) [20] and released a dataset that contained sequences of CDMs received in support of real close encounters. The aim of the competition was to develop machine learning (ML) models capable of predicting the criticality of conjunction events by analyzing the time series of CDMs received up to 2 days prior to the predicted TCA, which is considered the cut-off time. The collision probability within the CDMs is computed through the Alfriend-Akella algorithm [21], which is described in subsection 2.3.3, and the final risk of each event is considered to be the risk contained in the last released CDM. Figure 1.5 illustrates the concept of AI in collision avoidance.

The competition showed that the naive/baseline approach (using the risk contained in the last CDM received until the cut-off time as the true risk value) is a strong predictor for this problem, with only 12 teams out of 97 managing to beat the benchmark solution [17]. Even in the cases where the teams man-

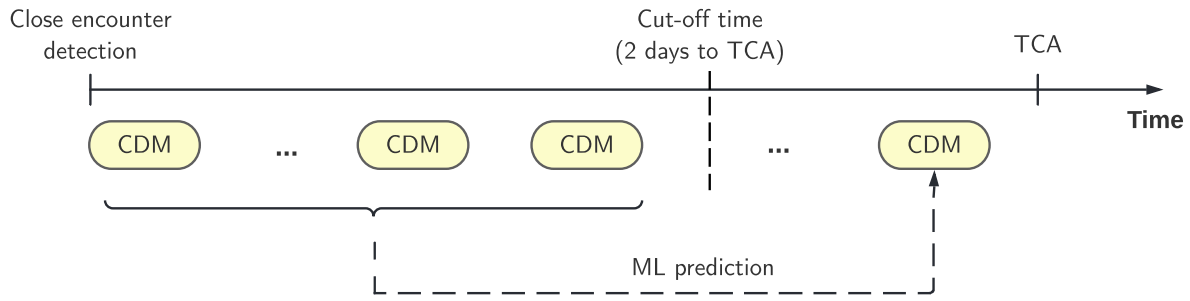


Figure 1.5: Illustration of the concept of the ML approach in collision avoidance. The cut-off time can take different values, depending on the mission. In this work, it was considered the same as in the CAC.

aged to overcome the naive forecast, their solutions didn't improve much when compared to the baseline. The team that presented the top solution used a step-by-step statistical approach to optimize the constitution of the test set and the competition metric. Another top-ranked team implemented Manhattan LSTMs [22], which have a siamese architecture based on recurrent neural networks, and only selected 7 of the features that showed strong predictive power by comparing non-anomalous events (events that are low-risk at the cut-off time and end up low-risk, and vice-versa for high-risk) with anomalous events (events that are low-risk at the cut-off time but end up being high-risk, and vice-versa for high-risk events). Gradient boosting trees also showed good performance.

After the competition, relevant work regarding the use of AI in collision avoidance has been conducted. Metz [23] implemented various models to predict the final chaser position uncertainties for each event and used those predictions to compute the risk using Akella's and Alfriend's algorithm [21]. Acciarini et al. [24] built a physics-based generative model using probabilistic programming to simulate the generation of CDMs, based on real data. Pinto et al. [25] used Bayesian deep learning with recurrent neural network architectures to also study the possibility of generating CDMs. Abay et al. [26] benchmarked the results for the state-of-the-art machine learning models that showed good results against the naive approach since the beginning of the competition.

As mentioned, the naive forecast, as well as its variants, are very strong predictors for collision risk assessment which indicate that the time series of CDMs may follow the Markov property [17], i.e., the information contained in the current CDM only depends on the values of the previous CDM. In this work, this property will be investigated through the implementation of Hidden Markov Models and, for that, two datasets are going to be used: the one that ESA released for the CAC challenge and a real-world raw dataset provided by *Neuraspace*.

1.3 Objectives and Contributions

This thesis aims to investigate whether the time-series evolution of the collision warnings released in support of close encounters between space objects follows the Markov property, that is, the values contained in one CDM only depend on the previous CDM, within each event. This hypothesis is tested by implementing HMMs for the risk prediction problem. In addition, the proposed work aims to benchmark

the performance of these models using Bayesian statistics, which allows the development of reliable predictive models that can provide prediction uncertainties reflecting the accuracy of the predictions, and can incorporate prior knowledge about the problem, within a rigorous theoretical framework.

1.4 Thesis Outline

Chapter 2 provides the necessary theoretical concepts used in this work, describing the conjunction detection and assessment process, the basic notions of Machine Learning, the method used in this work, and introducing the Bayesian approach. Chapter 3 presents and describes the CAC dataset. The methodology followed in this work is described in Chapter 4. Chapter 5 presents the data preparation and setup of the datasets used in this work. In Chapter 6, the implementation of the Bayesian models is described. The results are shown and discussed in Chapter 7. Lastly, the main conclusions of this work are presented in Chapter 8 as well as future work and research ideas.

Chapter 2

Background

This Chapter presents an introduction to the theoretical concepts used throughout this thesis. Section 2.1 introduces the reference frames used in this work to describe the orbital states of spacecraft. Section 2.2 provides some fundamentals about the underlying uncertainty in the position estimates of resident space objects, which is a key concept in risk estimation. In Section 2.3, the conjunction detection and assessment process is described. Section 2.4 provides some background on ML concepts and used metrics, and Section 2.5 provides insights into the model used in this thesis. Finally, in Section 2.6, Bayesian modeling is briefly explained.

2.1 Reference frames

To describe the orbital motion of objects in space, it is essential to define suitable reference frames. Without them, it would be impossible to precisely and unambiguously track and describe the motion of a satellite or space debris, much less calculate the risk of collision between two resident space objects. Throughout this thesis, two reference frames will be used: Earth-Centered Inertial (ECI) and Radial-Transverse-Normal (RTN).

The Earth-centered inertial (ECI) reference frame is a quasi-inertial frame that is mainly used to describe the motion of satellites [27]. It has its origin at the center of mass of the Earth with the positive $\mathbf{X}$-axis pointing in the vernal equinox direction, the $\mathbf{Z}$-axis aligned with the celestial North Pole and the $\mathbf{Y}$-axis completing the right-hand rule [28]. This reference frame is non-rotating with respect to the stars [28]. However, through time, there is a gradual change in the orientation of the Earth's rotation axis due to axial precession, which means that the equatorial plane is not fixed since it is normal to the Earth's rotation axis. Thus, it is necessary to define an epoch or a moment in time to be used as a reference. One commonly used ECI frame is the $J2000$ [29] which is defined with the Mean Equator and Equinox of the Earth at 12:00 Terrestrial Time on January 1, 2000. In the $J2000$ frame, the $\mathbf{X}$-axis points in the vernal equinox direction, and the $\mathbf{Z}$-axis is aligned with the North Pole as it was at that time. The $\mathbf{Y}$-axis completes the right-hand rule.

The radial transverse normal (RTN) reference frame is a local frame that is centered on the space-

craft. This reference frame moves with the spacecraft and is often used to describe its orbit. The $\mathbf{R}$-axis is aligned with the position vector $\mathbf{r}$ of the resident space object. The vector $\mathbf{r}$ has its origin in the center of mass of Earth and points towards the satellite. The $\mathbf{N}$-axis is normal to the orbital plane of the satellite and the $\mathbf{T}$-axis completes the right-hand rule. It is important to note that the $\mathbf{T}$-axis is not necessarily aligned with the velocity vector $\mathbf{v}$ of the spacecraft unless it is undergoing a circular orbit.

Figure 2.1 illustrates the ECI and RTN reference frames that are represented by the coordinate system $\mathbf{XYZ}$ and $\mathbf{RTN}$, respectively, and $\mathbf{r}$ represents the position vector of the satellite.

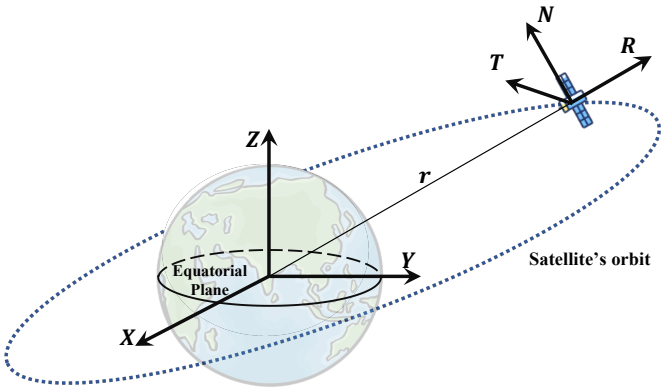


Figure 2.1: Example of the ECI and RTN reference frames, represented by the coordinate system $\mathbf{XYZ}$ and $\mathbf{RTN}$, respectively.

Consider that $\mathbf{r}$ and $\mathbf{v}$ are the satellite's position and velocity vectors in the $J2000$ ECI frame, respectively, and let $\hat{\mathbf{R}}$, $\hat{\mathbf{T}}$ and $\hat{\mathbf{N}}$ denote the unit vectors that compose the RTN reference frame. These unit vectors can be obtained from the coordinates of a satellite in the $J2000$ frame as follows:

$$\hat{\mathbf{R}} = \frac{\mathbf{r}}{\|\mathbf{r}\|}, \qquad \hat{\mathbf{N}} = \frac{\mathbf{r} \times \mathbf{v}}{\|\mathbf{r} \times \mathbf{v}\|}, \qquad \hat{\mathbf{T}} = \hat{\mathbf{R}} \times \hat{\mathbf{N}}, \tag{2.1}$$

where $\|\mathbf{a}\|$ is the Euclidean norm of vector $\mathbf{a}$ and $\mathbf{a} \times \mathbf{b}$ is the cross product between vectors $\mathbf{a}$ and $\mathbf{b}$.

The resulting vectors from equation (2.1) can be used to build the orthogonal matrix M_{RTN} that will be useful to map the covariance matrix from the $J2000$ frame into the coordinate system of the resident space object.

2.2 Uncertainty in position

With suitable reference frames and adequate equipment, one might think that it is possible to exactly determine and describe the position and velocity of satellites and other objects in space. However, these quantities are inevitably computed with a certain level of uncertainty, due to measurement errors. Additionally, the propagation of the orbit of a spacecraft makes the position uncertainty grow over time due to errors in propagation models that are unable to precisely describe the perturbing forces acting on a spacecraft [30]. A statistical estimation of these errors is extremely important and valuable to spacecraft operators since it provides information about the trustworthiness and reliability of the measurements and computations. This information is given in the form of covariance matrices.

2.2.1 Covariance matrix and error ellipsoid

A covariance matrix is a square, symmetric and positive semi-definite matrix whose entries represent the covariance between each pair of elements. Within the scope of the orbit determination problem, the covariance matrix describes the uncertainty in the determination of the orbital state of a resident space object and, hence, is a 9×9 matrix that represents the uncertainties in the estimation of the three components of the position, the velocity, and the drag. But, to compute the collision probability, only the position uncertainties are necessary, according to Akella and Alfriend [21], because it is assumed that the uncertainty in velocity can be neglected (as it will be seen in Section 2.3.3), so only the 3×3 submatrix containing the position covariances shall be considered hereafter. Thus, the covariance matrix for the estimation errors of the position of a resident space object in the ECI frame can be written as follows:

$$\Sigma_{\mathsf{ECI}} = \begin{bmatrix} \sigma_{\mathsf{X}}^2 & c_{\mathsf{XY}}\,\sigma_{\mathsf{X}}\,\sigma_{\mathsf{Y}} & c_{\mathsf{XZ}}\,\sigma_{\mathsf{X}}\,\sigma_{\mathsf{Z}} \\ c_{\mathsf{XY}}\,\sigma_{\mathsf{Y}}\,\sigma_{\mathsf{X}} & \sigma_{\mathsf{Y}}^2 & c_{\mathsf{YZ}}\,\sigma_{\mathsf{Y}}\,\sigma_{\mathsf{Z}} \\ c_{\mathsf{XZ}}\,\sigma_{\mathsf{Z}}\,\sigma_{\mathsf{X}} & c_{\mathsf{YZ}}\,\sigma_{\mathsf{Z}}\,\sigma_{\mathsf{Y}} & \sigma_{\mathsf{Z}}^2 \end{bmatrix}, \tag{2.2}$$

where σ_{X}, σ_{Y}, and σ_{Z} are the standard deviations in the directions of the ECI frame and c_{XY}, c_{XZ} and c_{YZ} are the correlation coefficients. To rotate the covariance matrix from the ECI frame into the coordinate system of the satellite, one can use the orthogonal matrix M_{RTN}, whose columns are the vectors of equation (2.1):

$$\Sigma_{\mathsf{RTN}} = M_{\mathsf{RTN}}\,\Sigma_{\mathsf{ECI}}\,M_{\mathsf{RTN}}^{T}. \tag{2.3}$$

It is assumed that the noise in the position estimates is normally distributed, so the three-dimensional distribution of the position uncertainties can be visualized as an ellipsoid centered on the spacecraft. In theory, the error ellipsoid can stretch to infinity, since it is a representation of normal distributions, but the probability decreases significantly as the distance increases. Hence, these ellipsoids often represent the 1σ position error. The three components of the error ellipsoid are known as along track or transverse, radial, and normal, which represent the directions $\mathbf{T}$, $\mathbf{R}$ and $\mathbf{N}$, respectively. The standard deviations in the diagonal elements of the covariance matrix determine the length of each component of the ellipsoid and the correlation coefficients on the off-diagonal entries define its orientation in space. If the position errors are uncorrelated, i.e., if the covariance matrix is diagonal, the along-track axis of the ellipsoid will be aligned with the trajectory of the spacecraft. Figure 2.2 illustrates the position uncertainty ellipsoid of a spacecraft.

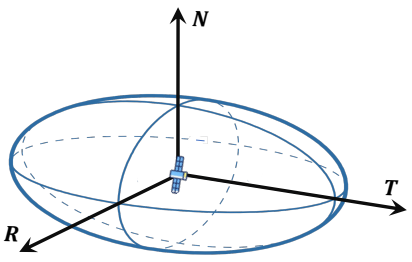


Figure 2.2: Illustration of a position uncertainty ellipsoid of a spacecraft, where it is assumed that the noise in the position estimates is normally distributed and the position uncertainties are uncorrelated.

Typically, in LEO, the along-track component of the ellipsoid is much larger than the radial and normal components, in other words, the level of uncertainty in the along-track direction is higher. This is due to the fact that non-conservative forces like the drag or solar radiation tend to affect the mean motion of the spacecraft (consequently affecting the along-track movement), but not the inclination [29, p. 747].

2.2.2 Combination of covariance matrices

Another useful concept in this work is the combination of covariance matrices. In conjunction events, the target and chaser have their respective position uncertainties and, to compute the probability of collision between them, it is necessary to add their covariance matrices, as will be seen in Section 2.3.3. This combination can only be performed if both covariance matrices are described in the same reference frame, but the datasets commonly contain the standard deviations and correlation coefficients of the resident space objects described in their own RTN reference frame. In this thesis, this problem is solved by rotating the covariance matrix of the chaser object into the RTN coordinate system of the target object.

This transformation has the following steps. Let $\Sigma_{\text{ECI}}^{\text{chaser}}$, $\Sigma_{\text{RTN,C}}^{\text{chaser}}$ and $\Sigma_{\text{RTN,T}}^{\text{chaser}}$ denote the covariance matrix of the chaser in the ECI reference frame, in its own RTN system and in the RTN frame of the target, respectively. Firstly, it is necessary to rotate the chaser covariance matrix from its coordinate system into the ECI reference frame. This can be done by, first, computing the transformation matrix $M_{\text{RTN,C}}$ that maps the covariance matrix from the $J2000$ frame into the chaser coordinate system, using the vectors from equation (2.1) and, then, obtaining its inverse $M_{\text{RTN,C}}^{-1}$ that represents the inverse rotation, i.e., maps the covariance matrix from the chaser coordinate system into the $J2000$ frame. It is important to note that the rotation matrix $M_{\text{RTN,C}}$ is orthogonal, so $M_{\text{RTN,C}}^{-1} = M_{\text{RTN,C}}^{T}$. Hence, the chaser covariance in the ECI frame is given by:

$$\Sigma_{\text{ECI}}^{\text{chaser}} = M_{\text{RTN,C}}^{-1}\, \Sigma_{\text{RTN,C}}^{\text{chaser}}\, (M_{\text{RTN,C}}^{-1})^{T} = M_{\text{RTN,C}}^{T}\, \Sigma_{\text{RTN,C}}^{\text{chaser}}\, M_{\text{RTN,C}}. \tag{2.4}$$

Secondly, it is necessary to rotate the chaser covariance matrix from the $J2000$ ECI reference frame into the RTN coordinate system of the target, using the transformation matrix $M_{\text{RTN,T}}$. This rotation matrix can be obtained using the vectors from equation (2.1) with input vectors $\mathbf{r}_T$ and $\mathbf{v}_T$ that represent the position and velocity vectors of the target in the $J2000$ reference frame. With the transformation matrix $M_{\text{RTN,T}}$ and the chaser covariance in the ECI frame, the chaser covariance in the coordinate system of the target can be obtained by:

$$\Sigma_{\text{RTN,T}}^{\text{chaser}} = M_{\text{RTN,T}}\, \Sigma_{\text{ECI}}^{\text{chaser}}\, M_{\text{RTN,T}}^{T}. \tag{2.5}$$

Finally, considering that the position errors of the target and chaser are independent, the combined covariance matrix of the two objects is given by:

$$\Sigma_{\text{RTN,T}}^{\text{combined}} = \Sigma_{\text{RTN,T}}^{\text{chaser}} + \Sigma_{\text{RTN,T}}^{\text{target}}, \tag{2.6}$$

where $\Sigma_{\text{RTN,T}}^{\text{target}}$ represents the covariance matrix of the target in its RTN reference frame.

2.3 Conjunction Detection and Assessment

The need to consider collision avoidance operations to prevent damage to spacecraft and the cascade effect of colliding objects in near-Earth orbit is evident. However, it is impossible to precisely predict ahead of time whether a collision of two space objects will happen or not, because the orbit determination and propagation of the objects are subject to errors and uncertainties, as explained in the previous Section. So, the decision about the need for a collision avoidance maneuver is based on the analysis of the computed collision probability that is associated with that event. But, it would be infeasible to analyze the risk of collision of all objects in space, so the collision probability is only assessed for the events that pass the conditions of a close encounter. Thus, collision avoidance procedures typically have four steps: an automated screening for potential collisions, risk assessment of "close conjunction events", refinement of the risk assessment with updated orbital data and collision avoidance action [31]. This Section shall focus on a brief explanation of the conjunction assessment process, identification of conjunction events, computation of the collision probability, and the definition of high-risk events.

2.3.1 Current conjunction assessment process

Collision avoidance can only be performed if the position of the space objects is known. Usually, satellite operators know with reliable precision the position of their satellites, in LEO, using the global positioning system (GPS) [32]. However, it is hard to track all the other non-active satellites and space debris, so space operators rely on surveillance data from other sources, mainly non-European.

The United States Space Command (USSPACECOM) is the US government entity responsible for space situational awareness (SSA), routinely tracking and cataloging space objects using their Space Surveillance Network (SSN) [33], which consists of a variety of high-quality sensors that detect and track resident space objects orbiting Earth. This surveillance data is maintained in a catalog and is given as input to identify close approaches. The conjunction assessment is performed by the 18th Space Control Squadron (18 SPCS) that uses the SSN observations and implements "Special Perturbation" orbit propagation theory to perform orbit determination for each man-made object in the catalog. Orbit determination is usually performed every eight hours, in which the orbits of the objects are typically propagated seven days into the future and their positions are then updated in the 18 SPCS High Accuracy Catalog (HAC) [33]. To search for close approaches, 18 SPCS screens the propagated orbits of the objects against the rest of the objects in HCA and, when the conditions for a close approach are met (see subsection 2.3.2), the 18 SPCS notifies the owner/operator using CDMs, through the SSA sharing program. After receiving these collision warnings, ESA also performs its own collision probability estimation using the internal data about its satellites, which is assumed to be more accurate and reliable than the data of the satellite provided by USSPACECOM.

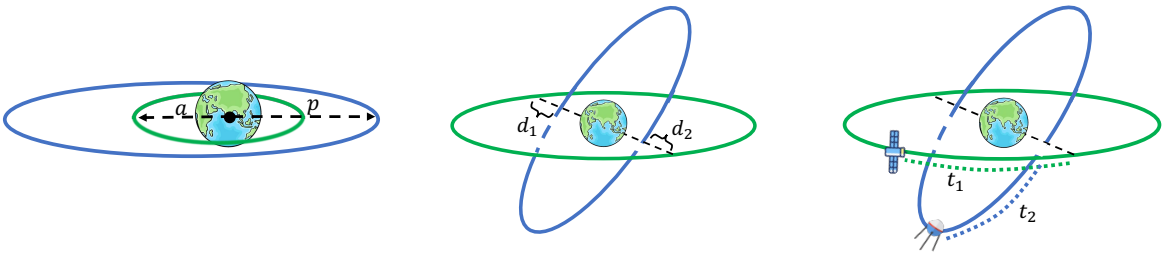
The next subsection briefly explains how conjunction events are identified.

2.3.2 Identification of Conjunction events

When the cataloged data of a monitored spacecraft is screened/compared against the data in the rest of the catalog, filtering techniques are used in order to identify close conjunctions. These techniques provide an efficient mechanism for finding close approaches, by quickly identifying pairs of target/chaser that cannot come close enough together to yield a conjunction. Among different methods, the process described hereafter follows the techniques proposed by Klinkrad [34].

Close conjunction events can be detected by the successive application of an altitude filter, a plane geometry filter, and a phase filter. The first filter is based on orbit altitudes: it is assumed that object pairs whose range of altitude values of the two trajectories do not overlap cannot produce conjunctions and are eliminated [35]. Given a pair of space objects, let p denote the largest value of the two perigees and a denote the smallest of the two apogees, where the perigee is the nearest point to the Earth in the orbit of a spacecraft and the apogee is the furthest point to the Earth of an orbit. Hoots et al. [36] formulated that, if $p-a \geq D$, where D represents a threshold distance (that takes into account the orbital decay and short-periodic altitude variations of both objects), a collision cannot occur. An example of a pair of objects that would be eliminated by this filter is a Geostationary Earth Orbit (GEO) and a LEO satellite pair because a GEO satellite operates at an altitude of, approximately, 35 800 km and a LEO spacecraft operates at a maximum of 2 000 km. Next, the plane and geometry filter is applied for those objects that passed the apogee/perigee filter. In this step, the pairs that, at the intersection of the chaser and target orbit plane, have position separations greater than the conjunction threshold are eliminated. This filter is based on the radius/size of the objects on the intersection line between the two orbits [31] and the threshold is defined as the Hard Body Radius (HBR) which is given by the sphere centered in the target and with radius equal to the sum of both spacecraft circumscribing radii. When the computed distance is smaller than the threshold, the event passes to the phase filter. In this last step, the times of passing the intersection node for both the target and chaser are compared and overlaps are checked. This filter eliminates all the objects whose propagation time spans don't overlap.

Figure 2.3 illustrates the conjunction filters.



(a) Altitude Filter. The pair is eliminated if $p-a \geq D$, where D represents a threshold distance.

(b) Plane Geometry Filter. The pair is eliminated if d_1 and d_2 are larger than the HBR.

(c) Phase Filter. The pair is eliminated if the time spans don't overlap.

Figure 2.3: Illustration of the conjunction filters that allow the identification of close approaches. The blue and green ellipses describe the trajectories of each object of the pair that is under analysis for close conjunction.

The object pairs that pass all the filters are identified as close conjunction events.

2.3.3 Collision probability computation

Once a conjunction event is identified, the probability of collision can be estimated and the decision of a collision maneuver is then based on a defined threshold designated as the accepted collision probability level [31]. When that threshold is exceeded, an avoidance maneuver is needed.

There are various methods for computing the collision probability, but the one used in this thesis and discussed in this Section is based on the Alfriend and Akella method [21], developed in 2000, that computes the collision probability by performing a two-dimensional integration of the rigid body projection in the encounter plane. This approach, due to the short duration of the conjunction events, is based on the following assumptions/approximations, around the time of the close encounter:

- the target and chaser nominal trajectories can be represented by straight lines with constant velocities;
- there is no velocity uncertainty for both objects and position uncertainties of the target and chaser are assumed to remain constant during the encounter and equal to the ones at the TCA;
- the uncertainties in the positions of the objects are independent and follow Gaussian distributions;
- the target and chaser are considered spherical objects with radii R^{target} and R^{chaser}, respectively.

Consider that, at the time of closest approach, the nominal positions of the target and chaser, in an arbitrary reference frame, are given by $\bar{\mathbf{r}}^{\text{target}}$ and $\bar{\mathbf{r}}^{\text{chaser}}$, respectively. Adding the position uncertainties variations to the position vectors of the objects, one can obtain

$$\mathbf{r}^{\text{target}} = \bar{\mathbf{r}}^{\text{target}} + \delta^{\text{target}}, \qquad \mathbf{r}^{\text{chaser}} = \bar{\mathbf{r}}^{\text{chaser}} + \delta^{\text{chaser}}, \tag{2.7}$$

where it is assumed that the uncertainties follow normal distributions given by $\delta^{\text{target}} \sim \mathcal{N}(0, \Sigma^{\text{target}})$ and $\delta^{\text{chaser}} \sim \mathcal{N}(0, \Sigma^{\text{chaser}})$, in which Σ^{target} and Σ^{chaser} are 3×3 matrices corresponding to the covariance matrices of the target and chaser positions propagated to the TCA, respectively, and δ^{target} and δ^{chaser} represent the position uncertainties in the position of the target and chaser, respectively. The relative position of the objects is defined as

$$\mathbf{r}_{\text{rel}} = \mathbf{r}^{\text{chaser}} - \mathbf{r}^{\text{target}} = (\bar{\mathbf{r}}^{\text{chaser}} - \bar{\mathbf{r}}^{\text{target}}) + (\delta^{\text{chaser}} - \delta^{\text{target}}) = \bar{\mathbf{r}}_{\text{rel}} + \delta^{\text{combined}}, \tag{2.8}$$

in which $\bar{\mathbf{r}}_{\text{rel}} = \bar{\mathbf{r}}^{\text{chaser}} - \bar{\mathbf{r}}^{\text{target}}$ represents the nominal relative position between the target and chaser and $\delta^{\text{combined}} = \delta^{\text{chaser}} - \delta^{\text{target}}$ represents the combined position uncertainty. Based on equation (2.8), the miss vector can be divided into a deterministic component — $\bar{\mathbf{r}}_{\text{rel}}$ — and a stochastic component that expresses uncertainty — δ^{combined}. Since the term $\bar{\mathbf{r}}_{\text{rel}}$ is deterministic, the variance of $\mathbf{r}_{\text{rel}}$ is equal to the variance of δ^{combined}, as $\mathbb{E}[\mathbf{r}_{\text{rel}}] = \bar{\mathbf{r}}_{\text{rel}}$ and $\text{cov}[\mathbf{r}_{\text{rel}}] = \Sigma^{\text{chaser}} + \Sigma^{\text{target}} = \Sigma^{\text{combined}}$. Note the combination of the target's and chaser's covariance matrices is equal to their sum, because it is assumed that δ^{target} and δ^{chaser} are independent and are given in the same reference frame. Therefore, it is possible to conclude that

$$\mathbf{r}_{\text{rel}} \sim \mathcal{N}(\bar{\mathbf{r}}_{\text{rel}}, \Sigma^{\text{combined}}), \tag{2.9}$$

where the components of the miss-vector are denoted by $\mathbf{r}_{\text{rel}} = [\, r_1, r_2, r_3 \,]^T$.

As mentioned before, the target and chaser are considered to be spherical objects with radii R^{target} and R^{chaser} and their combined sphere is described by the hard body radius $\text{HBR} = R^{\text{target}} + R^{\text{chaser}}$, with volume V_{HBR}. It is considered that a collision occurs if the two spheres intersect, i.e., if the volume of the combined sphere with radii HBR is equal to the sum of the volume of the target's and chaser's separate spheres. Therefore, the probability of collision (P_c) can be computed from the volume integral of the probability density function presented in (2.9), over the spherical region V_{HBR}:

$$P_c = \frac{1}{(2\pi)^{\frac{3}{2}} \left|\Sigma^{\text{combined}}\right|^{\frac{1}{2}}} \iiint_{V_{\text{HBR}}} \exp\Big(-\frac{1}{2} (\mathbf{r}_{\text{rel}} - \bar{\mathbf{r}}_{\text{rel}})^T (\Sigma^{\text{combined}})^{-1} (\mathbf{r}_{\text{rel}} - \bar{\mathbf{r}}_{\text{rel}}) \Big)\, dr_1\, dr_2\, dr_3, \tag{2.10}$$

in which $|a|$ represents the determinant of the matrix a. However, Alfriend and Akella show that, in the TCA, there is no uncertainty in the relative position vector in the direction of the relative velocity, which motivates the definition of an encounter plane described by a new orthogonal coordinate system [21]. The encounter plane is perpendicular to the relative velocity vector between the chaser and target at the TCA and it is shown that the miss-vector $\mathbf{r}_{\text{rel}}$ lies within the newly defined plane, allowing the projection of the three-dimensional position error into the two-dimensional encounter plane, through an orthogonal transformation matrix T. All the details, steps and justifications to perform these projections can be found in Alfriend's and Akella's paper [21] and shall be skipped at this point. Figure 2.4 illustrates the projections into the encounter plane, in the vicinity of the TCA.

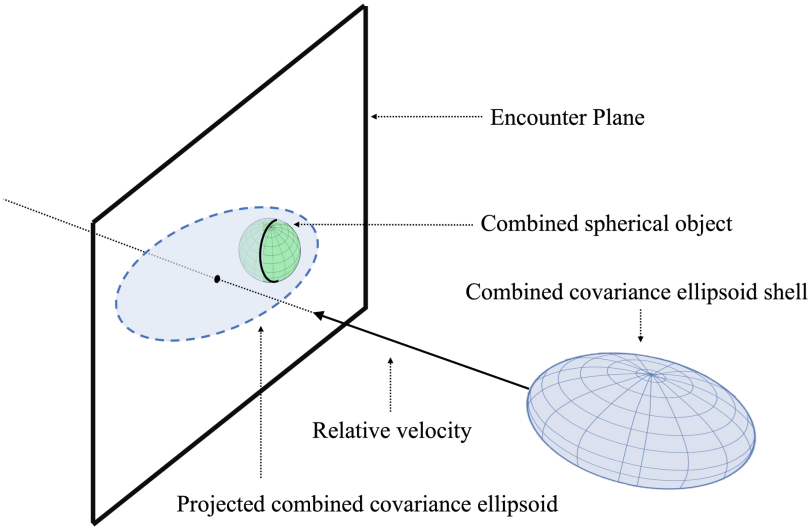


Figure 2.4: Illustration of the geometry of the conjunction encounter in the vicinity of the TCA. The combined error ellipsoid is located at the chaser and the combined spherical object is centered at the target.

Given the transformation matrix T, the projection of the nominal relative position and combined covariance in the encounter plane, denoted by $\bar{\mathbf{r}}^*_{\text{rel}}$ and Σ^*, respectively, can be obtained as follows:

$$\bar{\mathbf{r}}^*_{\text{rel}} = T\, \bar{\mathbf{r}}_{\text{rel}} \tag{2.11}$$

$$\Sigma^* = T\, \Sigma^{\text{combined}}\, T^T. \tag{2.12}$$

Since that the three-dimensional position errors follow a Gaussian distribution, its projection into the encounter plane will also follow a Gaussian distribution [37]. Hence, it is possible to conclude that

$$\mathbf{r}^*_{\text{rel}} \sim \mathcal{N}(\overline{\mathbf{r}}^*_{\text{rel}}, \Sigma^*), \tag{2.13}$$

in which its components are going to be denoted as $\mathbf{r}^*_{\text{rel}} = [\, x_1, x_2 \,]$.

Finally, the probability of collision can be obtained by the two-dimensional integral over the area defined by the sum of the radii of the target and chaser:

$$P_c = \frac{1}{2\pi|\Sigma^*|^{\frac{1}{2}}} \int_{-\text{HBR}}^{\text{HBR}} \int_{-\sqrt{\text{HBR}^2 - x_1^2}}^{\sqrt{\text{HBR}^2 - x_1^2}} \exp\Big(-\frac{1}{2}(\mathbf{r}^*_{\text{rel}} - \overline{\mathbf{r}}^*_{\text{rel}})^T \Sigma^{*-1} (\mathbf{r}^*_{\text{rel}} - \overline{\mathbf{r}}^*_{\text{rel}}) \Big)\, dx_2 dx_1. \tag{2.14}$$

2.3.4 Definition of high-risk events

If the collision probability between a pair of objects exceeds a defined threshold, the event is considered a high-risk event. The definition of this threshold is based on a trade-off between the maximum risk reduction and the estimated number of maneuvers: a higher collision probability threshold leads to a risk reduction at the cost of a higher number of maneuvers and vice-versa. In practice, a threshold of 10^{-4} is used as default, since it offers a good trade-off value [16], but it can usually range between 10^{-5} and 10^{-4}, depending on the mission [17].

In the *Collision Avoidance Challenge*, the threshold was chosen as 10^{-6}, because of the imbalance nature of the data or, in other words, in order to have a higher number of high-risk events [17]. In this work, since the dataset of the competition is used, the same threshold for the collision probability is chosen. Note that, throughout this work, the base-10 logarithm of the collision probability is commonly used and its value is referred to as *risk*. Therefore, in this thesis, the events with a risk equal to or higher than –6 are considered high-risk events.

2.4 Machine Learning Overview

In order to predict the risk of collision at the time of closest approach, machine learning (ML) methods are used. This section introduces the learning scenarios of ML and presents the evaluation metrics used in this work to assess the performance of the implemented models.

2.4.1 Learning Tasks

ML methods can be applied to a large variety of different problems; depending on the nature of the learning data, the learning scenarios can be divided, mainly, into three categories:

- Supervised learning: refers to models and algorithms that receive training samples with the corresponding labels (prespecified description, target, or desired outputs). In these tasks, the goal is to find the rules that map the input data to the corresponding outputs and, then, make predictions and determine the label (or value) for new/unseen inputs.

- Unsupervised learning: in this type of scenario, the models and algorithms receive input data without label. In these cases, the aim is to find hidden information/patterns within the data without any reference and, then, make predictions for new points. This type of learning is commonly used, among many other tasks, in anomaly detection and clustering.

- Reinforcement learning: this ML technique enables the algorithms to learn in a dynamic and interactive environment using feedback from the data and, in this scenario, the training and evaluation/testing phases are intermixed [38].

In this work, the training data is a set of conjunction events and the goal is to find learning rules that map the CDMs received until a decision threshold (two days before the TCA) to the corresponding risk contained in the last CDM of that event. Since the training data is labeled and the aim is to predict the labels for new input data, supervised learning methods will be used. Supervised learning can be further separated, based on the type of output variables, into two subcategories: classification and regression. Classification aims to assign one of a set of discrete categories to each input sample, whereas regression consists of predicting quantitative outputs [37, p. 3].

The goal of this work is to create an algorithm capable of, not only correctly identifying the risk of collision between two space objects (a continuous target variable), but also distinguishing between high and low-risk events (two categories). So, both regression and classification problems are the focus of this thesis. Supervised ML problems in these two subcategories use different metrics to evaluate the corresponding performance of regression and classification models.

2.4.2 Evaluation metrics

One extremely important step in the ML workflow is the assessment of the performance of the implemented algorithms and, in supervised learning, this is done by comparing the predictions of the test samples with the true labels contained in the test set. If the training set was used to measure the performance of the model, the evaluation would be too optimistic and the model could overfit the training data. Data overfitting refers to models that fit the training data too well and do not generalize well to new data. This happens when, during the training phase, a model learns the details and noise in the training data that are not present in the overall data, negatively impacting the performance of the algorithm on unseen samples. Therefore, it is important to evaluate the models with a set of data that is independent of the set used for training.

In this work, to verify how close the predictions are to the true values, two regression metrics are used: root mean squared error (RMSE) and mean absolute error (MAE). RMSE is the square root of the average of the squared difference between the true and predicted values and is measured in the same units as the target variable. MAE is the average of the absolute differences between the predicted values and target values, measuring the average of the residuals. These metrics are given by:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}, \qquad \text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|, \tag{2.15}$$

in which y represents the true values, $\hat{y}$ the predicted values, N the number of samples in the set and the subscript i represents the i-th data point. The RMSE penalises large prediction errors more, since the errors are squared before being averaged, and MAE returns more interpretable results, because it is just the average of the prediction errors.

Regarding the evaluation of the classification task of identifying whether an event is of low or high risk (a binary classification problem), classification metrics are needed. Binary classification problems involve a *positive* class and a *negative* class, where the first is often referred to the class one is most interested in predicting (in this case, the high-risk events) and the latter is referred to the other class (in this case, the low-risk events) [39, p. 276]. In this thesis, the classification performance will be assessed through confusion matrices, in which the predictions can result in four different outcomes: true positives (TP, instances correctly classified as positive - high-risk events correctly classified), true negatives (TN, negative instances correctly classified as negative - low-risk events correctly classified), false positives (FP, miss-classifications of negative instances - low-risk events wrongly classified as high-risk) and false negatives (FN, miss-classifications of positive instances - high-risk events wrongly classified as low-risk). Within the scope of this work, since the data is extremely imbalanced (there are very few high-risk events), it is important to take the distribution of the classes into consideration, when evaluating the model. Hence, precision and recall are going to be used. Precision measures the number of positive class predictions that are actually positive and recall measures the proportion of the positive samples that are captured by the positive predictions. These two metrics are given as follows:

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \qquad \text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{2.16}$$

Additionally, to evaluate both recall and precision in one metric, the F_β score can used. This metric is given by

$$F_\beta = (1 + \beta^2)\frac{\text{precision} \cdot \text{recall}}{(\beta^2 \cdot \text{precision}) + \text{recall}}, \tag{2.17}$$

in which the β parameter determines the weight of recall in the combined score and, as β increases, more weight is placed on recall. The particular case where $\beta = 1$ is known as the F_1-score and is the harmonic mean of precision and recall. The F_2-score will also be considered, because, in space applications, it is more important to minimize the number of FN than the number of FP, i.e., it is more important to minimize the number of samples wrongly predicted as low-risk than to minimize the number of events wrongly predicted as high-risk. False positives can lead to potentially unnecessary avoidance maneuvers and to energy waste, but false negatives give spacecraft operators false confidence about the outcome of a conjunction event, which can lead to catastrophic collisions.

2.5 Probabilistic Graphical Models

Probabilistic graphical models provide a natural tool for dealing with two problems: uncertainty and complexity, through the combination of probability and graph theory [37]. These models allow a natural and intuitive integration of expert/domain and physical knowledge, proving to be a fundamental tool in

probabilistic modeling. In this work, hidden Markov models (HMM) are going to be used, which are a type of graphical model. This section introduces the main concepts of Markov chains and presents a brief explanation of Hidden Markov Models and some of their properties.

2.5.1 Markov Chains

A Markov chain (or Markov process) is a stochastic model that describes a sequence of possible observable events. It is a probabilistic model that experiences transitions between states, according to certain probabilistic rules [40], and satisfies the Markov property. This property states that "*the probability distribution of future states of the process conditioned on both the past and present states depends only on the present state*" [41]. In simpler terms, in Markov models, it is assumed that to predict the future observation in a sequence, all that matters is the current state. The state space S of a Markov chain can be either continuous or discrete, but, since HMM are based on transitions between a finite number of states, Markov chains with a discrete state space will be the focus of this explanation.

It will be considered, without loss of generality, that in Markov chains with a discrete state space the set of possible states S is composed by a finite set of K elements, $S = \{1, 2, ..., K\}$, and that the observed variables are discrete multinomial variables describing which state the observation corresponds to. Thus, a Markov chain can be seen as a series of states (observations) drawn from the set S. At this point, it is convenient to define some notation to describe the assignment of data points to the possible states of the model. Vectors (all considered column vectors) are denoted, just like in Bishop's book [37], as bold lowercase letters, such as $\mathbf{z}$, and its components are denoted by z_i. Hence, a sequence of N observations in a Markov chain can be defined as $\mathbf{Z} = \{\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_N\}$. For each observation $\mathbf{z}_n$, since there are K possible states, an *1-of-K representation* will be used. With this notation, each variable $\mathbf{z}_n$ is represented by a K-dimensional vector, in which one of its elements z_{nk} equals 1 and the other elements equal 0 [37, p.74], according to which state of the set S that observation corresponds to. E.g., consider a Markov chain with three possible states and two observations, i.e, $S = \{1, 2, 3\}$ and $\mathbf{Z} = \{\mathbf{z}_1, \mathbf{z}_2\}$. The second observation $\mathbf{z}_2$ corresponds to the first state if $z_{21} = 1$ and $z_{22} = z_{23} = 0$, so that $\mathbf{z}_2 = [1, 0, 0]^T$.

To illustrate a Markov chain, consider a sequence of N observations given by $\mathbf{Z} = \{\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_N\}$. Accordingly with the Markov property, the conditional distribution for any observation $\mathbf{z}_{n+1}$, given all the previous observations up to n, is given by

$$p(\mathbf{z}_{n+1}|\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_n) = p(\mathbf{z}_{n+1}|\mathbf{z}_n). \tag{2.18}$$

Hence, a Markov chain representing a sequence of N observations can be represented as the graphical model from Figure 2.5.

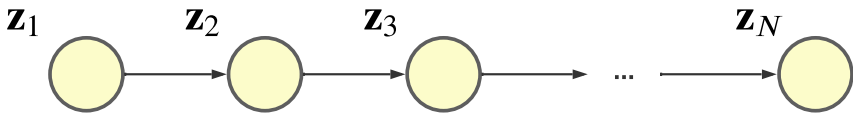


Figure 2.5: Graphical model of a Markov chain for a sequence of observations $\mathbf{Z} = \{\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_N\}$.

To formally specify a discrete Markov chain, three elements are needed: a finite set of K states, a transition probability matrix and an initial probability distribution. The initial probability distribution is given by the vector $\boldsymbol{\pi} \in \mathbb{R}^K : \boldsymbol{\pi} \geq 0,\ \boldsymbol{\pi}^T \mathbb{1} = 1$ (in which $\mathbb{1}$ denotes a K-dimensional vector with elements equal to 1), where π_k is the probability of the chain starting in state k, such that $\pi_k \equiv p(z_{1k} = 1)$ and $\sum_{k=1}^{K} \pi_k = 1$. The transition matrix will be represented by $\mathbf{A} \in \mathbb{R}^{K \times K} : \mathbf{A} \geq 0,\ \mathbf{A}\mathbb{1} = \mathbb{1}$, in which i-th row of $\mathbf{A}$, that will be denoted as $\mathbf{A}_i \in \mathbb{R}^K : \mathbf{A}_i \geq 0,\ \mathbf{A}_i^T \mathbb{1} = 1$, is a conditional probability distribution that describes the probabilities of transitioning to one of the K possible hidden states, given that the chain is in state i, and each element $\mathbf{A}_{ik}$ represents the probability of moving from state i to state k, i.e., $\mathbf{A}_{ik} \equiv p(z_{nk} = 1 | z_{n-1,i} = 1)$, with $0 \leq \mathbf{A}_{ik} \leq 1$ and $\sum_{k=1}^{K} \mathbf{A}_{ik} = 1$.

Therefore, a Markov chain follows the following probabilistic rules:

$$p(\mathbf{z}_1 | \boldsymbol{\pi}) = \prod_{k=1}^{K} \pi_k^{z_{1k}}, \qquad p(\mathbf{z}_n | \mathbf{z}_{n-1}, \mathbf{A}) = \prod_{k=1}^{K} \prod_{i=1}^{K} \mathbf{A}_{ik}^{z_{n-1,i}\, z_{nk}}. \tag{2.19}$$

One of the properties of the Markov chain is that every state can be observed directly, however, in some problems, it is not possible to directly observe the sequence of states, but only their effects.

2.5.2 Hidden Markov Models

When it is not possible to observe the states themselves, but only the result of some probability function of the states, Hidden Markov Models are used. HMM is a tool for representing probability distributions over sequences of observations [42] that is based on augmenting the Markov Chain. In HMM, the observations are produced by an underlying stochastic process, whose states cannot be directly observed, i.e., are *hidden* [43]. In the context of this work, there is a physical process happening in space in which the two objects in risk of colliding approach each other. This process cannot be seen (is hidden) and the CDMs can be interpreted as measurements that occur from that physical process.

In HMMs, the hidden process that generates the observations is a first-order finite state Markov chain and, hence, respects the Markov property. The sequences of hidden states will be denoted as $\mathbf{Z} = \{\mathbf{z}_1, \mathbf{z}_2, ...\}$ and the sequences of observations as $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ...\}$, where each hidden state $\mathbf{z}_n$ generates the corresponding observation $\mathbf{x}_n$ (which may be of different type or dimension to the latent variable [37, p. 609]). Therefore, a HMM can be interpreted as an extension of a mixture model in which the choice of mixture component (hidden state $\mathbf{z}_n$) depends on the choice of component for the previous observation $\mathbf{z}_{n-1}$ [37, p. 610]. Note that the states of the underlying Markov chain are considered to be drawn from a finite vocabulary of K elements given by $S = \{1, 2, ..., K\}$, as it was previously explained.

A HMM with N observations is depicted as a graphical model in Figure 2.6, in which the nodes of the hidden states are uncolored (because the variables are not observed) and the nodes of the observations are shaded in yellow.

A HMM is then characterized by the following elements: a finite set of K possible hidden states, a transition probability matrix $\mathbf{A}$, an initial probability distribution $\boldsymbol{\pi}$ and the emission probabilities. Since the underlying stochastic process in HMM is a Markov chain, it follows the same probabilistic rules as

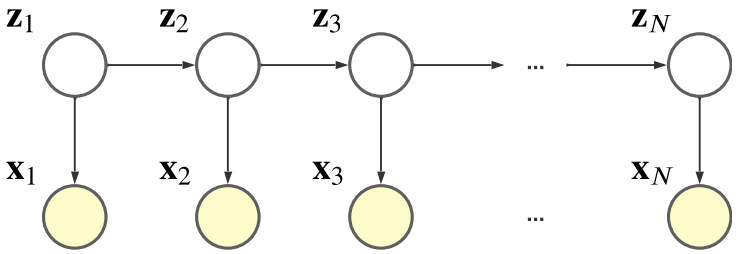


Figure 2.6: Graphical model of a Hidden Markov Model for a sequence of N observations.

the ones presented in equation (2.19), where $\mathbf{A} \in \mathbb{R}^{K \times K} : \mathbf{A} \geq 0,\ \mathbf{A}\mathbb{1} = \mathbb{1}$ represents the probability of transitioning from one hidden state to another and $\boldsymbol{\pi} \in \mathbb{R}^K : \boldsymbol{\pi} \geq 0,\ \boldsymbol{\pi}^T \mathbb{1} = 1$ represents the hidden state initialization probability. The specification of a HMM is completed by defining the emission probabilities: the conditional distributions of the observed variables $p(\mathbf{x}_n|\mathbf{z}_n, \boldsymbol{\phi})$, where $\boldsymbol{\phi}$ are the parameters that govern the distribution. These emission probabilities can be either discrete or continuous and, since each hidden state $\mathbf{z}_n$ is a 1-of-K vector of K possible states, the distribution $p(\mathbf{x}_n|\mathbf{z}_n, \boldsymbol{\phi})$ also consists of a K-dimensional vector, for a given value of $\boldsymbol{\phi}$ [37, p. 611], corresponding to the possible K states of $\mathbf{z}_n$. The emission probabilities are given as follows:

$$p(\mathbf{x}_n|\mathbf{z}_n, \phi) = \prod_{k=1}^{K} p(\mathbf{x}_n|\phi_k)^{z_{nk}}. \tag{2.20}$$

For convenience, the compact notation $\theta = (\mathbf{A}, \boldsymbol{\pi}, \boldsymbol{\phi})$ will be used to indicate the set of parameters of the model and it will be considered that θ is D-dimensional, i.e, $\theta \in \mathbb{R}^D$. During the training phase, these parameters must be learned, which leads to one of the fundamental problems in HMM: the learning problem. In this problem, one is given a sequence of observations $\mathbf{X}$ and the set of possible states S and the goal is to learn the HMM parameters θ. This is typically solved by finding the set of parameters that maximizes the distribution $p(\mathbf{X}|\theta)$ (maximum-likelihood estimation) using an expectation-maximization algorithm known as Baum-Welch method [43]. However, in this thesis, rather than estimating a single set of parameters, an entire joint distribution for θ will be inferred. This is done by using a Bayesian approach, in which Bayes' theorem is used to update the values of the parameters conditioned on the observed data (posterior distribution) [44]. Section 2.6 covers the Bayesian modeling approach with more detail, but it is important to note that it relies on the likelihood distribution of the model to estimate the parameters.

Likelihood distribution

In HMMs, the likelihood distribution $p(\mathbf{X}|\theta)$ describes the joint probability of the sequence of observations $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$ conditioned on the set of parameters θ and it is given as follows [37]:

$$p(\mathbf{X}|\theta) = \sum_{\mathbf{z}_N} \alpha(\mathbf{z}_N), \tag{2.21}$$

where $\alpha(\mathbf{z}_n)$ is known as the forward variable, which represents the joint distribution of all the observations up to n and the latent variable $\mathbf{z}_n$, i.e, $\alpha(\mathbf{z}_n) = p(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n, \mathbf{z}_n|\theta)$. The forward variable, for each step, is given by

$$\alpha(\mathbf{z}_n) = p(\mathbf{x}_n|\mathbf{z}_n) \sum_{\mathbf{z}_{n-1}} \alpha(\mathbf{z}_{n-1}) p(\mathbf{z}_n|\mathbf{z}_{n-1}), \tag{2.22}$$

and, to start the recursion, the initial condition can be written as $\alpha(\mathbf{z}_1) = p(\mathbf{z}_1)p(\mathbf{x}_1|\mathbf{z}_1)$. All the steps and justifications to obtain these equations can be found in Bishop's book [37, pp. 619-623].

Predictive Distribution

In this work, since the aim is to predict the risk of collision of two resident space objects contained in the CDMs received after a certain threshold, a problem that must be tackled is the prediction problem, in which the observed data $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$ is given and the goal is to predict the next k observations: $\mathbf{x}_{N+1}$, $\mathbf{x}_{N+2}$, ..., $\mathbf{x}_{N+k}$. To do this, one must compute the predictive distribution $p(\mathbf{x}_{N+k}|\mathbf{X}, \theta)$.

In Bishop's book [37], it is possible to find the detailed derivation of the distribution $p(\mathbf{x}_{N+1}|\mathbf{X}, \theta)$ that is given as follows:

$$p(\mathbf{x}_{N+1}|\mathbf{X}, \theta) = \frac{1}{p(\mathbf{X}|\theta)} \sum_{\mathbf{z}_{N+1}} p(\mathbf{x}_{N+1}|\mathbf{z}_{N+1}, \theta) \sum_{\mathbf{z}_N} p(\mathbf{z}_{N+1}|\mathbf{z}_N, \theta)\alpha(\mathbf{z}_N). \tag{2.23}$$

In this work, the performance of the HMMs is benchmarked only by predicting the next CDM/ observation of each event, which is denoted as $\mathbf{x}_{N+1}$. In real-life operations, during a conjunction event, 2 or 3 CDMs are received, on average, per day and the last CDM received is assumed to be the best knowledge about the outcome of that event. Thus, future work may test the performance of predicting the next k observations of an event, using HMMs.

2.6 Bayesian Modeling

In probabilistic models, the set of parameters θ is typically obtained by finding the parameters that result in the best match between the model and the observed data $\mathbf{X}$. However, in this work, as already mentioned, rather than estimating a single set of parameters, an entire joint distribution for θ will be inferred, reflecting its uncertainty and providing interpretable results. This is possible by adopting a Bayesian approach, in which the unknown parameters are treated as random variables and probability theory is used to learn its values conditioned on the data [44]. The Bayesian interpretation considers that the associated randomness of θ encapsulates the prior belief one holds about the problem and that the belief is updated by the available observations [45, 46]. Hence, this approach provides a natural way of combining the prior knowledge about a given problem with data, within a rigorous theoretical framework.

Bayesian modeling is based on the Bayes' theorem, which states that

$$p(\theta|\mathbf{X}) = \frac{p(\mathbf{X}|\theta)\, p(\theta)}{p(\mathbf{X})}, \tag{2.24}$$

in which $p(\theta)$ denotes the prior distribution, $p(\mathbf{X}|\theta)$ the likelihood, $p(\mathbf{X})$ the evidence, and $p(\theta|\mathbf{X})$ the posterior distribution. These distributions can be defined as follows:

- the likelihood distribution describes the joint distribution of the observed data conditioned on the parameters of a model, and it expresses how likely the observed data is for different values of θ [37];
- the prior distribution is a distribution that expresses one's prior knowledge/belief about an unknown quantity of interest (in this case, the parameters of a model) before any data is observed or taken into account;
- the evidence or marginal likelihood represents the probability of generating the data $\mathbf{X}$ from the priors;
- the posterior distribution is the joint distribution over the parameters of the model conditioned on the observed data and is a compromise between the likelihood and prior distributions [47].

Once the posterior is computed, it can be used to obtain predictions for new input data. However, computing the distribution $p(\theta|\mathbf{X})$ analytically is usually unfeasible, since it depends on the computation of the term $p(\mathbf{X})$ (the marginal likelihood), which is a normalizing factor,

$$p(\mathbf{X}) = \int_{\theta} p(\mathbf{X}, \theta)\, d\theta = \int_{\theta} p(\mathbf{X}|\theta)p(\theta)\, d\theta, \tag{2.25}$$

where it is necessary to integrate over all the possible values of θ, which becomes intractable for a large number of parameters since it turns into a high dimensional integral. To address this issue, Markov chain Monte Carlo (MCMC) methods can be used. In this work, Bayesian statistical models are implemented using a probabilistic programming framework called *PyMC* [48] that allows for flexible and intuitive specification and fitting of the models, featuring highly efficient MCMC methods.

2.6.1 Markov Chain Monte Carlo algorithms

MCMC methods approximate the posterior distribution using samples. These algorithms generate posterior samples iteratively, in which the next sample is only dependent on the value of the current sample, hence the notion of Markov chain. MCMC methods are based on the construction of a continuous state-space Markov chain that has the distribution that is desired to sample from (in this case, the posterior distribution) as its stationary distribution. A stationary distribution for a Markov chain (if it has one) is a distribution that, after running the chain for a large number of iterations, is the probability distribution for every new observation, and the observations can be interpreted as draws from the stationary distribution.

Through an accept-reject routine of proposed samples in a Monte Carlo simulation and the evaluation of the prior and likelihood distributions at different parameter values, MCMC methods induce the previously described Markov chain and guarantee a set of samples that are a good approximation to the desired distribution [44, 49].

Metropolis Hastings

One well-known MCMC method is Metropolis-Hastings (MH) [50, 51], which is a simple algorithm that provides a foundation for more complex and sophisticated methods. This algorithm generates proposed samples using a proposal distribution, and then the sample is accepted or not based on an acceptance probability that ensures that the distribution of samples is similar to the distribution that is desired to sample from.

MH is one of the most popular MCMC methods due to its simplicity and easy implementation, however, its performance greatly depends on the choice of the proposal distribution, which, if poorly chosen, can lead to a low acceptance rate and a slow convergence to the target distribution, making this algorithm inefficient. Additionally, the MH methods tend to generate samples that are close to each other, especially if the variance of the proposal distribution is small, which can cause a poor exploration of the density regions of the posterior distribution and can cause autocorrelation within the sampled values. These problems can be solved or reduced by using more sophisticated methods, such as Hamiltonian Monte Carlo (HMC).

Hamiltonian Monte Carlo

The HMC [52] algorithm is a version of the MH algorithm that uses gradient information and physics concepts to generate new proposed samples and is only applicable to continuous parameters. HMC borrows ideas from physics, more specifically, it uses Hamiltonian dynamics to generate proposed samples with high acceptance probability, making it more efficient in exploring the density regions. A detailed explanation of the HMC algorithm, as well as physical intuition about how it works can be found in Appendix A.

When compared to MH, HMC reduces the autocorrelation between samples, by proposing distant values for θ, and increases the acceptance probability, by adopting a Hamiltonian dynamics approach, which makes it a very efficient sampler. The HMC algorithm assigns, for each sampling iteration, a direction to explore in the parameter space, instead of searching it with random proposals, allowing a coherent and efficient exploration of the desired distribution. However, HMC heavily relies on the selection of some hyperparameters (the number of steps L at each sampling iteration, the leapfrog stepsize ε, and the mass matrix M - see Appendix A for more details), which, if poorly chosen, can lead to a reduction in the efficiency of this algorithm. *PyMC* has built-in adaptive dynamics HMC methods that can adapt the parameters during sampling [44]: the *No-U-Turn Sampler* algorithm [53], which is an extension of the HMC that allows the automatic adaptation of the number of steps L at each sampling iteration. Additionally, *PyMC* estimates the mass matrix M and the stepsize ε based on warm-up/tuning iterations.

2.6.2 Diagnosing MCMC Inference

As described in the previous subsection, MCMC algorithms allow the approximation of the posterior distribution without having to solve Bayesian models analytically. However, problems may arise during

inference, so it is necessary to evaluate the results, in order to verify if they are of any use. This subsection describes the main problems and diagnosis tools for MCMC methods.

Convergence diagnosis

Under certain conditions, MCMC algorithms generate samples from the desired distribution once the induced Markov Chain has converged to equilibrium, i.e., when the stationary distribution of the Markov Chain is achieved. In theory, the convergence to the target distribution is guaranteed when the number of draws of the MCMC algorithms approaches infinity [44, 54], but, in practice, the number of samples is finite, so there are no guarantees whether the induced Markov chains have converged. Therefore, it is essential to evaluate the convergence of the algorithms.

In equilibrium, the distribution of samples obtained from the numerical methods should always be the same, independently of the initial starting values of the chain [55]. Hence, a good way to ensure the convergence to the stationary distribution of a MCMC algorithm is by sampling multiple independent chains and verifying if the distribution of samples in each chain is similar [54]. Additionally, running multiple chains can help to detect potential problems in sampling, since different chains can explore different regions of the posterior distribution. Visualizing trace plots (charts that show the parameter samples for each chain) can give a good insight into the convergence of the sampled chains, but for a model with a large number of parameters, it can be difficult to interpret and analyze a large number of charts. Hence, a numerical summary is needed. One of the most widely used convergence diagnostics is the potential scale reduction factor $\hat{R}$. The $\hat{R}$ statistic for each quantity of interest is computed as "*the standard deviation of that quantity from all the chains included together, divided by the root mean square of the separate within-chain standard deviations*" [54]. In equilibrium, all the sampled chains should have a similar distribution and, hence, the variance between chains should be identical to the variance within chains. Thus, $\hat{R}$ should be close to one and, in practice, a value of $\hat{R} \leq 1.1$ is recommended [44].

Autocorrelation diagnosis

Another problem posed by MCMC algorithms is the autocorrelation within chains. The samples obtained from MCMC methods have always some degree of autocorrelation which can lead to an unrepresentative approximation of the true underlying posterior distribution since correlated samples contain less information than random independent draws of $p(\theta|\mathbf{X})$. Although MCMC samples are dependent, the inferences can still be valid, if the number of iterations is large enough to explore the density regions of the posterior distribution, but the efficiency of the sampler can be largely affected. One way to evaluate the autocorrelation of the sampled chains is through the effective sample size (ESS) which can be defined as "*the equivalent number of independent samples generated directly from the target distribution, which yields the same efficiency in the estimation obtained by the MCMC [...] algorithms*" [56]. In other words, ESS measures the level of the information content of the sampled chains by providing the number of draws one would have if the sample was actually taken randomly from the desired distribution. For more details and estimation steps of the ESS, the reader is encouraged to read the work of Vehtari

et al. [54], which provides an improved version of the ESS estimate proposed by Gelman et al. [47]. In practice, a value of ESS greater than 400 is recommended for each sampled quantity [44]. In order to inspect the level of autocorrelation, it can also be useful to visualize autocorrelation plots.

Label Switching

Before analyzing the convergence and autocorrelation of the inferences, it is often necessary to deal with a very common problem in Bayesian inference of mixture models: the *label switching* problem. *Label switching* refers to the problem in which the posterior distribution is invariant to permutations in the labeling of the parameters and, hence, during MCMC sampling, the labels of the parameters can switch between or within chains [57]. This happens when the likelihood distribution and the priors are the same for all permutations of θ, making the posterior invariant to switching component labels [58]. To illustrate the label switching problem, consider a Gaussian Mixture model with two components, in which the mean values of the normal distributions are given by μ_1 and μ_2, the shared standard deviation is given by σ and the mixing weight by $w \in [0, 1]$. For a set of N independent observations denoted by $\mathbf{Y} = \{y_1, y_2, ..., y_N\}$, the likelihood is given as:

$$p(\mathbf{Y}|\mu_1, \mu_2, \sigma, w) = \prod_{i=1}^{N} [\, w\, \mathcal{N}(y_i|\mu_1, \sigma) + (1 - w)\, \mathcal{N}(y_i|\mu_2, \sigma)\,]\,. \tag{2.26}$$

It can be seen that the likelihood yields the same result for the permutation of parameters μ_1 and μ_2 between the two Gaussian distributions and, if the priors are similarly invariant to these permutations, switching component labels can occur during Bayesian inference, as it is illustrated in Figure 2.7, that shows the between chain label switching for the parameters μ_1 and μ_2.

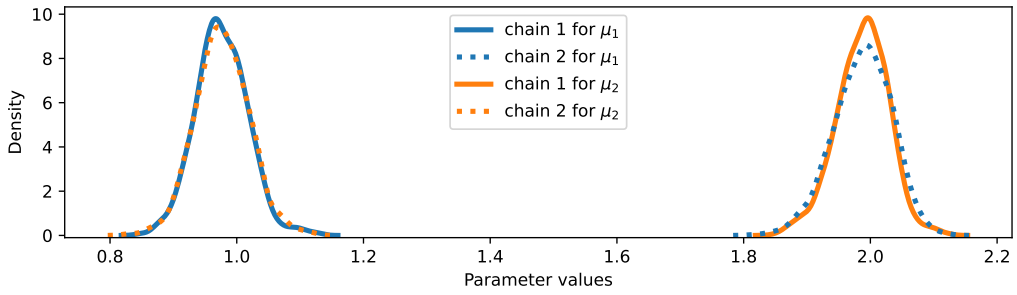
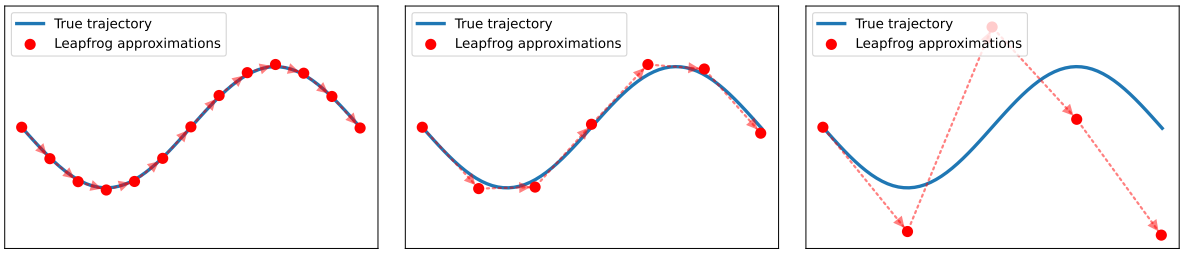


Figure 2.7: Example of label switching between chains. It shows the distribution of two sampled chains for the parameters μ_1 and μ_2. If there was no label switching, the distribution of the two sampled chains of each parameter should be similar.

An obvious response to this problem would be to break the symmetry of the prior distribution by imposing identifiability constraints on the parameters, such as $\mu_1 < \mu_2$. But, if the choice of suitable constraints is not done with great care, this approach often does not solve the switching problem and may lead to wrong inference [58]. More sophisticated solutions have been proposed [57–60], but, if the sampled chains only contain switching component labels between different chains, a very effective and simple technique to solve this problem is to post-process the sampled chains and relabel them according to some statistical analysis (like the median or mean of the sampled chains).

Divergent Transitions

The HMC methods, which are currently the most commonly used MCMC algorithms due to their efficiency, rely on the discretization of the trajectory of a fictitious particle (representing parameter values) rolling on the density regions of the negative log posterior distribution, via the leapfrog algorithm — read Appendix A for more details. These discretizations introduce errors in the simulated trajectory and, consequently, the total energy of the particle at the end of the trajectory (H^*) may be different from the total energy at the beginning (H), even though the energy is conserved — read Appendix A for more details. Typically, the values of H^* and H diverge by a factor of 10^{-7} [55], but, when the divergences are too high (normally the divergence threshold is set to a factor of 10^3 [55]), it is considered that the simulation is untrustworthy, since the approximation of the path of the particle is far from the true trajectory, failing to correctly explore the posterior density regions. These divergent transitions happen when the step size ε is too large relative to the curvature of the desired distribution. Figure 2.8 illustrates how ε can affect the approximation of the true path of a particle, showing that the smaller the step size, the more accurate the discretized trajectory.



(a) $\varepsilon = 0.52$ (b) $\varepsilon = 1.01$ (c) $\varepsilon = 1.72$

Figure 2.8: Example of the leapfrog algorithm with different step sizes for solving the differential equation $\frac{\partial^2 x(t)}{\partial t^2} + x(t) = 0$, i.e., approximating a trajectory given by $x(t) = \sin(t)$ for a period of $T = 2\pi$, where x is the position and t is time. The number of steps L was defined so that $L \times \varepsilon = T$.

Figures 2.8a and 2.8b demonstrate that, in each leapfrog iteration, since ε is small relative to the trajectory curvature, the algorithm tends to correct the error made in the previous step. On the other hand, figure 2.8c shows that, for large ε, the estimated path is not a good representation of the true trajectory of the particle and, consequently, the Hamiltonian at the end of the trajectory is significantly different from its initial value, causing a divergent transition. Divergence can occur when the desired distribution is hard to explore, due to highly varying posterior density curvatures. One way to solve this problem is to adjust the parameter ε that is estimated during the warm-up phase to match a defined acceptance rate. This can be done by adjusting the target acceptance rate to higher values in order to reduce the number of rejected samples and have more accurate estimations of the Hamiltonian in each HMC iteration, consequently decreasing the step size. Another solution is to re-parameterize the model so that the posterior density curvatures are less sharp and easier to explore.

Chapter 3

Data Analysis

In this chapter, the CAC dataset is analyzed. Section 3.1 provides a brief description and overview of the dataset and, in Section 3.2, an exploratory analysis is carried out.

3.1 Data Description

One of the datasets used in this work is the one provided and published by ESA for the Kelvins Collision Avoidance Challenge in 2019 [20], which consists of CDMs collected by the ESA Space Debris Office between 2015 and 2019, in support of collision avoidance operations in the LEO region. Each row of the dataset corresponds to a single CDM and each CDM has 103 features that contain information about the close encounter. It is important to note that all the parameters contained in the CDMs have their values propagated to the TCA. The CDMs are identified by an event ID and data messages from the same conjunction event are grouped under the same identifier. Hence, each event represents a time series of CDMs that typically covers one week leading up to the TCA. Table 3.1 presents a general description of the dataset.

CAC Dataset	
CDMs (rows)	162 634
Events	13 154
CDMs per Event	$\sim$12
Features (columns)	103

Table 3.1: CAC Dataset description

A complete description of all the parameters contained in the dataset can be found in `https://kelvins.esa.int/collision-avoidance-challenge` [20].

3.2 Exploratory Data Analysis

Exploratory data analysis (EDA) refers to the approach/process of analyzing the main characteristics of the data, its patterns, trends, and relationships, detecting anomalies and checking assumptions. The concept of EDA was first introduced by Tukey [61] and is strongly associated with graphical and visual representations of statistical analysis of the data [62]. The dataset contains a large number of parameters and, consequently, a large number of statistical analyses can be carried out. In this Section, only the most relevant information and analysis are presented, in order to provide a good insight into the data and the context of the problem, without going beyond a reasonable scope for this work.

Through the analysis of the parameters of the CAC dataset (that are described in [20]), it is possible to divide the data analysis problem into six categories: (i) events and CDMs analysis, (ii) TCA and risk analysis, (iii) conjunction location and geometry, (iv) object properties, (v) orbital elements and (vi) estimation errors. This Section shall focus on a descriptive analysis of each category.

When analyzing a dataset, it can also be advantageous to perform a correlation analysis of its features, which measures the statistical relationship between different parameters, can help to understand the features that have a strong correlation with the label/target variable, and suggests what feature associations should be further explored by ML models. However, to benchmark the performance of Bayesian HMMs, in this work, each model will only learn the evolution of one single feature of the dataset, as it will be explained in Chapter 4. Therefore, in the context of this thesis, a correlation analysis is not of great importance and shall not be presented in this Chapter.

3.2.1 Events and CDMs analysis

As presented earlier, there are 13 154 events and 162 634 CDMs in the dataset, giving an average of, approximately, 12 CDMs per event. Figure 3.1 shows the distribution of the number of CDMs received per event.

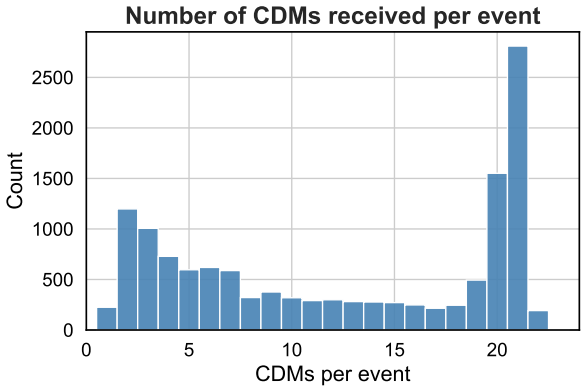


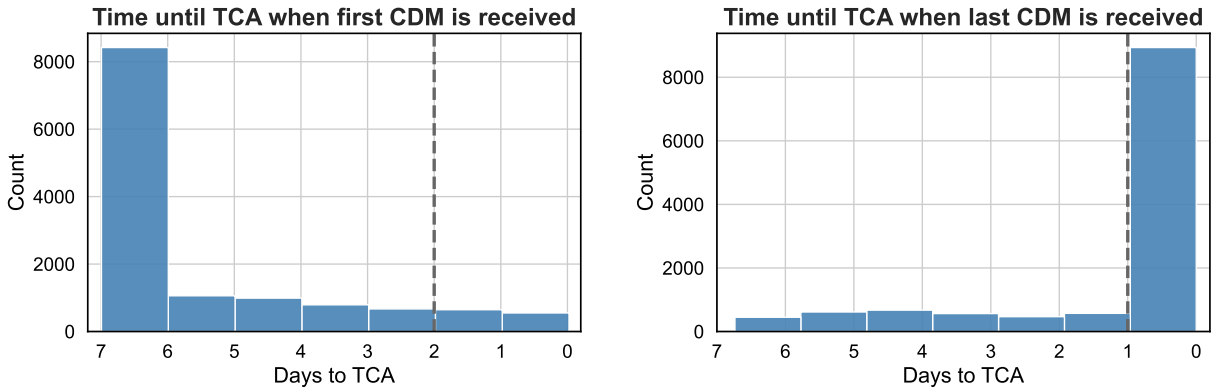
Figure 3.1: Distribution of the number of CDMs received per event. It is possible to verify that the the number of CDMs received per event is not uniformly distributed with higher values at the extremes of the distribution.

It is possible to see that the number of CDMs per event is neither uniformly distributed nor is centered near the average of 12 CDMs per event. In fact, there are higher values at the extremes of the distri-

bution, with 50% of the events containing less than 5 or more than 20 CDMs. However, not all events contained in the dataset are eligible for the ML approach. The events should follow some constraints, since spacecraft operators need time to make a decision regarding the performance of an avoidance maneuver. These constraints are the following [17]:

- the events must have at least 2 CDMs, one to learn and one to use as label;
- the first CDM has to be released before the cut-off time (2 days until TCA);
- the last CDM has to be released within 1 day of the TCA.

Regarding the first constraint, it was verified that 1.71% of the events contain only one collision warning. As for the other requirements, Figure 3.2a and 3.2b represent the distribution of the time in which the first and last CDMs are received, respectively.



(a) Distribution of the time in which the first CDM is received in each event.

(b) Distribution of the time in which the last CDM is received in each event.

Figure 3.2: Distribution of the time in which the first and last CDM are received, in each event. In the left figure, the dashed line represents the cut-off time for the decision threshold and, in the right figure, it marks the time to the TCA after which the last CDM has to be released. The plots show that some events do not satisfy the CDM requirements (at least one CDM before the cut-off time and one CDM after the one day until TCA threshold).

Figure 3.2a shows that the majority of the events receive the first CDM before the cut-off time, but there are some events that only receive the first collision warning after this threshold. In fact, 1212 events (9.2%) do not satisfy this requirement. Figure 3.2b shows that there are various events (approximately 28%) that receive their last collision warning before the 1 day to the TCA threshold, with 1829 events receiving their last CDM before 4 days until the TCA. This can happen, for example, when after the initial detection of a close encounter, refined and updated observations show that the event no longer passes the close conjunction criteria (described in subsection 2.3.2) and, consequently, no CDMs are issued after that. In total, 4904 events (approximately 37.2%) do not satisfy the CDM requirements.

3.2.2 TCA and Risk analysis

In the dataset, there are 391 CDMs with a negative value for the parameter *time_to_tca* (that represents the days before the TCA in which the collision warning is generated), which means that some

collision warnings were created after the closest point of approach. These CDMs (if they are not created a long time after the TCA) can offer good knowledge about the outcome of the close encounter, representing a good label for the ML approach.

Regarding the criticality of the events contained in the dataset, Figure 3.3 represents the distribution of the final risk of collision of each event (risk contained in the last released CDM). The histogram in Figure 3.3 shows that the data is extremely imbalanced, with the large majority of the events being low-risk. In the dataset there are 12 789 low-risk events (97.23%) and only 365 high-risk conjunctions (2.77%), highlighting the rare occurrence of high-risk events. It is important to note that the values of the risk contained in the CDMs provided by the 18 SPCS are truncated at a lower bound of –30, creating a peak at that value. In fact, approximately 63.5% of the total number of conjunction events contained in the CAC dataset has a final risk of –30.

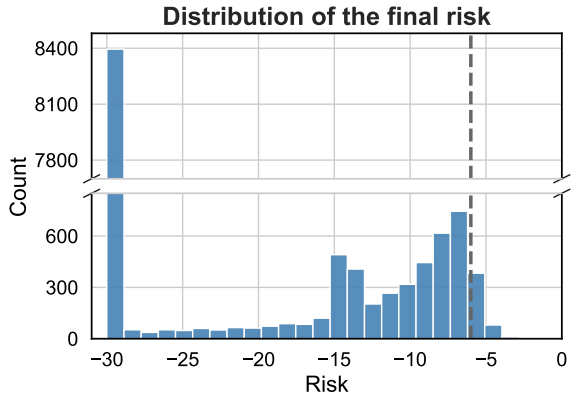
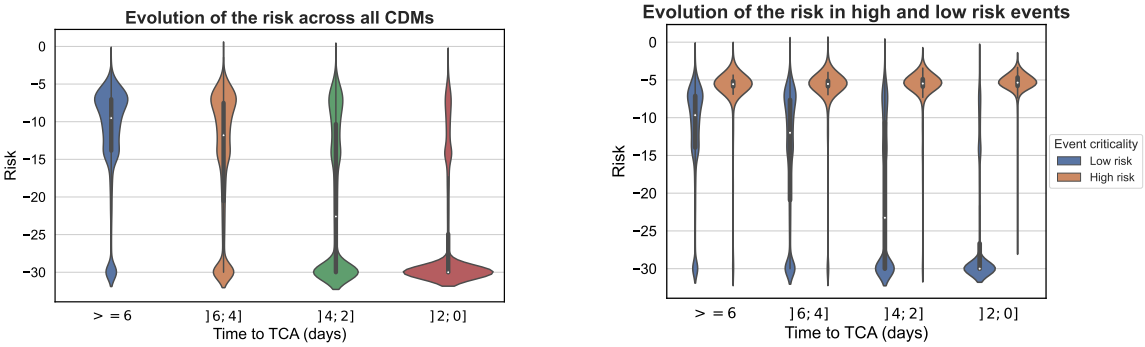


Figure 3.3: Distribution of the final risk of the events contained in the CAC dataset. The vertical dashed line marks the risk threshold above which the events are considered high-risk. It can be seen that the data is extremely imbalanced, with the large majority of the events being low-risk. It is also possible to verify a peak at the risk value –30.

The evolution of the risk across the CDMs in the dataset, in certain time intervals until the TCA, can be visualized through the violinplots of Figure 3.4.



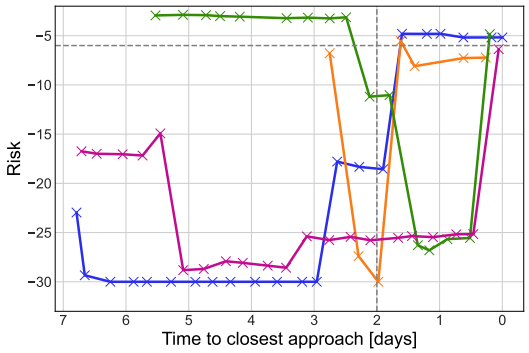
(a) Distribution of the risk across all CDMs over time, until the TCA.

(b) Distribution of the risk across all CDMs over time, until the TCA, for low and high-risk events.

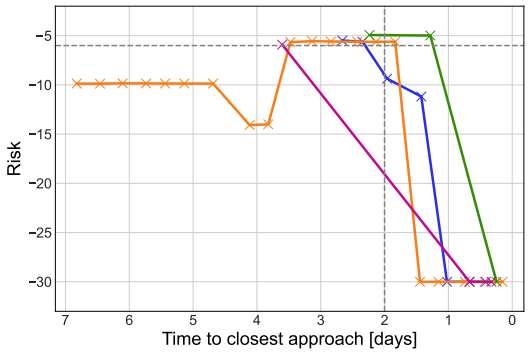
Figure 3.4: Distribution of the risk across all CDMs over time, until the TCA, for all the events in the dataset (left) and for the low and high-risk events, separately (right). It is possible to conclude that the risk tends to shift towards lower values, as the TCA gets closer, and that the high-risk events tend to start, since the initial conjunction detection, with high-risk values.

Figure 3.4a reveals that, overall, the risk tends to shift to lower values the closer to the TCA the CDMs are generated. This, again, can be explained by the refinement of the observations through time: in the first CDMs, typically released a long time before the TCA, the position of the objects has to be propagated for a larger time span which translates into an increase of the position errors that usually results in a higher collision probability value, because there is a lot of uncertainty regarding the position of the objects. Towards the TCA, the positions have to be propagated over a smaller time span which leads to smaller position uncertainties and more accurate position estimates, which usually results in a decrease of the value of the risk of collision. Figure 3.4b shows that the majority of low-risk events have starting risk values that are very close to the high-risk threshold, but over time its criticality reduces, eventually having lower risk values. On the other hand, Figure 3.4b shows that the high-risk events tend to start with high-risk values since the initial conjunction detection. Furthermore, both violinplots of Figure 3.4 reveal that, even after the cut-off time (2 days until TCA), the distribution of the risk values tends to change, which highlights the problems of making manoeuvre decisions based on preliminary knowledge and emphasizes the benefits of predicting the criticality of events after this time threshold.

The violinplots of Figure 3.4 show the overall evolution of the risk over time and may create an idea that the collision risk evolves in a predictable manner, within the events. However, this idea can be misleading. Figure 3.5 shows the time-series evolution of the risk of some events in which the collision risk updates exhibit an unpredictable behaviour.



(a) Example of events whose risk evolution shows big jumps from low to high-risk values.



(b) Example of events whose risk evolution shows big jumps from high to low-risk values.

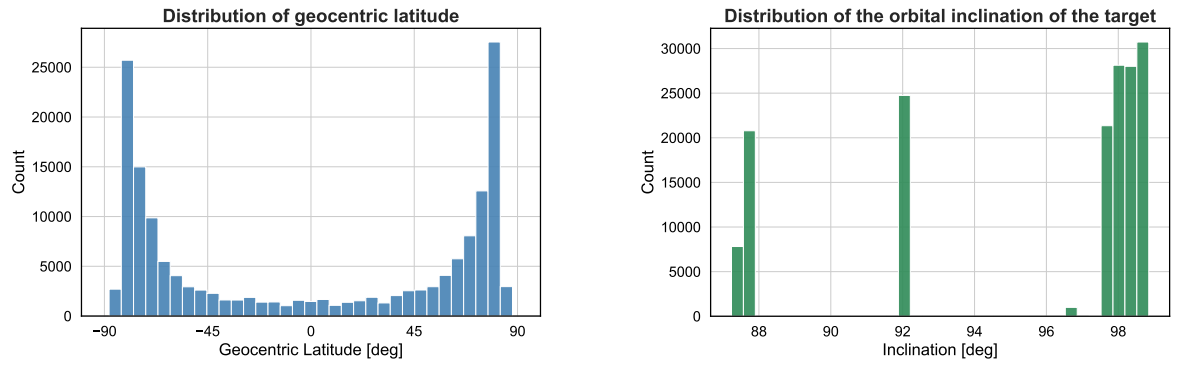
Figure 3.5: Time series evolution of the risk for some events that show an unpredictable behaviour, with extreme jumps in risk values. The risk within the CDMs of each event is marked with crosses and the colored lines represent the events. The vertical and horizontal dashed lines mark the cut off time and the high-risk threshold, respectively.

Figures 3.5a and 3.5b show that some events in the dataset, over time, experience a significant change in their risk values, with no visible trend in the collision probability evolution. On the one hand, Figure 3.5a demonstrates that there are events that experience jumps from very low to very high-risk values, contrarily to what the violinplot of Figure 3.4a suggests for the typical evolution of the collision risk - a decrease over time until the TCA. Additionally, Figure 3.5a also shows that not all high-risk events start with high-risk values since the initial conjunction detection, as the violinplot of Figure 3.4b hinted. On the other hand, Figure 3.5b shows that some events have big jumps from high-risk values to low-risk values, without a smooth transition. In both figures, it is possible to see that there are significant changes

in the risk of collision, even after the cut-off time, which further adds to the importance of applying AI methods for predicting the collision risk after this threshold since the collision probability values before the cut-off time can be misleading. The complex behavior of the evolution of the collision risk also emphasizes the difficulty of the task at hand.

3.2.3 Conjunction Location and Geometry

The collision warnings are generated when fragments orbiting Earth come close to colliding with operational satellites, so it is expected that most CDMs arise at the locations where spacecraft typically operate. Figure 3.6 represents the distribution of the geocentric latitude in which the conjunction events occur and the distribution of the orbital inclination of the target, across all CDMs of the dataset.



(a) Distribution of geocentric latitude.

(b) Distribution of orbital inclination of the target.

Figure 3.6: Distribution of geocentric latitude (left) and orbital inclination of the target objects (right), across all CDMs. It is possible to conclude that most CDMs arise from close encounters near the Earth poles, due to a higher density of active satellites in that location, which can be confirmed by the high inclination values of the orbit of the target.

Figure 3.6a shows that most conjunction events occur near the poles of Earth, with most of the CDMs containing geocentric latitude values near the $\pm 90^{\circ}$ mark. This is due to the fact that a large number of scientific satellites have a polar or near-polar orbit, in which a large variety of missions are typically performed, like Earth-mapping or weather forecasting. Polar orbits are defined as orbits in which the spacecraft passes above or nearly above both Earth poles, typically at altitudes from 200 km to 1000 km [63], and have an inclination of approximately 90° with respect to the equatorial plane of Earth (deviations up to 20° or 30° are still classified as polar orbits [63]). Figure 3.6b shows the distribution of the orbital inclination of the target objects, across all CDMs, where it is possible to confirm that the primary objects have high orbital inclinations, further adding to the fact that a large number of active spacecraft have polar or near-polar orbits, which justifies the high number of CDMs containing high absolute latitude values for the location of the close conjunctions.

Other features contained in the dataset that can give good insights into the occurrence of close encounters are the conjunction geometry parameters. The collision warnings contain the three components of the relative position and relative velocity between the chaser and the target, expressed in the target RTN reference frame, as well as the miss distance and relative speed, which can be interpreted as the magnitude of the relative position and relative velocity vectors, respectively. This analysis shall

focus on the study of the miss distance and relative speed features since they can provide more relevant and interpretable information about the conjunction events. Figure 3.7 represents the distribution of the relative speed versus the miss distance values, contained in the last CDM released for each event after the cut-off time, for high and low-risk conjunctions.

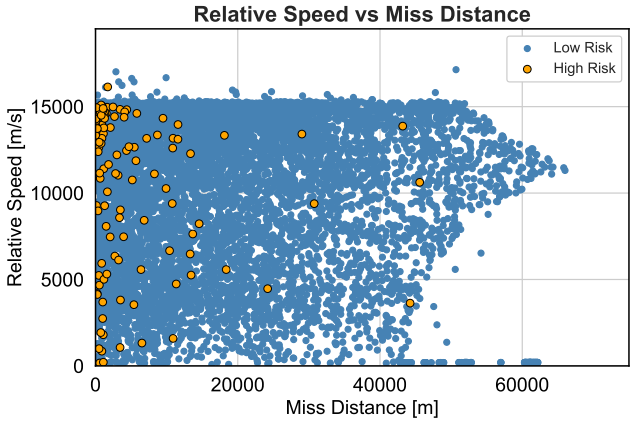
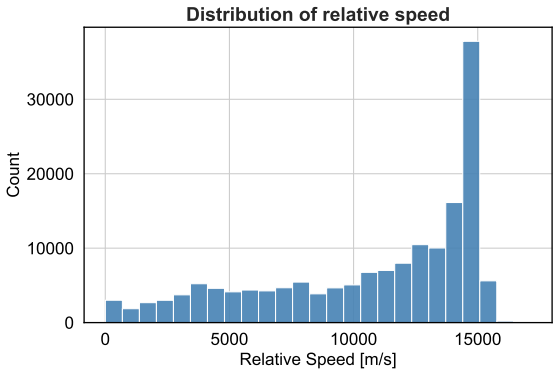
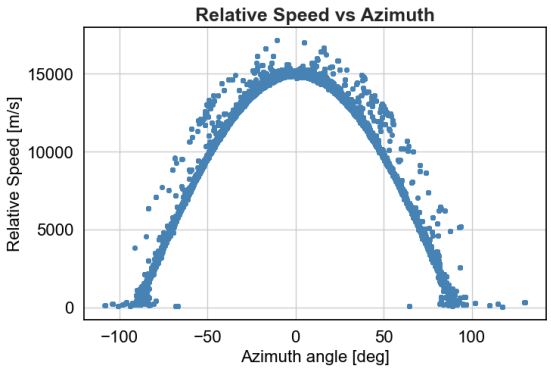


Figure 3.7: Distribution of the relative speed versus the miss distance values, contained in the last CDM released for each event after the cut-off time, for high and low-risk conjunctions. It is possible to conclude that the majority of high-risk conjunctions occur at high relative speeds and low miss distances.

Figure 3.7 shows that most high-risk events occur at low relative distances and high relative speeds. The miss distance is an extremely important parameter in risk computation and has a big influence on the criticality of the events. The smaller the relative distance at the TCA, the closer the objects are expected to come together and, hence, the higher the risk of collision. To better understand the influence of the relative speed on conjunction events and on the conjunction geometry, Figure 3.8 presents the distribution of the relative speed and the relation between the relative speed and the azimuth angle.



(a) Distribution of the relative speed.



(b) Distribution of the relative speed against the azimuth angles.

Figure 3.8: Distribution of the relative speed (left) and distribution of the relative speed values against the azimuth angle (right), across all CDMs of the dataset. It can be concluded that most CDMs arise from conjunction events in which the objects have high relative speeds, which are associated with low azimuth angles, meaning that both objects approach at nearly head-on trajectories.

Figure 3.8a demonstrates that the majority of CDMs contain high relative speed values, showing that the conjunction events tend to happen between two objects with high relative velocities. In addition, another interesting observation can be made: a large number of CDMs contain relative speed values

close to 15 km/s which is around twice the value of the speed at which LEO resident space objects typically orbit, which is 7.8 km/s [64]. This suggests that the target and the chaser approach each other almost on head-on trajectories. The relative speed is the sum of the speed of both objects, thus, those cases yield a result that is twice the speed of LEO space objects. Figure 3.8b shows the relation between the azimuth angle of the relative velocity vector and the relative speed. The azimuth is the angle between the relative velocity vector and the target velocity vector in the along-track and normal plane [23], so an angle of 0° indicates that the two vectors have the same direction (because the elevation angle, which is the angle between the relative velocity vector and the target velocity vector in the along-track and radial plane, has values near zero, in all CDMs). This means that the target and chaser are on head-on trajectories, consequently leading to higher relative speeds. This can be confirmed through the plot of Figure 3.8b, which shows that high relative speeds are associated with low absolute values for the azimuth angle, meaning that most CDMs in the dataset arise from events in which the target and chaser are on opposite orbital directions. Collisions at such speeds involve a lot of impact energy and can lead to the complete destruction of a spacecraft, highlighting the importance of collision avoidance procedures.

3.2.4 Object properties

Each issued CDM contains, in addition to the risk and conjunction parameters, information about the properties and characteristics of the target and chaser objects. To not exceed a reasonable scope of this work, only the most relevant information regarding these parameters is briefly presented.

The chaser object can have multiple origins and, hence, can be of multiple types. Figure 3.9 shows the distribution of the chaser object types in the CAC dataset.

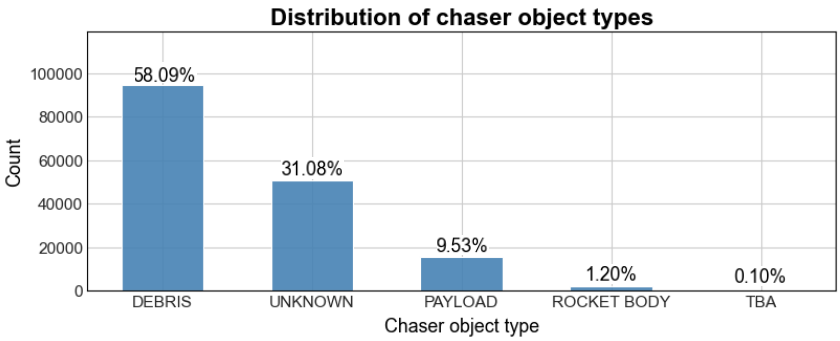


Figure 3.9: Distribution of the chaser object types.

It can be seen that, as expected, the majority of the chaser objects are classified as debris. However, a large number of conjunction partners are of unknown origin and type, which means that most of their characteristics and properties are also not known. In fact, all the CDMs that are originated by a conjunction event in which the chaser object is classified as *unknown* have missing data. The low percentage of the object type *rocket body* is likely due to the fact that the detachment of the rocket stage from the spacecraft happens in low orbital altitudes, where there are not as many active satellites as in higher altitudes and, consequently, there are fewer close encounters. However, it is important to note that these

rocket stages can carry residual fuel that can give origin to on-orbit explosions, generating numerous fragments that are spread in various directions, making this type of object extremely dangerous from a space safety point of view. The type *TBA* refers to objects whose type is still to be assessed.

Regarding the physical properties and characteristics of the objects, one of the most relevant parameters is their size. As mentioned in Subsection 2.3.3, to compute the collision probability between two resident space objects, both the chaser and target are considered spherical objects, with radius denoted by R_c and R_t, respectively. The size and radius of the target are assumed to be known with reliable precision since the target is an active spacecraft. However, the chaser objects have an arbitrary shape and their radius is usually not known, hence, it has to be estimated. These estimations depend on a lot of factors and, often, cannot be obtained with desired precision. In real-life operations, when it is not possible to provide accurate estimates of the size of the chaser object, a default value of $R_c = 2$ m is established. Table 3.2 shows a descriptive analysis of the chaser and target object radius. Although there are millions of pieces of fragments in space with sizes smaller than 10 cm [1], the minimum value of the radius of the chaser object contained in the CDMs is the default value of 2 m. This is likely because it is extremely difficult to obtain reasonable estimates of objects with very small sizes with unknown shapes orbiting Earth at very high speeds. So, it is safer to assume a minimum value of 2 m. In addition, it can be seen that there are CDMs that contain values for the chaser radius almost ten times greater than the 95-th percentile for that parameter, translating into values similar to the radius of active spacecraft. At first, these values for the chaser radius may seem outliers, but, after a closer look into the dataset, it was verified that the CDMs containing these values for R_c are issued in events in which the chaser object is classified as *payload*. This type of object can refer to the equipment carried on spacecraft, or part of satellites that no longer serve a useful function and can be, for example, solar panels, that have a very large size, which justifies the large values for R_c contained in the dataset.

Parameter	min	5-th percentile	median	95-th percentile	max
Chaser Radius	2.0	2.0	2.0	3.6	30.5
Target Radius	0.1	1.0	9.1	21.0	27.73

Table 3.2: Descriptive analysis of the object radius of the target and chaser. All the values contained in the table are expressed in meters.

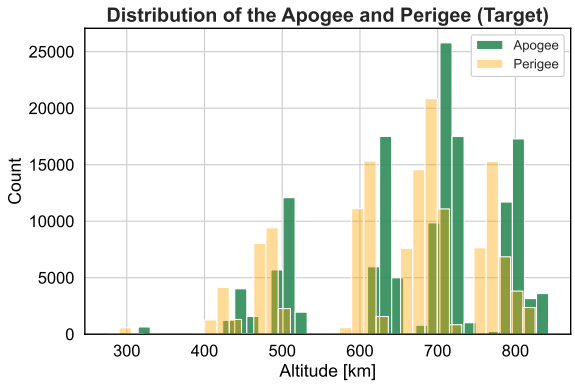
Other two important parameters, contained in the dataset, are the ballistic parameter and the energy dissipation rate. The ballistic parameter of a spacecraft is a measure of its ability to overcome air resistance caused by the atmosphere and is given by $B = c_D A/m$, where A is the cross-sectional area in the direction to the direction of the spacecraft motion, m its mass and c_D the drag coefficient [65]. The higher the ballistic parameter, the more sensitive is the spacecraft to the air resistance in orbit. The energy dissipation rate is the parameter that determines the rate at which the orbital energy is dissipated. A closer examination of the dataset showed that there are CDMs that contain negative values for the ballistic parameter and energy dissipation rate of the target and chaser, which is physically not possible. In addition, it was seen that both parameters contain outliers, especially for the chaser object.

3.2.5 Orbital elements

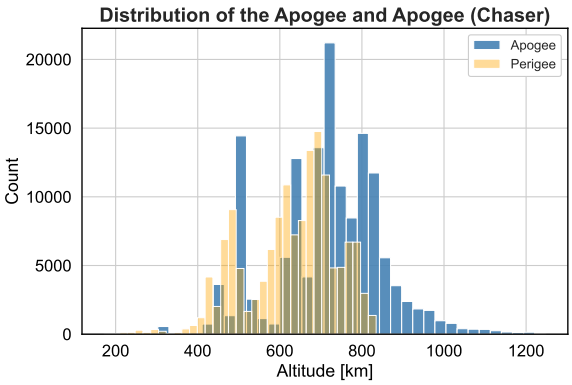
The orbital elements of the target and chaser contained in each CDM are the following: inclination, perigee, apogee, eccentricity, and semi-major axis. These parameters describe and identify the orbit of space objects and, hence, can provide a lot of relevant information regarding the context of the problem. To capture and visualize the distribution of the orbital parameters in a compact and clear way, only the values up to the 95-th percentile are considered, in the following plots of this Section.

Firstly, the inclination is the angle between the orbital plane of an object and the equatorial plane of the Earth and, as it was previously verified when discussing the conjunction locations, most primary satellites in the dataset contain high inclination values. Regarding the chaser, 90% of the CDMs contain inclination values between 65° and 99° , highlighting the fact that most of the events occur between objects with high orbital inclinations and, again, justifies the high number of conjunctions taking place near the polar regions.

Secondly, the perigee and apogee are the points in the orbit of a space object at which the object is nearest and furthest to the Earth, respectively. These two points are known as *apsis*. Figures 3.10a and 3.10b show the distribution of the perigee and apogee altitudes of the target and chaser orbits. It is important to note that, in the dataset, the values of the apogee and perigee are measured from the surface of Earth, and, to obtain these quantities measured from the center of Earth, it is necessary to sum the radius of Earth (it is considered that the Earth is a perfect sphere).



(a) Distribution of the perigee and apogee altitudes for the target.

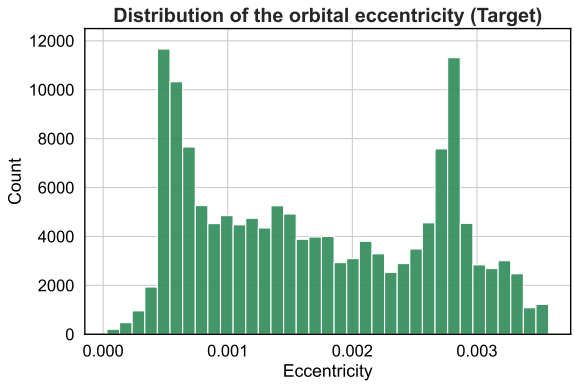


(b) Distribution of the perigee and apogee altitudes for the chaser.

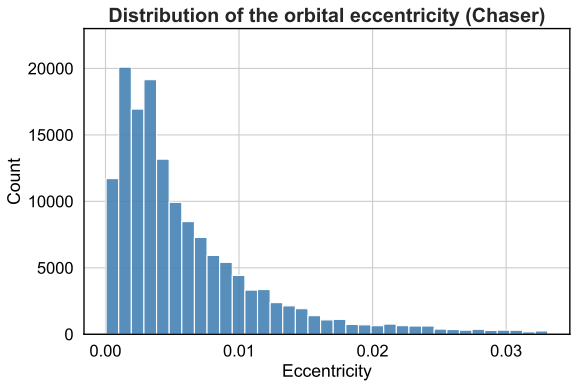
Figure 3.10: Distribution of the perigee and apogee altitudes for the target (left) and chaser (right).

Figure 3.10 shows that the objects have, mostly, orbital altitudes between 200 km to 1 000 km, which are typical altitudes for spacecraft with polar orbits [63]. Furthermore, it can be seen that the perigee and apogee have similar distributions for both the target and chaser, suggesting that the objects may have a circular (or near-circular) orbit. This can be confirmed through Figure 3.11, which shows the distribution of orbital eccentricity contained in the CDMs. The orbital eccentricity is a measure of how much the orbit of an object deviates from a circle. It is a dimensionless parameter, in which a value of 0 represents a circular orbit, values from 0 to 1 represent an elliptic orbit, a value of 1 denotes a parabola, and values greater than 1 represent a hyperbola. Figure 3.11 shows that the majority of collision warnings contain eccentricity values very close to zero, which indicates that the objects follow circular or near-circular

orbits.



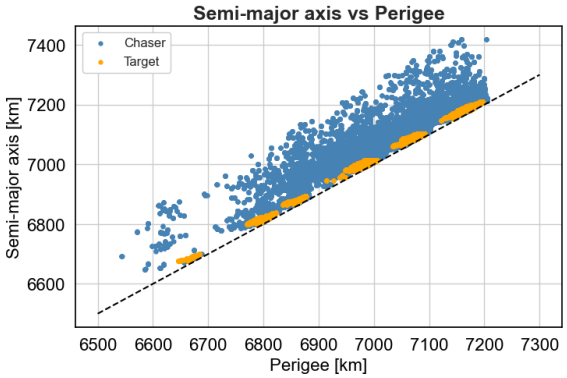
(a) Distribution of the eccentricity for the target.



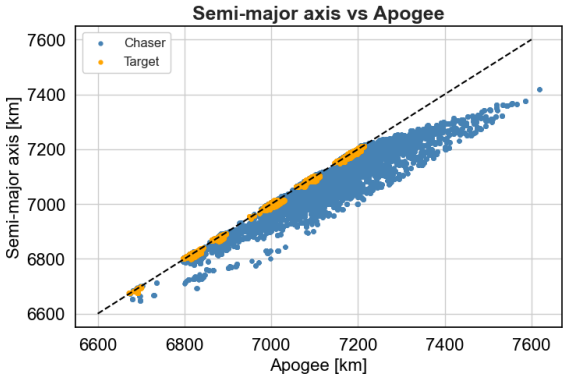
(b) Distribution of the eccentricity for the chaser.

Figure 3.11: Distribution of the eccentricity for the target (left) and chaser (right). The distributions of the eccentricity for both objects are very close to zero, indicating that the objects have circular (or near-circular) orbits.

Lastly, the semi-major axis is the distance from the center of the orbit of a spacecraft to the longer end of the orbit, and, for circular orbits, it is equal to the orbit radius. In the CAC dataset, most objects follow near-circular orbits, so the semi-major axis values are expected to be very close to the values of the perigee and apogee contained in each collision warning. This can be confirmed by Figure 3.12, which shows that, in each CDM, the values of the semi-major axis are close to the apsis values (measured from the center of Earth), for both the target and chaser. The semi-major axis values of the chaser have a larger deviation from the perigee and apogee values, because, as can be verified from the difference in scale between Figures 3.11a and 3.11b, the eccentricity values of the chaser have a larger deviation from the value zero, meaning that the orbit is not completely circular and, consequently, the semi-major axis will be different from the apsis values. The fact that the orbits are not purely circular also justifies the fact that the perigee values are lower and the apogee values are greater than the semi-major axis because the center of the orbit is not exactly at the center of the Earth.



(a) Distribution of the semi-major axis values vs the perigee.



(b) Distribution of the semi-major axis values vs the apogee.

Figure 3.12: Distribution of the semi-major axis values vs the perigee (left) and apogee (right), for both the target and chaser. The dashed lines mark the values where the semi-major axis is equal to the apsis values and it can be seen that, in each CDM, the semi-major axis values are close to the perigee and apogee, indicating that the objects follow circular or near-circular orbits.

3.2.6 Estimation Errors

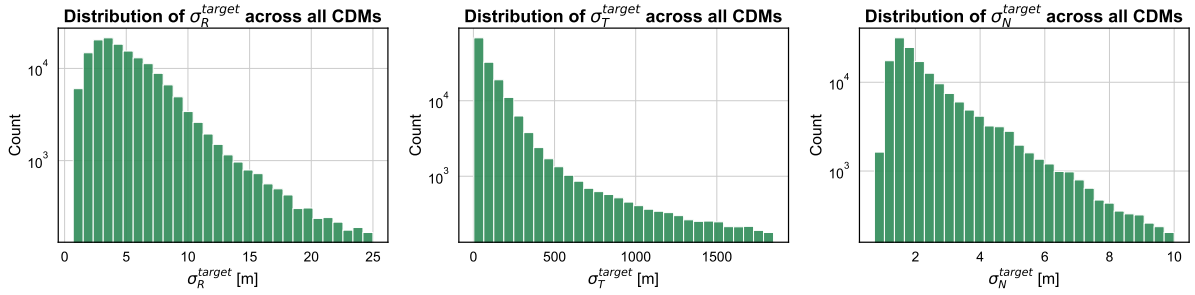
As explained in Section 2.2, the orbital state of resident space objects cannot be determined with complete accuracy and, consequently, is subject to errors. A statistical estimation of these errors is given in the form of covariance matrices, in which the diagonal entries are defined by the standard deviations and the off-diagonal entries are defined by both the standard deviations and correlation coefficients. Since the risk equation only uses the position uncertainty components of the error estimation, the position errors are the focus of this Section. The position standard deviations express the amount of uncertainty in the propagated positions of a spacecraft in the three components of the error ellipsoid and the correlation coefficients define its orientation in space. Although the correlation coefficients are relevant for the risk computation, it is assumed that they don't contribute as much as the position standard deviations, which define the "size" of the error ellipsoid of the objects. Therefore, this analysis shall focus solely on the position standard deviations of the objects, which are considered the main contributors to the collision probability.

The following nomenclature for the position standard deviations of the target and chaser shall be used hereafter:

- Chaser position errors - σ_R^{chaser}, σ_T^{chaser} and σ_N^{chaser};
- Target position errors - σ_R^{target}, σ_T^{target} and σ_N^{target},

where the subscripts R, T, and N denote the radial, along-track, and normal components of the position errors, respectively, and the superscripts *target* and *chaser* represent the object that the parameter is referred to.

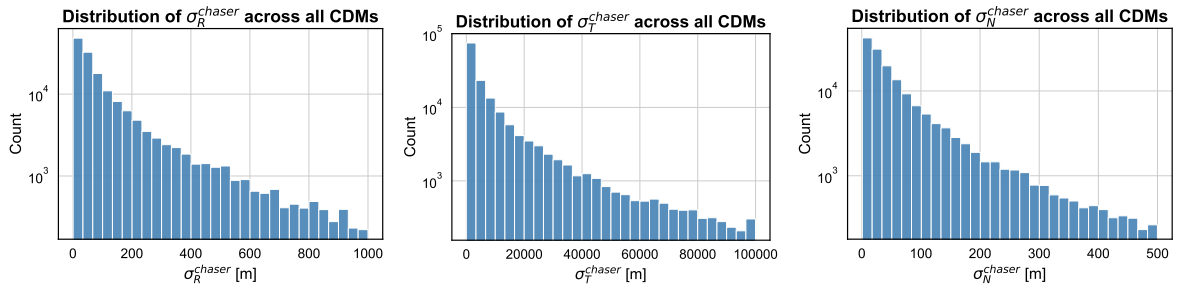
Figure 3.13 and Figure 3.14 show the distribution of the three position uncertainties of the target and chaser, respectively, in their own RTN coordinates. Due to the imbalanced and uneven distribution of the parameters, a logarithmic scale on the y-axis is used.



(a) Radial component. (b) Along-track component. (c) Normal component.

Figure 3.13: Distribution of position uncertainties of the target. To be able to visualize the distribution in a clear way, only the values up to the 95-th percentile are considered. The y-axis is in logarithmic scale.

As explained in Section 2.2, the along-track component of the position error is subject to a higher degree of uncertainty due to the perturbations that affect the mean motion of the object and cannot be exactly simulated by current propagators. This can be confirmed by Figures 3.13 and 3.14, in which the difference in scale of the propagated standard deviations to the TCA shows that the along-track component of the position errors has a considerable higher magnitude than the radial and normal components,



(a) Radial component. (b) Along-track component. (c) Normal component.

Figure 3.14: Distribution of position uncertainties of the chaser. To be able to visualize the distribution in a clear way, only the values up to the 95-th percentile are considered. The y-axis is in logarithmic scale.

for both the chaser and target, and the position errors on the radial component have the second highest uncertainty value. In addition, it is possible to verify that the position errors of the chaser object are significantly higher than the position uncertainties of the target. This can be justified by the fact that there is a lot more available information about the primary objects (active spacecraft) and, consequently, their orbit determination is more precise which leads to better location estimates and lower position uncertainties. As previously seen, the chaser object can be a piece of debris, payload, or rocket body with varying shapes and physical properties, and, in addition, a large percentage of the secondary objects are classified as *unknown*, which means that their characteristics are also unknown. Due to the lack of available information about the chaser objects, the orbit determination of the chaser is not as accurate as the position estimates of the target, which can explain the higher position uncertainties.

To better understand the distribution of the position uncertainty parameters, Table 3.3 presents a descriptive analysis of the position errors of the target and chaser.

Object	Parameter	min	5-th percentile	median	95-th percentile	max
Chaser	σ_R^{chaser}	0.001	0.008	0.068	1.078	63781.363
	σ_T^{chaser}	0.005	0.138	3.965	110.748	76548.834
	σ_N^{chaser}	0.001	0.004	0.04	0.458	63781.363
Target	σ_R^{target}	0.001	0.002	0.005	0.016	63781.363
	σ_T^{target}	0.003	0.011	0.086	1.856	63781.363
	σ_N^{target}	0.001	0.001	0.002	0.007	63781.363

Table 3.3: Descriptive analysis of the position errors of the target and chaser. All the values contained in the table are expressed in kilometers.

Table 3.3 shows that all the position uncertainties have a minimum value greater than zero, which, within the scope of this work, is extremely important, since the position standard deviations define the diagonal entries of covariance matrices, that have to be positive semi-definite. It is also possible to conclude that these features contain extreme outliers, with some CDMs having values for the standard deviations greater than ten times the Earth radius, which is approximately 6 371 km. This is likely due to orbit determination or measurement errors. In fact, in real-life operations, when it is not possible to estimate the position errors of the resident space objects, a default value of 63 781.363 km is given to the position standard deviations. In addition, through the analysis of the 95-th percentile of each

parameter, it can be verified that the along-track component of the chaser is by far more dominant than all the other position uncertainties of both objects and, hence, it can be seen as the key factor in the collision probability estimation.

Recalling that the CDMs contain all its parameters propagated to the TCA, the magnitude of the position uncertainties contained in the time series of collision warnings of a given event is expected to decrease with each new update. As explained in previous chapters, perturbation and propagation models cannot exactly simulate all the perturbing forces acting on resident space objects and, consequently, if the position of the objects has to be propagated over a large time span, there will be a continuous miss modeling of the orbital state of the objects, which leads to larger position uncertainties. This can be confirmed through the inspection of the violinplot of Figure 3.15, which represents the evolution of the along-track position error of the chaser object (which is considered the key factor in risk estimation) through time until the TCA, showing that the distribution of the along-track position error shifts towards lower values over time.

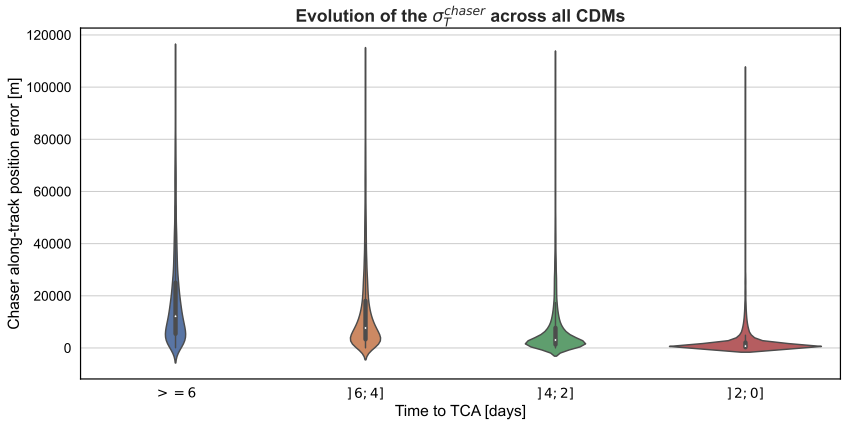


Figure 3.15: Distribution of the along-track standard deviation of the chaser object, across all CDMs, over time. Only the CDMs containing the values up to the 95-th percentile are considered for this viloinplot. As expected, it can be seen that the position uncertainties shift towards lower values, the closer to the TCA the CDMs are generated.

Chapter 4

Methodology

4.1 Problem Framing

In collision avoidance, it is extremely important to identify high-risk events in order to prevent catastrophic collisions between space objects that can damage operational spacecraft and endanger the whole space population due to the number of fragments that are further released into orbit. Predicting low-risk events as high-risk can lead to unnecessary avoidance maneuvers and waste of energy, but failing to identify high-risk events can give false confidence to spacecraft operators about the outcome of a given close encounter, which can result in a wrong decision regarding the performance of an avoidance maneuver and, consequently, lead into a collision. Thus, the main goal of all methods for collision avoidance is to correctly and accurately predict the criticality of high-risk conjunction events.

The analysis of both datasets shows that the data is extremely imbalanced, with only a very small percentage of the events having a risk higher than −6 in the final released CDM, which constitutes one of the major challenges of applying AI for the collision avoidance problem. In addition, in Chapter 3, it was seen that the evolution of the risk within the events can have complex and unpredictable transitions, highlighting the difficulty of using the risk of collision as the prediction target. These problems encouraged Metz [23] to explore another approach: using the position uncertainties as the prediction target and then recomputing the probability of collision with Akella's and Alfriend's formula. With this approach, one does not solely depend on the rare occurrence of high-risk conjunctions and can explore the most important component of the computation of the risk: the covariance matrix that represents the position errors. As mentioned in Section 2.2, current propagation models are not perfect and fail to exactly simulate some of the perturbations forces that can influence the orbit of an object in space, like the drag and gravity perturbations. If the orbit of the objects could be exactly simulated, there would be no need to assess the probability of collision between two space objects, because their future position could be exactly computed. Hence, the position uncertainties can be seen as a key factor in collision avoidance, and being able to correctly predict them can have tremendous advantages in collision risk assessment. The work done by Metz [23] showed that this is a solution that can outperform the naive forecast and should be further explored. Therefore, in this work, in addition to directly predicting the risk

of collision between two space objects, the position covariances will also be used as prediction targets, using HMMs. A more detailed explanation of the latter approach is given next.

Akella's and Alfriend's formula to compute the collision probability, which is described in Section 2.3.3, can be decomposed into the following parameters at the time of closest approach: (i) the relative position between the objects, (ii) the combined covariance matrix of the target and chaser and (iii) the hard body radius. However, the datasets used in this work do not directly contain some of these parameters and, to obtain them, the following features are needed for both the target and chaser:

- the position coordinates;
- the standard deviations that describe the position error;
- the correlation coefficients;
- the radius of the sphere that encapsulates the object.

Additionally, the velocity vectors of both objects are also needed, in order to transform the covariance matrices into the same reference frame (a detailed explanation is provided in the end of this Section).

In this work, only the time series of the position errors will be modeled, in order to explore its influence in the final risk prediction and, hence, all the other necessary parameters for the computation of the collision probability are going to be taken from the last released CDM before the cut-off time. In addition, just like in Metz's work [23], other approximations will be made. As previously mentioned, there is much more available information regarding the target object (operating spacecraft), which leads to more accurate orbit determination and propagated position estimates with lower uncertainties, as it was seen in Section 3.2.6. Hence, in this work, it is assumed that the propagated orbital states of the target (including its position uncertainties) at the TCA are fairly precise and do not require the application of ML algorithms, so the values of the orbital states of the target are going to be taken from the last released CDM before the cut-off time. Furthermore, the correlation coefficients define the orientation of the error ellipsoid in space and it is assumed that it has a smaller influence on the collision probability when compared to the standard deviations, which represent the "length" of the components of the position errors. So, it is also considered that the last CDM received until the cut-off threshold offers a sufficiently good approximation of the correlation coefficients. In summary, only the three positional standard deviations of the chaser object, defined in its RTN reference frame, are chosen as the prediction targets: σ_R^{chaser}, σ_T^{chaser}, and σ_N^{chaser}, in which the subscripts R, T and N represent the radial, along-track, and normal components, respectively. Figure 4.1 illustrates this approach.

After predicting the positional standard deviations of the chaser, the covariance matrix must be assembled, then it has to be combined with the covariance matrix of the target, and the result must be given as input to the collision probability computation formula. However, at this point, a problem arises. In both the CAC and Neuraspace datasets, the position errors of the target and chaser are given in each object RTN reference frame, so the respective covariance matrices are defined in different coordinate systems and cannot be summed directly. In this work, this is solved by rotating the chaser's covariance matrix into the RTN frame of the target. Section 2.2.2 explains all the steps to perform this transformation with greater detail, but it is important to note that the rotation matrices that allow this transformation

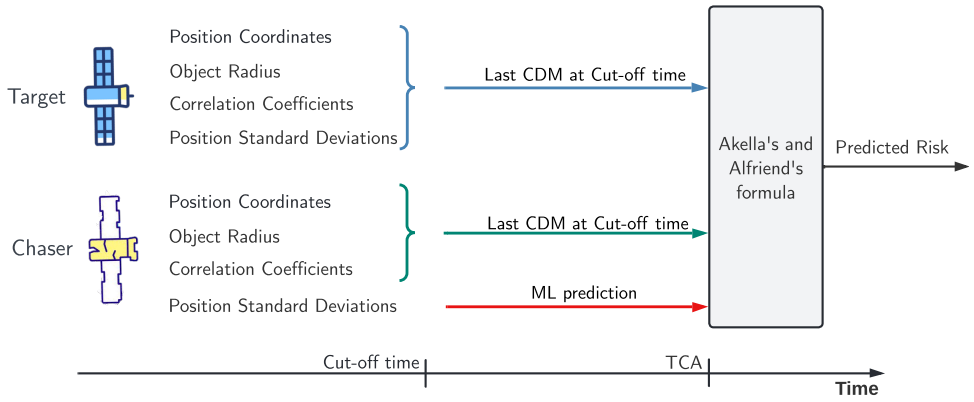


Figure 4.1: Illustration of one of the approaches followed in this work: the chaser position standard deviations are predicted using ML models and the rest of the parameters are taken from the last released CDM, before the cut-off time. Then, the risk is re-computed using Akella's and Alfriend's formula.

are based on the position coordinates of the target and chaser and velocity vectors at the TCA which are parameters that are not available in the public dataset. The CAC dataset is an anonymized dataset released by ESA that was manipulated in order to protect confidential and classified information and, hence, some features were kept private, like the position of the spacecraft. Thus, it is not possible to use the dataset of the CAC to predict the chaser position errors and then compute the final risk of collision. On the other hand, the Neuraspace dataset is a collection of CDMs that contains real-world raw data in which all the features are available and can be used to explore this approach.

In summary, this work presents two approaches for predicting the criticality of each event:

- **Approach A**: directly model and predict the risk of collision contained in the CDMs.
- **Approach B**: model the chaser uncertainties and use the predictions to compute the collision probability.

The nomenclature **Approach A** and **Approach B** shall be used hereafter in order to identify and distinguish the two methods used in this thesis.

4.2 Proposed Models and Baseline Solution

In this section, the models used for the two approaches followed in this work are described, as well as the baseline model.

4.2.1 Baseline

To understand the effectiveness of the predictions given by the models, it is important to establish a baseline solution. The baseline used to compare each model, in this work, is one of the proposed in the Spacecraft Collision Avoidance Challenge [17]: use the last known observation as the baseline prediction. In other words, the baseline solution for a given parameter is the value of that parameter that is contained in the last released CDM before the cut-off time.

Regarding the classification problem of identifying high-risk and low-risk events (two categories), the baseline solution is obtained by defining a threshold in the baseline risk prediction: if the baseline risk solution is equal to or greater than −6, the event is predicted as a high-risk event; otherwise, it is predicted as a low-risk conjunction. This was the threshold used during the CAC [17].

During the data analysis, it was verified that the large majority of the conjunction events have a final risk value of −30, for both the CAC and Neuraspace datasets. Figure 4.2 shows the distribution of the true risk values of the events that the baseline predicts as having a final risk of −30, for both the datasets, after the removal of CDMs containing extreme outliers and parameters with impossible values.

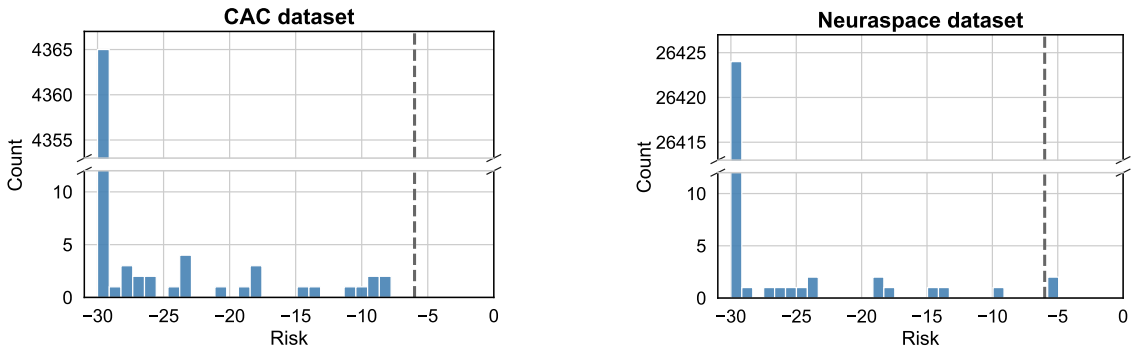


Figure 4.2: Distribution of the true risk values of the events that the baseline predicts as having a final risk of −30, for the CAC dataset (left) and for the Neuraspace dataset (right). The vertical dashed lines mark the high-risk threshold

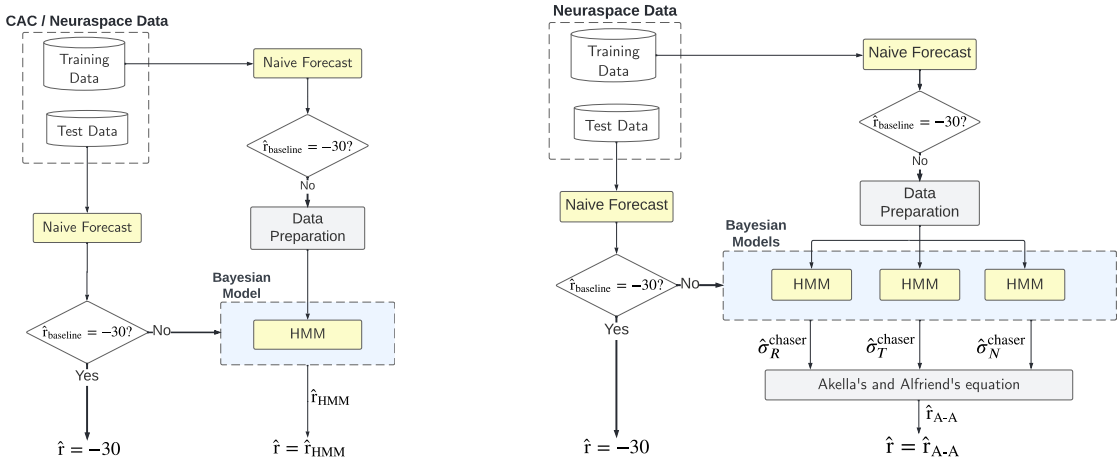
It is possible to conclude that, in most cases, when the baseline solution for a given event is −30, the event contains in its last released CDM a risk value of −30. In fact, in the CAC dataset, 99.34% of the naive forecast −30 predictions are correct and, in the Neuraspace dataset, 99.93% of these predictions are events that have a final minimum risk value. In addition, in the CAC dataset, there are no miss-classifications of high-risk conjunctions and, in the Neuraspace dataset, there are only two. Hence, in this work, it is assumed that the −30 predictions by the naive forecast are trustworthy and, consequently, it is considered that those events don't require the application of ML models. Thus, the following approach will be tested in this work: the events that the baseline predicts as having a final risk of −30 will not be used for training and, during the test phase, these events will not be given as input to the trained models and will be directly predicted with a final risk of −30. With this approach, a significant amount of low-risk conjunctions are removed (4 391 events in the CAC data and 26 439 events in the Neuraspace dataset), which can help deal with the data imbalance problem and, consequently, allows a better analysis, by the ML models, of the evolution of the parameters of events that end up having higher risk values. It also reduces the volume of training data, resulting in lower memory requirements and a lower computational time. However, it is important to note that this method is not perfect, since in the Neuraspace dataset there are two high-risk conjunctions that are predicted as having the minimum risk value, which, again, highlights the difficulty of predicting the risk of collision between resident space objects, because it shows that the risk contained in the last CDM received before the cut-off time (which is considered the best knowledge about the outcome of the event before the decision threshold) can be very misleading. Due to the inherent advantages of not using the events with a baseline forecast of −30

in the ML models, this approach is followed in this thesis, but future work should tackle the previously described problem and a better model that can detect the –30 events without any miss classifications of high-risk conjunctions should be developed.

4.2.2 Models for the risk and covariance evolution

As explained in the previous Section, only the events in which the baseline solution shows a final risk different from –30 will be used as training data, to infer the parameters of the models. In addition, the main goal of this work is to establish a benchmark solution for the use of Bayesian HMM in predicting the risk of collision between two resident space objects, and, in this thesis, this is performed by each implemented model only learning the evolution of one single feature of the dataset, i.e., in **Approach A**, only the risk sequences contained in the CDMs will be analyzed by the HMM and, in **Approach B**, three separate HMMs will be used to learn the evolution of the three components of the position errors of the chaser object: the radial, along-track and normal. This way, this thesis provides a foundation for future research regarding the implementation of HMMs, with Bayesian statistics, to the challenge of applying AI in collision avoidance. Regarding the prediction phase, the events whose risk is predicted as being higher than –30 by the naive forecast go through the prediction process of the implemented models and the other events are directly predicted as having a final risk of –30.

A schematic representation of the learning and prediction process is presented in Figure 4.3, for both the approaches followed in the proposed work, where $\hat{r}_{baseline}$ and $\hat{r}$ denote the baseline and final model risk predictions, respectively, and the circumflex accent above the other variables indicate their predicted values.



(a) Learning and prediction procedure of **Approach A** . (b) Learning and prediction procedure of **Approach B** .

Figure 4.3: Schematic diagram of the learning and prediction procedure used for **Approach A** (left) and **Approach B** (right).

The area shaded in blue in Figures 4.3a and 4.3b represents the Bayesian HMMs that need to be specified and that receive the cleaned and prepared data to learn the evolution of the risk and position error within the events. Chapter 5 describes the preprocessing steps of the original dataset and Chapter 6 focuses on the definition and implementation of the Bayesian models used in this work.

Chapter 5

Data Preparation and Setup

This chapter describes the preparation of the CAC and Neuraspace datasets, taking into consideration the analysis of both datasets and the methodology followed in this work, which is described in Chapter 4. This chapter is divided into two sections: Section 5.1 describes the data cleaning that was performed in the datasets and Section 5.2 presents the data setup for the training and testing of the implemented models.

5.1 Data preparation

As previously explained, this work benchmarks the use of HMMs in predicting the risk of collision between two resident space objects only by evaluating the evolution of one single feature in each implemented model, meaning that a total of four features (the risk and the three components of the position standard deviations of the chaser) are used for training. However, it is extremely important to remove the CDMs that may contain errors or extreme outliers also in other features, because the other parameters can affect the computation of the probability of collision and can result in untrustworthy and misleading observations. It is important to note that, in this work, the datasets were kept as close to the original as possible, in order to benchmark the performance of Bayesian HMMs with data that is representative of real collision avoidance missions.

Before the data cleaning, some feature engineering is performed in the Neuraspace dataset: the feature *risk* is created (which is the base 10 logarithm of the probability values) and the positional covariances were transformed into standard deviations and correlation coefficients. In addition, in the Neuraspace dataset, the maneuvered events and the CDMs with the originator *Minicat* are removed. The rest of the data preparation is very similar for both datasets, in which the steps are described hereafter.

The first step is to eliminate CDMs that contain physically impossible values or extreme outliers. Some CDMs in both datasets contain negative values for the ballistic coefficient or the energy dissipation rate, which is physically impossible and, hence, those collision warnings are eliminated. The Neuraspace dataset also contains CDMs with negative positional variance values, which are removed. In addition, it

was verified that some parameters contain extreme outliers, like the position standard deviations of both objects, that have a maximum value greater than ten times the radius of the Earth. Accianiri et al. [66] also identified this problem and considered the following values as the upper thresholds for the position standard deviations:

- 20 m, 2 000 m and 10 m for σ_R^{target}, σ_T^{target} and σ_N^{target}, respectively;
- 1 000 m, 100 000 m and 450 m for σ_R^{chaser}, σ_T^{chaser} and σ_N^{chaser}, respectively.

After a closer look into Table 3.3, presented in the analysis of the position errors of the objects in the CAC dataset, in Subsection 3.3, it is possible to conclude that these values are very close to the 95-th percentile of each parameter. Although the 95-th percentile of the position standard deviations contained in the CDMs of the Neuraspace dataset are slightly below these upper thresholds, these values are assumed to be reasonable and the CDMs containing position errors above those values are removed.

The second step is only to consider the parameters of each CDM that are used for risk prediction. For **Approach A**, only the risk contained in each collision warning is considered, and, for **Approach B**, the three chaser position errors and the other necessary parameters to recompute the collision probability (explained in Section 4.1) are considered. Then, the collision warnings containing missing data in the necessary features to compute the risk (and in the *risk* feature itself) are removed, which leads to the removal of 655 CDMs (0.118%) in the Neuraspace dataset and does not remove any collision warning in the CAC data. The final step is to eliminate all the events that do not satisfy the CDM constraints described in Subsection 3.2.1. Thus, all the events that do not contain at least two CDMs, one released at least 2 days before TCA and one released within 1 day to the TCA, shall be removed.

The data preparation described in this Section results in a total of 5 917 events and 44 399 CDMs being removed from the CAC dataset, which ends up with 7 187 (99.3%) low-risk events and only 50 (0.7%) high-risk conjunctions. Regarding the Neuraspace dataset, 27 683 events and 1 064 451 collision warnings are removed during the data preparation phase (note that most of the CDMs are removed when selecting the collision warnings of the 18 SPCS), resulting in 27 449 (99.89%) low-risk events and only 29 (0.11%) high-risk conjunctions. Dealing with such imbalanced data poses to be the major challenge of this work and highlights the difficulty of using the risk as the prediction target. In addition, the imbalance of the data is even more evident in the Neuraspace dataset which emphasizes the challenges of using real data.

5.2 Data setup

After the data cleaning, the CDMs of each event are arranged in descending order regarding the time to the predicted TCA and, hence, each event can be viewed as a collection of observations indexed in time order. The datasets are then split, in a stratified way (to preserve the same proportion of samples of each class), into train and test sets. For the CAC dataset, a ratio of 80:20 is used and for the Neuraspace dataset, since it contains more data, a 75:25 split is performed. The test set is only used at the end to evaluate the performance of the final model and, in each test event, only the CDMs released before the

cut-off time can be used as input of the models, in order to simulate real-life operations, in which ML algorithms must predict the risk of collision with the available information until 2 days to TCA. The training set is used to infer the parameters of the HMMs and, as previously described, only the events that show a baseline risk different from –30 are used to train the models. However, at this point, a challenge arises. Recall that, to obtain the inferred parameters, the likelihood distribution must be evaluated. More specifically, current MCMC samplers require the evaluation of the log-likelihood density at each set of observations for each proposed set of parameters θ to be sampled. But, each event has a different number of CDMs, hence, to obtain the log probability of the model, it would be necessary to separately compute, in a loop, the logarithm of (2.21) for every set of observations of each event and then sum the result to obtain the joint log probability of the model. This would make the training of the model extremely slow and inefficient since this process would have to be repeated for every θ to be sampled.

A solution for this problem is to vectorize the sequences of observations and compute the log-likelihood density for each sequence at once and then sum the result, but this is only possible if the sequences have the same length, i.e., if the events have the same number of CDMs. So, an approximation must be done regarding the data setup of the training set, in order to allow the vectorization of the input sequences. Typically, in real collision events, 3 CDMs are released per day [17] during the week leading up to the TCA, where the latest CDM available is always considered the best knowledge about the outcome of the close approach. Thus, an approach to ensure that all input sequences (events) have the same number of observations (CDMs), is to verify whether 3 CDMs are received each day and, if not, the latest CDM received is repeated until there are 3 on that day. If there are no CDMs received previous to that day, the first observation received is repeated. This process is done for all days during the week leading up to the TCA and, after this, the events that don't match the highest number of observations are, again, manipulated by repeating the first released CDM. This data setup process is schematized in Figure 5.1.

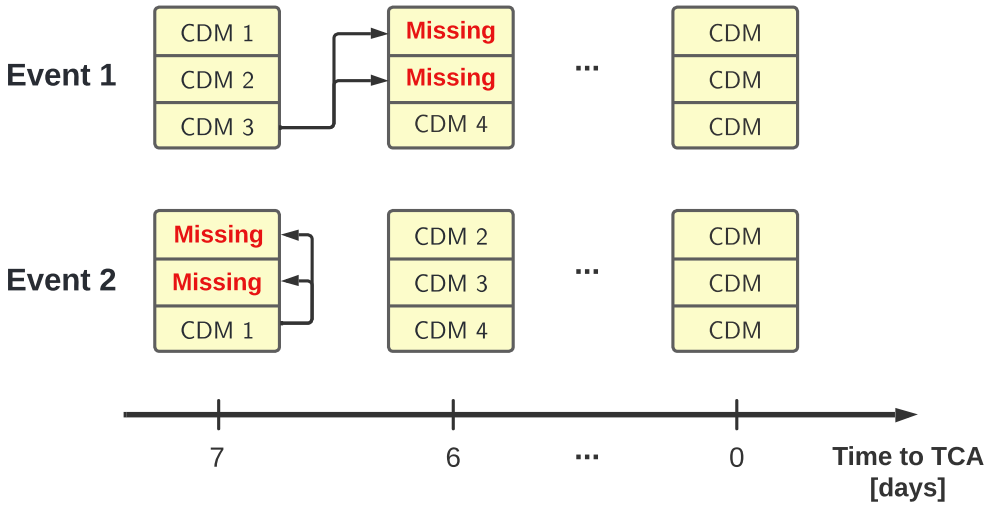


Figure 5.1: Data Setup schematization.

After the data cleaning and setup, the training data is split into five folds to perform cross-validation and find the best HMMs, for **Approach A** and **Approach B**.

Chapter 6

Bayesian Sequential Models

This chapter focuses on the definition and implementation of the probabilistic models used in this work. As described in Chapter 4, two approaches are going to be tested: using an HMM to directly predict the risk contained in the CDMs after the cut-off time (**Approach A**) and using separate HMMs to predict the chaser covariances and then use Akella's and Alfriend's algorithm to compute the final risk (**Approach B**), in agreement with what was observed in the data. In this work, it is assumed that both the sequence of the risk and sequences of the chaser covariances contained in the CDMs released during the conjunction events can be modeled by HMMs in which the latent variables generate observations through Normal distributions, i.e, it is expected that the risk/position errors generated by each state should be near a specific value and the occurrence of risk/position errors far from that value is less frequent. Physically, it is expected that the risk and position uncertainties updates within the CDMs have distributions that depend on the state of the hidden Markov chain at each time instant. However, it is important to note that the variables to model must follow some constraints. On the one hand, the values of the risk contained in the CDMs provided by the 18 SPCS are truncated at a lower bound of –30, as previously mentioned, and cannot be greater than 0, because the risk is defined as the $\log_{10}$ of the collision probability. On the other hand, the standard deviations that represent the position errors of the chaser are restricted to be greater than zero. To take these constraints into account, truncated normal distributions are used as the emission distributions of the HMMs of both **Approach A** and **Approach B**. A truncated normal distribution is a transformation of a normal distribution in which the random variable is bounded by a lower, an upper value, or both, and its probability density function is zero outside those bounds.

Since each HMM will learn and predict only one feature of the dataset, as explained in Section 4.2.2, the emissions can be specified by univariate distributions (in this case, univariate truncated normal distributions). Therefore, the parameters that must be learned for each implemented HMM are the following, where K represents the number of possible hidden states:

- the transition probabilities represented by the stochastic matrix $\mathbf{A} \in \mathbb{R}^{K \times K} : \mathbf{A} \geq 0,\ \mathbf{A}\mathbb{1} = \mathbb{1}$;
- the initial probability distribution represented by the vector $\boldsymbol{\pi} \in \mathbb{R}^{K} : \boldsymbol{\pi} \geq 0,\ \boldsymbol{\pi}^{T}\mathbb{1} = 1$;

- the mean values $\boldsymbol{\mu} \in \mathbb{R}^K$ of the emission distributions, where $\boldsymbol{\mu}$ denotes the set of mean values of the emissions as $\boldsymbol{\mu} = [\,\mu_1, \mu_2, ..., \mu_K\,]$, in which μ_k represents the mean of the truncated normal emission generated by the hidden state k.

- the standard deviations $\boldsymbol{\sigma} \in \mathbb{R}_+^K$ of the emission distributions, where $\boldsymbol{\sigma}$ denotes the set of standard deviations of the emissions as $\boldsymbol{\sigma} = [\,\sigma_1, \sigma_2, ..., \sigma_K\,]$, in which σ_k represents the standard deviation of the truncated normal emission generated by the hidden state k;

- the lower and/or upper bounds of the emission distributions.

Most of these quantities depend on the number of hidden states in the HMMs. So, to find the best number of possible states, in each approach, various HMMs with different values for this parameter are inferred and the one that shows the best performance is chosen as the final model. To avoid overfitting problems, this is performed using a stratified cross-validation with five folds in the training set.

As discussed in Section 2.6, in order to define the probabilistic models using a Bayesian approach, the following distributions are needed: the likelihood and the priors. The likelihood for HMMs describes the joint probability of the observed data as a function of the parameters and is already defined in equation (2.21). The prior distributions are a key part of Bayesian inference, reflecting the prior beliefs/knowledge one holds about the parameters. In the work discussed in this thesis, to perform Bayesian inference on HMM, it is essential to define the prior distributions for each of the parameters $\theta = (\mathbf{A}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma})$. The lower/upper bounds of the truncated normal distributions are hyperparameters. Section 6.1 presents the definition of the priors for the HMM of **Approach A** and Section 6.2 describes the priors selected for the three separate HMMs of **Approach B**. One approximation followed in this work, in order to implement the predictive distribution is also explained.

6.1 HMM for Risk Evolution

This Section describes the selected priors for the HMM used to analyze the risk sequences of the conjunction events and it is divided into two parts: priors for the transition and initial distributions, and priors for the parameters of the emission distributions. In the end of this section, an approximation followed in this work, regarding the implementation of the predictive distribution of the HMMs, is also described.

Priors for the transition and initial distributions

Regarding the parameter $\boldsymbol{\pi}$, which describes a probability distribution for the initial state, a natural choice of prior is the Dirichlet distribution. The Dirichlet distribution over the variable $\boldsymbol{\pi}$ is confined to a simplex, i.e, all elements of $\boldsymbol{\pi}$ in this distribution belong to the interval $[0, 1]$ and $\sum_{i=1}^{K} \pi_i = 1$ [37]. Therefore, its output is a vector in which all the elements are positive and sum up to one, describing in this case the probability of the first latent variable in the chain being one of the K possible states. The distribution is parameterized by the vector $\boldsymbol{\alpha} \in \mathbb{R}^K : \boldsymbol{\alpha} > 0$, whose elements must be positive real

numbers. This parameter defines the density over the simplex and, if all the elements of $\boldsymbol{\alpha}$ are equal, the Dirichlet is symmetric. In those cases, if the values of the parameters are greater than one, the density is higher in the center of the simplex favoring values for the elements of $\boldsymbol{\pi}$ that are similar to each other and, if the values of the parameters are less than one, the density has peaks on the vertices of the simplex, favoring sparse distributions [67]. In the special case in which all the values of $\boldsymbol{\alpha}$ are equal to one, the Dirichlet distribution is equivalent to a uniform distribution over the simplex. In this work, there is no prior knowledge about the first state that generates the risk in each event, so there is no reason to favor one component of $\boldsymbol{\pi}$ over another. Therefore, a symmetric Dirichlet distribution with the elements of $\boldsymbol{\alpha}$ equal to one will be used as a prior for $\boldsymbol{\pi}$.

The same reasoning can be applied to the rows of the transition matrix $\mathbf{A}$. As it was discussed in Section 2.5.1, each row of the matrix is a probability distribution that describes the probability of transitioning from one hidden state to another, and, in this work, there is no prior knowledge about these transitions, so a symmetric Dirichlet distribution with the elements of $\boldsymbol{\alpha}$ equal to one will also be used as a prior for each row of $\mathbf{A}$, representing the transition probabilities for a given state.

In conclusion, the priors for the transition and initial distributions are given as follows:

$$\begin{aligned} \boldsymbol{\pi} &\sim \text{Dir}(\boldsymbol{\alpha} = \mathbb{1}), \\ \mathbf{A}_i &\sim \text{Dir}(\boldsymbol{\alpha} = \mathbb{1}), \qquad \forall\, i \in \{1, ..., K\}, \end{aligned}$$

in which $\mathbb{1}$ is used to denote a K-dimensional vector with all the elements equal to one and $\mathbf{A}_i$ represents the i-th row of $\mathbf{A}$.

Priors for the parameters of the emissions

Regarding the truncated normal emission distributions, the priors for $\boldsymbol{\mu} \in \mathbb{R}^K$ and $\boldsymbol{\sigma} \in \mathbb{R}_+^K$ must be specified, as well as the lower and upper bounds. It is known that the observations (the risk of collision) can take values between -30 and 0, so these values are defined as the lower and upper bounds of the emission distributions, respectively. In addition, it is assumed that the emissions must also have mean values within this range. A very common prior for the mean values of normal distributions, or its variants, is itself a normal distribution. However, the mean values of the distributions are restricted to be within the -30 to 0 range, so, truncated normal distributions are also used as the prior distributions of $\boldsymbol{\mu}$, with lower and upper bounds of -30 and 0, respectively. To have good coverage of all the possible values that the observations can take, the mean of the prior distributions for the elements of $\boldsymbol{\mu}$ are equally spaced within the range of -30 to 0 and the standard deviations are set to 4. For example, if $K = 4$, the priors for the elements of $\boldsymbol{\mu}$ will be: $\mu_1 \sim \mathcal{TN}(\mu = -30, \sigma = 4)$, $\mu_2 \sim \mathcal{TN}(\mu = -20, \sigma = 4)$, $\mu_3 \sim \mathcal{TN}(\mu = -10, \sigma = 4)$ and $\mu_4 \sim \mathcal{TN}(\mu = 0, \sigma = 4)$, in which the values of the lower and upper bounds of the truncated normal distribution are not shown, because these are fixed throughout at the values -30 and 0, respectively. To write the priors for $\boldsymbol{\mu}$ in a clear and compact way for a HMM with K possible hidden states, the following

notation will be used:

$$\boldsymbol{\mu} \sim \mathcal{TN}\big(\,\mu = m,\, \sigma = 4,\, \mathrm{LB} = -30,\, \mathrm{UB} = 0\,\big),$$

in which $m \in \mathcal{M}_K(-30, 0)$, where $\mathcal{M}_K(a, b)$ is the set of K evenly spaced numbers between a and b. In addition, LB and UB denote the lower and upper bounds of the truncated distributions, respectively.

As for the priors of $\boldsymbol{\sigma}$, it is necessary to choose a distribution that can only take positive values, because standard deviations are constrained to be greater than zero. In the context of Bayesian modeling, it is extremely important for the priors of the elements of $\boldsymbol{\sigma}$ of the truncated normal distributions to have zero density at the value of 0. If the standard deviations are zero, the truncated normal distributions become degenerate [68], meaning that the distributions have infinite density at their mean values, posing a problem to current MCMC samplers and consequently leading to wrong inferences. Thus, the chosen distribution for the priors of the elements of $\boldsymbol{\sigma}$ is the inverse gamma distribution which is the conjugate prior (when the posterior is in the same probability distribution family as the prior distribution) for the variance of a univariate Gaussian [37]. This distribution is only defined on the set of positive real numbers and has two positive parameters: α, which defines the shape of the probability density function, and β, which defines its spread. After a trial and error process and considering the previously described constraints, the chosen values for the parameters of the inverse gamma distributions are $\alpha = 40$ and $\beta = 80$. In summary, the priors chosen for the elements of $\boldsymbol{\sigma}$ are given as follows:

$$\sigma_i \sim \mathcal{IG}(\alpha = 40, \beta = 80), \qquad \forall\, i \in \{1, ..., K\}.$$

Notice that the parameters of the priors for the emission distributions were chosen via a trial-and-error process (taking into consideration the described constraints for $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$), so they are not unique and can be further improved. In addition, in some of the implemented models, the parameters of the chosen priors for $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ had to be slightly adjusted.

Approximation of the predictive distribution

With the priors and likelihood, the posterior can be inferred and predictions for the HMM for the risk evolution can be obtained, using the predictive distribution shown in (2.23). In order to propagate the posterior uncertainty into the predictions, random draws for the parameters of the model are taken from the inferred posterior distribution and are given as input to the predictive distribution. For each drawn set of parameters, a different predictive distribution is obtained and a random draw is taken from each of them. However, since truncated normal distributions are used as the emission distributions of the HMMs, the computation of the predictive distribution would require the sum of the probability distribution of multiple truncated normal distributions, resulting in a multimodal probability density function that is very difficult to compute analytically. Due to time constraints, this distribution is approximated by a truncated normal distribution with mean value equal to the first momentum of the distribution of (2.23) and variance equal to the second momentum of the distribution. Future work should focus on computing

the true predictive distribution, without any approximation, but, in this work, this is simplified.

6.2 HMMs for Covariance Evolution

This Section explains the selected priors for the HMMs used to analyze the evolution of the position uncertainties of the chaser object and, just like in the previous Section, it is divided into two parts: priors for the transition and initial distributions, and priors for the parameters of the emission distributions.

Priors for the transition and initial distributions

Like in the HMM for the risk evolution, a natural choice of priors for $\boldsymbol{\pi}$ and $\mathbf{A}$ is the Dirichlet distribution. Again, in this work, there is neither any prior knowledge regarding the first state that generates the first observation of any of the three position uncertainties of the chaser object nor any belief about the probability of transitioning from one hidden state to another. Therefore, a symmetric Dirichlet distribution with the elements of $\boldsymbol{\alpha}$ equal to one is going to be used for the priors of $\boldsymbol{\pi}$ and of the rows of $\mathbf{A}$, for all the three separate HMMs.

Priors for the parameters of the emissions

When defining the priors for the parameters of the emission distributions of the HMMs of **Approach B** it is important to recall that the three variables to model (σ_R^{chaser}, σ_T^{chaser} and σ_N^{chaser}) represent standard deviations, so they are restricted to be greater than zero. Note that, in this work, the performance of the HMMs for the evolution of the standard deviations is benchmarked with truncated normal emissions, but future work should test the performance of gamma or inverse gamma emissions. A lower bound of 0.01 m is chosen for the truncated normal emission distributions, where it is reasonable to assume that the position uncertainties caused by measurement and/or propagation errors have a minimum value of 1 cm. In addition, since it is assumed that the observations of the chaser position uncertainties are always greater than 0.01 m, then the mean values of the emission distributions can only take values above this threshold. Hence, truncated normal distributions are used as the prior distributions of $\boldsymbol{\mu}$, with, again, a lower bound of 0.01 m. Before defining the other parameters of the prior distributions of $\boldsymbol{\mu}$ (mean value and standard deviation), it is important to note that, as per the data preparation phase, it is considered that the CDMs that contain values above 100 000 m, 1 000 m and 450 m for the along-track, radial and normal position errors of the chaser, respectively, are considered outliers. So, it can be assumed that the values of $\boldsymbol{\mu}$ of the emission distributions of the respective HMMs are below these upper thresholds. Similarly to what was presented in Section 6.1, the mean of the truncated normal priors for the elements of $\boldsymbol{\mu}$ of each of the three HMMs are equally spaced within the range from 0.01 m to the respective upper threshold of the variable to model. In summary, the priors chosen for the mean values of the emission distributions of the three HMMs are the following:

- HMM for σ_R^{chaser} : $\boldsymbol{\mu} \sim \mathcal{TN}\big(\mu = m_R,\, \sigma = 400,\, \text{LB} = 0.01\big)$;
- HMM for σ_T^{chaser} : $\boldsymbol{\mu} \sim \mathcal{TN}\big(\mu = m_T,\, \sigma = 10\,000,\, \text{LB} = 0.01\big)$;

- HMM for σ_N^{chaser} : $\boldsymbol{\mu} \sim \mathcal{TN}\big(\mu = m_N,\, \sigma = 100,\, \text{LB} = 0.01\big)$,

where $m_R \in \mathcal{M}_K(0.01, 1\,000)$, $m_T \in \mathcal{M}_K(0.01, 100\,000)$ and $m_N \in \mathcal{M}_K(0.01, 450)$. The standard deviations of the truncated normal distributions were chosen so that the density areas of the priors covered a large amount of possible values for the parameters of $\boldsymbol{\mu}$.

As for the prior distributions for $\boldsymbol{\sigma}$, the chosen distribution was, again, the inverse gamma, which is the conjugate prior for the variance of a univariate Gaussian and is defined by the parameters α and β. In order to allow the data to have more influence on the posterior for the values of $\boldsymbol{\sigma}$, the parameters α and β were chosen such that the density areas of the inverse gamma distribution could cover a large amount of possible values for the standard deviations of the emission distributions. After a trial and error process, the final chosen values of α and β for the inverse gamma priors for the standard deviations of the emission distributions of the three HMMs are the following:

- HMM for σ_R^{chaser} : $\sigma_i \sim \mathcal{IG}(\alpha = 7, \beta = 545), \quad \forall\, i \in \{1, ..., K\}$;
- HMM for σ_T^{chaser} : $\sigma_i \sim \mathcal{IG}(\alpha = 4.5, \beta = 28\,480), \quad \forall\, i \in \{1, ..., K\}$;
- HMM for σ_N^{chaser} : $\sigma_i \sim \mathcal{IG}(\alpha = 6, \beta = 500), \quad \forall\, i \in \{1, ..., K\}$.

With the priors and the likelihood, the posterior can be inferred and predictions can be made. Just like in the HMM for the risk evolution, the predictive distribution is approximated by a truncated normal distribution (read Section 6.1 for more details).

6.3 Inferences

With the likelihood and prior distributions, it is possible to use MCMC algorithms to infer the parameters of the implemented models. As previously described, a stratified cross-validation with five folds is performed in order to find the best value for K (the number of possible hidden states of the HMMs) and the best models for **Approach A** and **Approach B** are then trained using the entire training set and evaluated using the test set. To infer the parameters of the models, or, in other words, sample from the posterior distribution, the NUTS [53] is used. During cross-validation, 3 chains of 2 000 iterations are sampled (which is considered sufficient to analyze the convergence of the sampled chains [69]) for each model and, for the inference of the final HMMs of both **Approach A** and **Approach B** on the entire training set, 5 chains of 2 000 iterations are sampled. The number of warm-up/tuning iterations per chain is left at the default value of 1 000 and, after sampling, the samples used for tuning in each chain are discarded. In each sampling procedure, the target acceptance rate is set to a value of 0.8, which, in this work, allows a good compromise between the sampling time and the removal of divergent transitions.

After sampling, it is necessary to deal with switching component labels within the inferences of each HMM (read Section 2.6.2 for a detailed explanation of the label switching problem). In this work, the sampled chains only contained switching component labels between different chains, so the label switching problem was solved by post-processing the Bayesian estimation output and relabelling the chains according to the analysis of the median of each chain of the parameters $\boldsymbol{\mu}$. More specifically, the median of

each sampled chain of the parameters μ is computed and the chains that present an absolute difference between their medians below a defined threshold (the threshold is hand-tuned for each HMM) identify the labels of the HMM that need to be switched.

After dealing with the label switching problem, the convergence and autocorrelation of the sampled chains are checked using the diagnostic tools described in Section 2.6.2. If the inferences pass all the requirements, the samples of the posterior distribution can be used to obtain predictions.

Chapter 7

Results

This chapter presents the results obtained with the proposed models. In Section 7.1, the results obtained using the CAC and the Neuraspace datasets for **Approach A** are described and Section 7.2 presents the results for **Approach B**, using the Neuraspace dataset (the data from the CAC cannot be used for this approach, due to the fact that some of the necessary features are not available — read Chapter 4 for more details).

7.1 Approach A

This Section presents the results obtained with the proposed models for **Approach A** — directly model and predict the risk of collision within the events — and it is divided into two parts: Section 7.1.1 presents the results using the CAC dataset and Section 7.1.2 presents the results using the Neuraspace dataset. The methodology followed in this approach is schematized in Figure 4.3a of Chapter 4, where it is possible to see that, before making predictions, it is necessary to train the HMM for the analysis of the evolution of the collision risk within the events. As previously mentioned, in order to find the best number of possible states K for the HMMs, various models with different values for K are inferred, using stratified cross-validation with 5 folds in the training set, and the one that shows the best performance is chosen as the final model. Hence, each of the Sections follows the same structure: first, the cross-validation results are presented and, then, the results obtained with the final and complete model are described.

7.1.1 CAC Dataset

Stratified Cross-Validation Results

To find the best HMM for the risk prediction using the CAC dataset, the value of K is iterated between 4 and 10 states using cross-validation and, in each stratified fold, 3 chains of 2 000 iterations are sampled for each model. In HMMs, the predictions tend to be arranged near the mean values of the emission distributions (as will be seen in the next Section), so it is considered that HMMs with a number of possible

hidden states lower than 4 have poor coverage of all the possible values of the desired parameter (in this case, the risk) — having to have a large value for the standard deviations of the truncated normal emissions — and, hence, the performance of the model with a lower number of K is not tested in this work. For a larger number of K, the sampled chains for the parameters of the HMMs start converging towards different values, which indicates that the posterior distribution is multimodal. In the cases of multimodal posterior distributions, some of the parameters of the HMC/NUTS algorithms (like the mass-matrix and the leapfrog stepsize — see Appendix A) may only be locally optimized for one sharp density curvature of the posterior (one of the modes), during the warm-up/tuning phase, and the NUTS sampler can get stuck in that sharp region of the posterior density while sampling, failing to explore the rest of the density areas. Thus, by sampling randomly initialized chains, the sampler may get stuck in different modes, each time, which justifies the fact that the chains converge into different values. Future work might tackle this issue by using/developing an efficient sampler that follows Hamiltonian dynamics and can handle multimodality, but, in this work, this step is simplified. Hence, when the chains converge into different values for a given HMM with k possible hidden states, the model is reparameterized (i.e., the priors are slightly adjusted) and, if after reparameterization the chains do not converge, it is assumed that k is the maximum number of possible states for the HMM. Note that only 3 chains are sampled during cross-validation, due to the large computing time during Bayesian inference, so it is possible that, even if the chains converge, the sampler may only be exploring part of the posterior distribution. Although this is not ideal, it still offers good information regarding the posterior distribution, since it explores the density regions near a mode of the desired distribution, in contrast to the maximum likelihood estimation or maximum a posterior that only provide point estimates.

After sampling, the posterior draws are used to predict the final risk of each event. As previously mentioned, this work benchmarks the performance of HMMs by predicting only the values of the parameters contained in the next observation $\mathbf{x}_{N+1}$ of each event, in which the sequence of CDMs/observations received until the cut-off threshold is denoted by $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$, where N is the number of CDMs received until the cut-off time. Each newly received CDM is considered the best knowledge about the outcome of the close approach, so the observation $\mathbf{x}_{N+1}$ is considered more precise than the information contained $\mathbf{x}_N$. However, during a conjunction event, 3 CDMs are received, on average, per day [17], and since the defined cut-off time is 2 days before the TCA, it can be advantageous to predict the risk contained in the next k collision warnings after the last released CDM. Future work may test the performance of predicting the next k observations of an event but, in this work, this step is simplified. Notice that, although only the next observation of each event is predicted, the results are compared to the last released CDM of the given event (which is considered the label), even if more CDMs are received within 2 days of the TCA.

In order to propagate the uncertainty of the sampled posterior into the predictions of each observation, 400 draws are randomly taken from each sampled chain (which gives a total of 1 200 posterior draws) and are used as the input parameters of the predictive distribution of the HMM. Thus, the predictions of each observation provide a distribution that nicely reflects the posterior uncertainty and supports the trustworthiness and confidence of the predicted risk. The final predicted value for the risk of each

observation is the mean of the respective distribution.

After cross-validation, the final scores of each model are computed by averaging the scores obtained in each fold. The scores of cross-validation for HMMs with a different number of possible states are summarized in Table 7.1. Recall that in addition to the regression problem of predicting the risk values, this work aims to identify the risk classes of each event (i.e., if the risk is higher than –6, the event is considered high-risk, otherwise it is considered low-risk), so regression and classification metrics are used. The results show that, overall, the regression metrics (RMSE and MAE) tend to have lower values the higher the number of hidden states of the HMM. In HMMs, the predictions of the final risk of each event tend to be arranged near the mean value of the emission distributions and, since the emissions of the HMMs with a lower number of possible states have a poorer coverage of all the possible values of the risk, their predictions may be distant from the true values, which consequently increases the prediction errors.

States	RMSE	MAE	Precision (%)	Recall (%)	$\mathbf{F}_1$	$\mathbf{F}_2$
4	13.777	11.611	0.0	0.0	0.0	0.0
5	13.681	11.471	0.0	0.0	0.0	0.0
6	13.653	11.324	17.49	55.00	0.216	0.325
7	13.621	11.274	17.09	82.50	0.283	0.467
8	13.620	11.193	17.18	82.50	0.284	0.468
9	13.601	11.133	16.29	77.50	0.269	0.442
10	13.621	11.117	16.55	65.00	0.262	0.406

Table 7.1: Summary of the cross-validation results obtained with HMMs with different number of possible states, for **Approach A**, using the CAC dataset. The shaded cells highlight the best score for each metric.

Table 7.1 shows that the HMMs with K equal to 4 and 5 fail to identify all the high-risk conjunctions and, thus, these are not chosen as the final model. Although the HMMs with 6, 9, and 10 possible states have slightly better precision, RMSE and MAE metrics, respectively, they have significantly lower recall, F_1 and F_2 scores when compared to the other HMMs, which indicates that these models have poorer performance in the classification task, more specifically, they have worse performance in identifying high-risk conjunctions. Therefore, none of the HMMs with K equal to 6, 9, or 10 is chosen as the final HMM for **Approach A**. Between a number of K equal to 7 or 8, it is preferable to choose the HMM with K equal to 8, because it has slightly lower RMSE and MAE and higher precision, F_1 and F_2 scores. Therefore, the HMM with $K = 8$ is chosen as the final HMM for **Approach A**, using the CAC dataset.

Next, the performance on the test set of the complete model for **Approach A** (which is represented in Figure 4.3a) is compared to the baseline solution.

Final Model Results

Recall, from Chapter 4, the prediction process of the complete model of **Approach A**: the events in the test set that have a baseline risk solution equal to –30 are directly predicted as having a final risk

equal to –30, and the other events are analyzed by the final HMM (chosen via cross-validation) and their risk is predicted using the predictive distribution, presented in (2.23) of Chapter 2.

To infer the parameters of the final HMM with 8 possible states, 5 chains of 2 000 iterations are sampled using the entire training set. Then, to obtain predictions, 400 draws are randomly taken from each sampled chain, giving a total of 2 000 posterior draws that are used as the input parameters of (2.23), providing a distribution that reflects the posterior uncertainty for each predicted observation. The final predicted value is, again, the mean of the distribution. Table 7.2 summarizes the obtained performance metrics for both the complete model of **Approach A** and the baseline predictions, using the CAC dataset.

Model	RMSE	MAE	Precision (%)	Recall (%)	$\mathbf{F}_1$	$\mathbf{F}_2$
Approach A	8.403	4.318	17.50	70.0	0.280	0.438
Baseline	8.879	4.430	14.58	70.0	0.241	0.398

Table 7.2: Summary of the performance metrics of the complete model of **Approach A** and baseline solution, using the CAC dataset. The shaded cells highlight the best score for each metric.

The complete model of **Approach A** outperforms the baseline solution in all of the metrics, with the exception of the recall, which is equal to the baseline approach. The lower RMSE and MAE of the implemented model show that its predictions are closer to the true target values of the risk and its higher F_1 and F_2 scores reveal that the implemented model is slightly better at evaluating the class of the events (high or low-risk). These results show that, despite the approximations made to build this model (e.g. predicting only the next observation and manipulating the events during the training data setup), the complex behavior of the risk updates within the events, the data imbalance problem, and the fact that only one of the 103 features of the entire dataset is used, the implemented model manages to outperform the naive forecast, which is considered a very strong predictor for this problem [17]. Additionally, when compared to the ML methods benchmarked by Abay et al. [26], the model implemented in this work is the only one that outperforms the baseline solution in all metrics, even though most methods in [26] use many features and the model proposed in this work only uses the sequence of the values of the risk of collision within the events.

To better understand the classification performance of both methods, the confusion matrices for the implemented model and baseline approach are shown in Tables 7.3 and 7.4, respectively.

		Predicted	
		Low-risk	**High-risk**
True	**Low-risk**	1406 (TN)	33 (FP)
	High-risk	3 (FN)	7 (TP)

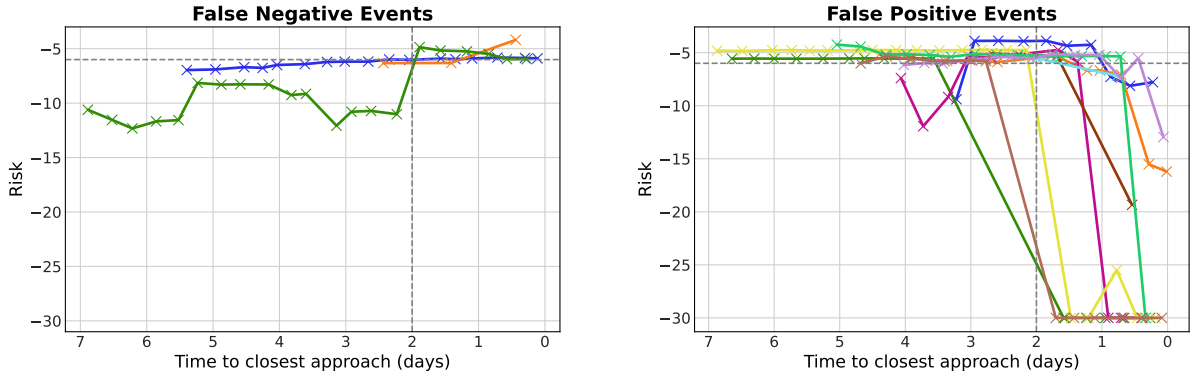
Table 7.3: Confusion matrix for the model of **Approach A**, using the CAC dataset.

		Predicted	
		Low-risk	**High-risk**
True	**Low-risk**	1398 (TN)	41 (FP)
	High-risk	3 (FN)	7 (TP)

Table 7.4: Confusion matrix for the baseline solution, using the CAC dataset.

It can be seen that both models have the same number of FN and TP, but the implemented model reduces the number of false positives by approximately 19.5%, which justifies its higher precision. The

true high-risk events that were wrongly classified as low-risk (false negatives) are the same for both models and the time-series evolution of the risk update within those events is shown in Figure 7.1a. The events that were wrongly classified as high-risk by the implemented model were also miss-classified by the baseline and the evolution of some of these events (that are a good representation of the evolution of the events miss-classified as the positive class) is shown in Figure 7.1b.



(a) Risk evolution of false negatives.

(b) Risk evolution of false positives.

Figure 7.1: Risk evolution of the events that were wrongly classified by both models as low-risk (left) and as high-risk (right), following **Approach A**, with the CAC dataset. The colored lines represent different events and the crosses mark the risk updates.

Both plots of Figure 7.1 show that the events that generated FN and FP predictions actually evolve in the vicinity of the high-risk threshold. All the false positive predictions have origin in events that have a risk value higher than the high-risk threshold before the cut-off time and, then, experience a significant jump toward lower values. This type of evolution highlights the complex and unpredictable behavior of the risk updates within the events and suggests that, to correctly predict these risk transitions, more features of the dataset should be analyzed by the ML models.

Figure 7.2 shows the predicted values of the risk (in the y axis) over the true values (in the x axis) for both the model of **Approach A** and the baseline predictions, using the CAC dataset. .

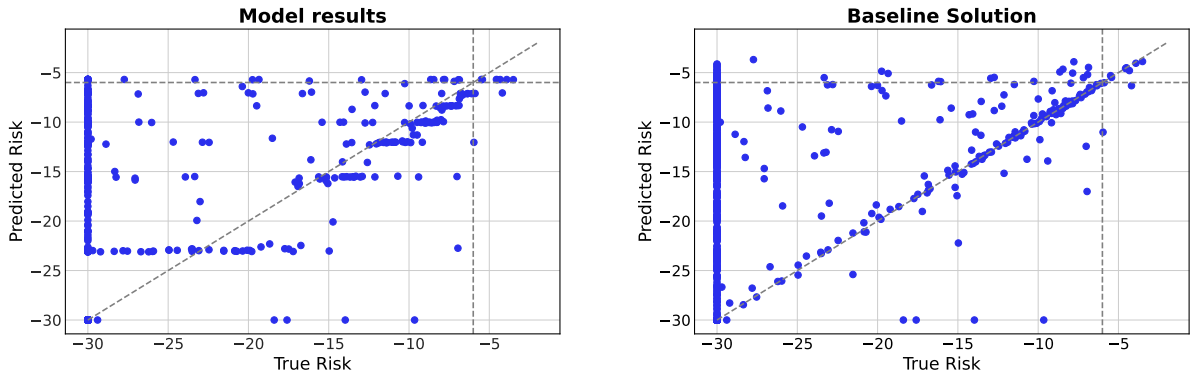


Figure 7.2: Prediction results for the model of **Approach A** (left) and for the baseline (right), using the CAC dataset. The diagonal dashed line represents the values where predictions of the risk are equal to the true values and the horizontal and vertical lines mark the high-risk threshold for the predictions and for the true risk, respectively.

It can be seen from the left plot of Figure 7.2 that the predictions made by the implemented model for the risk prediction tend to be arranged in 8 steps, which correspond to the sampled mean of the emission

distributions of the HMM (see Table B.1 of Appendix B), which are generated by the 8 possible latent variables. Since the final predictions are arranged near the mean values of the emission distributions, the predictions seem to be “truncated” at an upper value of approximately –5 (the maximum sampled value for the mean of the emission distributions) and, consequently, the predictions of the high-risk events are not exactly aligned with the true regression line. This is due to the data imbalance problem, suggesting that it could be beneficial to train the model with a larger dataset containing more high-risk conjunctions. Although the predictions obtained with the model are not precisely aligned with the dashed diagonal line that represents the true risk, most of the predicted data points are “assigned” to a prediction step that is close to their true label. Figure 7.2 also shows that, in both models, there is a big over-prediction of the risk of the events with a true label of –30. Figure 7.3 shows the actual time series evolution of some of the events that were over-predicted by both models and that represent the typical evolution of the risk of the over-predicted conjunction events.

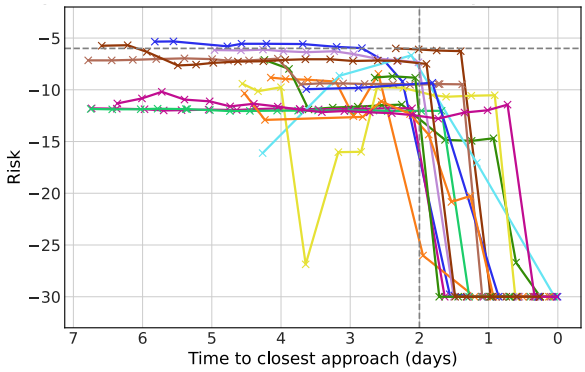


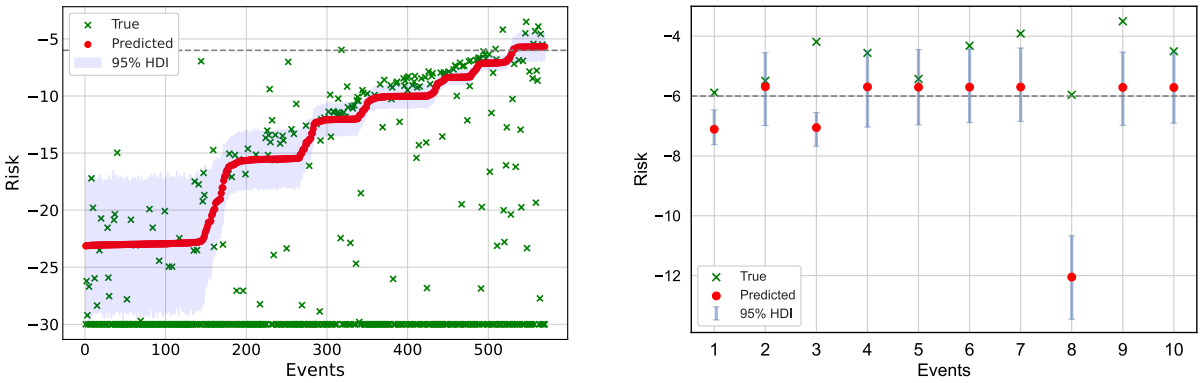
Figure 7.3: Typical time series evolution of the collision risk of the events that were over-predicted by the complete model of **Approach A** and the baseline, using the CAC dataset. It can be seen that the events experience a big risk transition from higher values to –30, after the cut-off time.

Figure 7.3 shows that most over-predicted –30 events evolve at high-risk values, but, after the cut-off time, they experience a big risk transition, that cannot be predicted, from higher risk values to –30. This type of risk evolution within the events also explains the high number of false positive predictions made by both models.

A closer examination of the predictions made by the implemented model and the baseline shows that the latter makes 218 big over-predictions of low true risk events, whereas the HMM only makes 172 over-predictions, where it is assumed that a large over-prediction of a –30 true risk event occurs when the corresponding predicted value is higher than –15. The implemented model reduces the number of over-predictions by approximately 21.1%, which shows that it has better performance than the baseline in identifying the risk transitions from higher to lower values.

As previously mentioned, 2 000 posterior draws are used as the input parameters of the predictive distribution of the HMM, which leads to 2 000 risk predictions per event, creating a distribution that can reflect the uncertainty associated with each prediction. Figure 7.4 shows the predictions made by the HMM and the final risk values (risk contained in last CDM of each event), as well as the 95% highest density interval (HDI) associated to each prediction, for all of the events that went through the prediction

process of the HMM (Figure 7.4a) and for the true high-risk events (Figure 7.4b). The HDI "*indicates which points of a distribution are most credible and [...] cover most of the distribution*" [70]. A 95% HDI is the smallest credible interval (interval in the domain of a specific distribution within which a parameter falls with a certain probability) in which the area under the probability density function curvature is 95% (for a univariate distribution). Note that, in the plot of Figure 7.4a, the predictions are arranged in ascending order, in order to be easier to visualize the prediction intervals.



(a) All events in the CAC dataset.

(b) True high-risk events in the CAC dataset.

Figure 7.4: Representation of the true risk values of each event (green), the HMM predictions (red) and the 95% HDI area (area shaded in blue), for all the events of the CAC dataset (left) and for the high-risk events (right), following **Approach A**.

Figure 7.4a shows, again, that the final predicted values (which are the mean of the predicted distributions) tend to be arranged near the sampled mean values of the emission distributions of the HMM (as it can be confirmed by B.1 of Appendix B) and that the HDI interval of the predictions within each prediction step is almost constant. In fact, the 95% HDI in each prediction step covers the risk values that lie approximately two standard deviations from the mean of the corresponding emission distributions, which are transformations of Gaussian distributions, indicating that most of the predictions that result from (2.23) are generated by Truncated Normal distributions that correspond to the emissions of the HMM. Figure 7.4a also shows that the prediction intervals do not have very good coverage of all the true risk values, failing to cover all of the events that have a final risk of –30 and only covering 44% of the true risk values of the events with a final risk higher than the minimum risk value. In addition, it can be seen from Figure 7.4b that most of the high-risk true values are not captured by the 95% HDI, with only 30% of the high-risk conjunctions being captured by the prediction intervals. However, it is important to note that even though this approach has shown poor performance in providing prediction intervals that cover the true risk values, the implemented model has outperformed the baseline solution in all of the metrics, despite the data imbalance problems, the complex behavior of the risk evolution within the events and the fact that only the *risk* feature of the entire dataset was used.

7.1.2 Neuraspace Dataset

This Section presents the results obtained with the models for **Approach A**, using the Neuraspace dataset, and, again, it is divided into two parts: the cross-validation results and the complete model

results.

Stratified Cross-Validation Results

To find the best HMM, the number of possible hidden states is iterated between 4 and 9 states in each stratified cross-validation fold. A higher number for K, in this case, causes the sampled chains to converge into different values, meaning that the posterior is multimodal with sharp density curvatures, as previously explained. Notice that K is iterated between different ranges for the HMMs of **Approach A** using the CAC and Neuraspace datasets. The CAC dataset is an anonymized dataset that was prepared and slightly manipulated by ESA in order to be publicly released and contain a higher percentage of high-risk conjunctions to test the use of ML models in predicting the risk of collision, whereas the Neuraspace dataset contains raw and unmanipulated data regarding real-life close approaches. Hence, using the CAC and Neuraspace data are two distinct problems, which justifies the different ranges tested for K, for the HMMs.

Just like in Section 7.1.1, the next observation of each event is predicted and compared to the true label. To propagate the uncertainty of the sampled posterior into the predictions of each observation, 400 draws are randomly taken from each sampled chain of the parameters (which gives a total of 1 200 posterior draws) and are used as the input of the predictive distribution of the HMM. Hence, for each observation, an entire distribution for the predictions is obtained, in which the final predicted value for the risk of each observation is the mean of the respective distribution.

Like in the previous Section, the final scores of each model for cross-validation are computed by averaging the scores obtained in each fold. The cross-validation results are presented in Table 7.5.

States	RMSE	MAE	Precision (%)	Recall (%)	$\mathbf{F}_1$	$\mathbf{F}_2$
4	12.469	10.338	0.0	0.0	0.0	0.0
5	12.408	10.087	17.27	35.0	0.227	0.285
6	12.363	9.875	26.21	75.0	0.384	0.537
7	12.308	9.758	31.93	95.0	0.472	0.671
8	12.272	9.714	33.92	95.0	0.493	0.686
9	12.262	9.674	33.92	95.0	0.493	0.686

Table 7.5: Summary of the cross-validation results obtained with HMMs with different number of possible states, for **Approach A**, using the Neuraspace dataset. The shaded cells highlight the best score for each metric.

Again, the values of the regression metrics (RMSE and MAE) tend to decrease as the number of states increases. The reasons for these lower prediction errors were already discussed in Section 7.1.1. Analyzing the results, it can be verified that the HMMs with K equal to 4, 5, 6, and 7 have worse RMSE, MAE, F_1 and F_2 metrics and, hence, none of them is chosen as the final HMM. The scores of the HMMs with 8 and 9 possible states are the same for the classification metrics, but the model with K equal to 9 has better RMSE and MAE, so it is chosen as the final HMM for **Approach A**, using the Neuraspace dataset.

Notice that the scores of the model tend to get better as the number of states increases, suggesting that future work should tackle the problem of sampling from multimodal posterior distributions with sharp density curvatures, in order to test the performance of HMMs with a higher number of possible hidden states.

Final Model Results

The parameters of the HMM with K equal to 9 are then inferred and 5 chains are sampled from the posterior distribution, using the entire training set. Then, for each parameter of the HMM, 400 posterior draws are randomly taken from each sampled chain (giving a total of 2 000 draws) and are used as the input parameters of the predictive distribution of the HMM, leading to 2 000 risk predictions per event and creating a distribution that reflects the accuracy of the predictions, in which the final predicted risk value is the mean of the distribution.

The performance metrics obtained with the complete model of **Approach A** (which is schematized in Figure 4.3a) and the baseline solution on the test set, using the Neuraspace dataset, is summarized in Table 7.6.

Model	RMSE	MAE	Precision (%)	Recall (%)	$\mathbf{F}_1$	$\mathbf{F}_2$
Approach A	2.541	0.399	31.25	62.50	0.417	0.521
Baseline	2.660	0.412	29.41	62.50	0.400	0.510

Table 7.6: Summary of the performance metrics of the complete model of **Approach A** and baseline solution, using the Neuraspace dataset. The shaded cells highlight the best score for each metric.

It can be seen that the complete model for **Approach A** outperforms the baseline solution in all the metrics (except for the recall which is the same for both models), using the Neuraspace dataset, even though that only the *risk* feature is used and some approximations are made. In addition, it is important to note that the Neuraspace dataset contains real data which further increases the difficulty of the problem. Even so, the implemented model managed to outperform the baseline solution, which is currently used as the risk predictor in most collision avoidance operations, emphasizing the good results obtained in this Section and further adding to the idea that this method should be further explored.

The confusion matrices for the implemented model and baseline approach are represented in Tables 7.7 and 7.8, respectively.

		Predicted	
		Low-risk	**High-risk**
True	**Low-risk**	6851 (TN)	11 (FP)
	High-risk	3 (FN)	5 (TP)

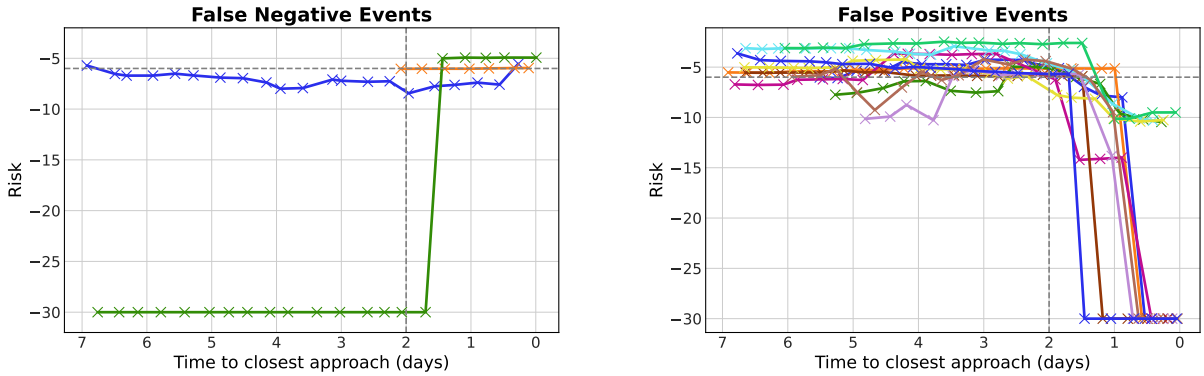
Table 7.7: Confusion matrix for the risk predictions of the implemented model, using the Neuraspace dataset, following **Approach A**.

		Predicted	
		Low-risk	**High-risk**
True	**Low-risk**	6850 (TN)	12 (FP)
	High-risk	3 (FN)	5 (TP)

Table 7.8: Confusion matrix for the risk predictions of the baseline solution, using the Neuraspace dataset.

The confusion matrices show that the classification performance of both models is very similar, where the implemented model only has one less false positive. The events that were wrongly classified as

low-risk (false negatives) are the same for both models and the miss-classified low-risk events (false positives) by the implemented model were also wrongly classified by the baseline. The risk evolution of these events is represented in Figure 7.5.



(a) Risk evolution of false negatives.

(b) Risk evolution of false positives.

Figure 7.5: Risk evolution of the events that were wrongly classified by both models as low-risk (left) and as high-risk (right), following **Approach A**, with the Neuraspace dataset. The colored lines represent different events and the crosses mark the risk updates.

Figure 7.5a shows that only one miss-classified high-risk event has a significant increase of the risk value after the cut-off threshold, in which the risk jumps from –30 to approximately –5. This event has a baseline solution of –30, indicating that the event did not go through the prediction process of the HMM, which further adds to the idea that future work should focus on the development of a more robust model that can identify the –30 events without any miss-classification of high-risk conjunctions. The other FN events actually have a risk evolution in the vicinity of the high-risk threshold. In addition, it can be seen that the event represented in orange only has one CDM before the cut-off time, suggesting that, in order to make reliable predictions, a minimum number of observations should be available before the cut-off threshold. To better analyze the source of the FN errors, more data would be needed, since that three events are not a sufficiently large sample to take conclusions from. From Figure 7.5b, it is possible to see that the majority of the events that correspond to the false positives have high-risk values before the cut-off time and, after this threshold, the risk experiences an extreme jump to low values, which could not be forecast. In addition, the right plot of Figure 7.5 events shows an interesting characteristic: most of the false positive events end up having a final risk of –30. The complex and unpredictable behavior of the risk transitions suggests, again, that more features should be analyzed by the ML model.

To better visualize how far the predictions are from the true values, Figure 7.6 shows the prediction results over the true risk values for both the model of **Approach A** and the baseline, using the Neuraspace dataset. Figure 7.6 shows that the predictions made by the implemented model tend to be arranged in 9 steps, which correspond to the sampled values for the mean of the emission distributions. Just like in the predictions made by the implemented model for the CAC dataset, the predictions are not completely aligned with the dashed diagonal line, but they tend to be close to the corresponding true risk values. In addition, it can also be concluded that the HMM would benefit from more data containing a higher proportion of high-risk values since the HMM predictions seem to be truncated at an upper threshold. From the two plots, it can also be seen a big over-prediction of –30 events. The true evolution

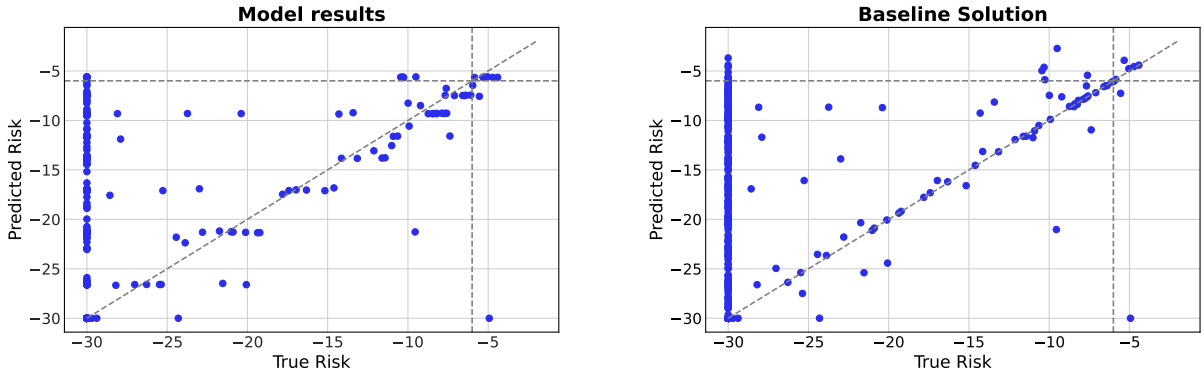


Figure 7.6: Prediction results for the model of **Approach A** (left) and for the baseline (right), using the Neuraspace dataset. The diagonal dashed line represents the values where predictions of the risk are equal to the true values and the horizontal and vertical lines mark the high-risk threshold for the predictions and for the true risk, respectively.

of the risk values within some of the events that were over-predicted by both models and that represent the typical evolution of the risk updates within the over-predicted events is represented in Figure 7.7. It is possible to verify that all of the events have extreme jumps in risk values from high values to –30, after the prediction threshold. A closer look into the predictions shows that the implemented model reduces the number of over-predicted events with a final risk of –30 by approximately 7.7% when compared to the baseline.

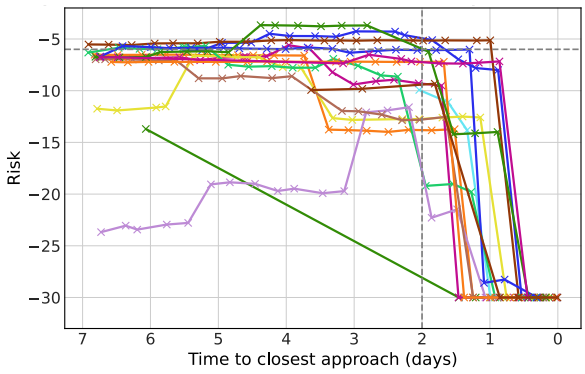
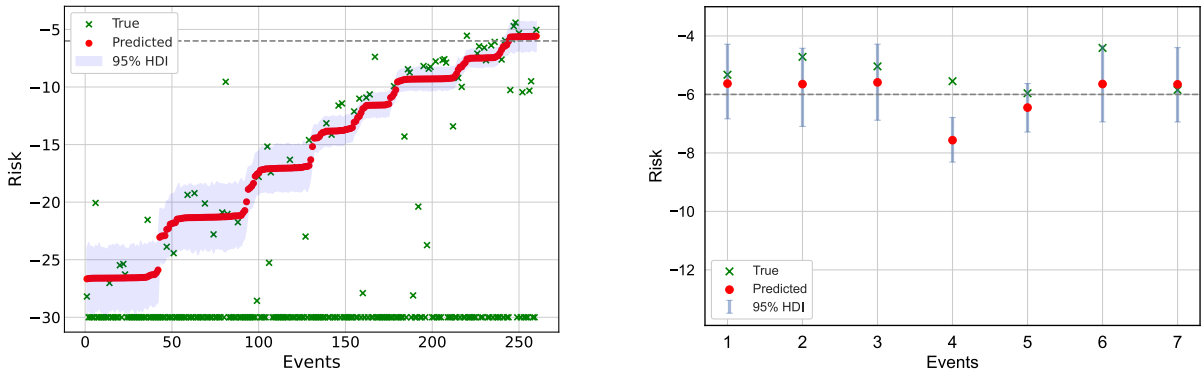


Figure 7.7: Typical time series evolution of the collision risk of the events that were over-predicted by the complete model of Approach A and the baseline, using the Neuraspace dataset. It can be seen that all the events experience a big risk transition from higher values to –30, after the cut-off time.

Since a distribution for the risk prediction of each event is obtained from the HMM, an interval that reflects the uncertainty regarding the predicted risk can be provided. Figure 7.8 shows the 95% HDI as a blue shaded area, the true risk values as green markers, and the predictions as red markers, for all the events that went through the prediction process of the HMM and for the high-risk events. In Figure 7.8a, the events are organized in ascending order regarding the value of the predicted risk.



(a) All events.

(b) True high-risk events.

Figure 7.8: Representation of the true risk values of each event (green), the predictions (red) and the 95% HDI area (area shaded in blue), for all the events that went through the prediction process of the HMM (left) and for the high-risk events (right), following **Approach A** and using the Neuraspace dataset.

Just like in the HMM predictions for the CAC dataset, it can be verified that, in each prediction step (that corresponds to the mean of the sampled mean values for the emission distributions — see Table B.1.2 from Appendix B), the 95% HDI approximately covers the risk values that deviate two standard deviations from the mean values of the corresponding Truncated Normal emission distributions. Figure 7.8a shows that most of the prediction intervals do not cover the true risk values. It can again be seen that the –30 events are largely over-predicted and that the 95% HDI doesn't capture the true risk of those conjunctions. For the other events, the risk intervals cover approximately 49.3% of the true risk values. From the plot of Figure 7.8b it can be taken that the 95% HDI intervals capture the large majority of the high-risk values and, in one of the two cases where the HMM fails to identify the high-risk conjunctions, the risk interval captures the true risk value. Notice that, in the test set, there are 8 true high-risk conjunctions, but in the plot of Figure 7.8b only 7 are represented. This is due to the fact that one of the high-risk conjunctions did not go through the prediction process of the HMM (because its baseline solution is –30) and its risk was directly predicted as –30, meaning that a prediction interval could not be provided.

7.2 Approach B

In this section, the results obtained with the proposed model for **Approach B** — model and predict the chaser position uncertainties and recompute the risk of collision — are presented. Recall that the CAC dataset cannot be used for this approach, because some necessary features are not released to the public, in order to protect confidential information (read Section 4.1 for more details). Hence, the results presented hereafter are obtained with the data from the Neuraspace dataset.

This section is divided into three parts: Section 7.2.1 shows the cross-validation results obtained for the three separate HMMs that were trained to learn the evolution of the three components of the position standard deviations of the chaser object (radial, along-track and normal), Section 7.2.2 presents the position errors predictions obtained with the final HMMs and the comparison with the baseline solution, and, in Section 7.2.3, the risk of collision is recomputed and the performance of the complete model for

Approach B is evaluated by comparing the results with the true final risk of each event.

7.2.1 Stratified Cross-Validation Results

To find the best number of possible states for the three separate HMMs, the value of K is iterated between 4 and 8 states, for each HMM, using cross-validation, where 3 chains of 2 000 iterations are sampled in each fold. A higher number of possible states, for each model, leads to chains converging into different values, which means that, in those cases, the posterior has different modes with sharp density curvatures.

Like in the previous Section, the values of the parameters contained in the next CDM/observation of each event are predicted and compared to the true value of the position errors (which are taken from the last released CDM, for each event). To propagate the posterior uncertainty into the predictions, 1 200 posterior draws are taken from the sampled parameter values and are given as input to the predictive distribution, for each HMM, and the final prediction is the mean of the respective distribution.

The performance metrics obtained in cross-validation for the HMMs used to learn the evolution of the radial (σ_R^{chaser})chaser along-track (σ_T^{chaser}) and normal (σ_N^{chaser}) standard deviations are presented in Tables 7.9, 7.10 and 7.11, respectively, in which the shaded cells highlight the best score for each metric. Recall that the true values of the standard deviations are considered the values contained in the last released CDM of each event.

States	RMSE	MAE
4	129.79	73.08
5	126.70	70.24
6	127.21	68.42
7	129.49	69.21
8	131.61	69.60

Table 7.9: Summary of the cross-validation results obtained with the HMM for the chaser radial position error evolution, using the Neuraspace dataset. The values of the RMSE and MAE are expressed in meters.

States	RMSE	MAE
4	12677.3	6732.8
5	12745.1	6584.4
6	11340.4	5633.9
7	6381.1	2242.0
8	6474.2	2123.2

Table 7.10: Summary of the cross-validation results obtained with the HMM for the chaser along-track position error evolution, using the Neuraspace dataset. The values of the RMSE and MAE are expressed in meters.

States	RMSE	MAE
4	52.11	32.00
5	49.12	29.51
6	48.79	28.33
7	46.59	26.56
8	47.84	26.70

Table 7.11: Summary of the cross-validation results obtained with the HMM for the chaser normal position error evolution, using the Neuraspace dataset. The values of the RMSE and MAE are expressed in meters.

Regarding the HMM for the evolution of the parameter σ_T^{chaser}, the models with K equal to 7 and 8 have lower RMSE and MAE when compared to the other HMMs with fewer hidden states. The HMM with K equal to 8 has a lower MAE and the model with 7 possible hidden states has a better RMSE. Since the RMSE penalizes large errors more than MAE, it is considered that the RMSE has more value than the MAE. Hence, the HMM with K equal to 7 is chosen as the final HMM for the evolution of σ_T^{chaser}. The choice of the number of possible states for the HMM for the parameter σ_R^{chaser} is similar to the HMM for the σ_T^{chaser} evolution: the models with K equal to 5 or 6 have better performance metrics than the others, but the chosen number of possible states for the HMM used to learn the σ_R^{chaser} evolution is 5,

because it has a lower RMSE. As for the HMM for the evolution of σ_N^{chaser}, a number of 7 possible states is chosen, because it has lower MAE and RMSE when compared to the other HMMs.

7.2.2 Results of the standard deviations predictions

The final three HMMs are then trained in the entire training set and 5 chains of 2 000 iterations are sampled from the posterior distribution. To obtain predictions, 2 000 posterior draws are randomly taken from the sampled chains and are given as input to (2.23), creating a distribution that reflects the uncertainty of the predictions. The final predicted value for each standard deviation of each event is the mean of the respective distribution.

The performance metrics obtained, on the test set, with the three HMMs for the standard deviations predictions as well as the baseline solution are shown in Tables 7.12, 7.13 and 7.14, where the shaded cells highlight the best result for each metric. The baseline predictions of the standard deviations are obtained from the last released CDM before the cut-off time, for each event.

Model	RMSE	MAE
σ_R^{chaser}	143.64	77.71
Baseline	144.15	63.31

Table 7.12: Summary of the performance metrics of the HMM and baseline solution for the prediction of σ_R^{chaser}. All the values are expressed in meters.

Model	RMSE	MAE
σ_T^{chaser}	12844.8	6663.3
Baseline	13452.3	6175.0

Table 7.13: Summary of the performance metrics of the HMM and baseline solution for the prediction of σ_T^{chaser}. All the values are expressed in meters.

Model	RMSE	MAE
σ_N^{chaser}	52.27	27.56
Baseline	46.45	20.91

Table 7.14: Summary of the performance metrics of the HMM and baseline solution for the prediction of σ_N^{chaser}. All the values are expressed in meters.

It can be seen that the predictions of the HMMs for σ_R^{chaser} and σ_T^{chaser} yield a lower RMSE but a higher MAE, when compared to the baseline, meaning that the implemented models have smaller errors. Table 7.14 shows that the baseline outperforms the HMM for σ_N^{chaser} in all metrics. The difference in scale of the prediction errors of the three models shows that the along-track component yields the largest errors.

Figures 7.9, 7.10 and 7.11 show the HMM predictions over the true values of each parameter (on the left plot) and the corresponding baseline solution over the true values (on the right plot).

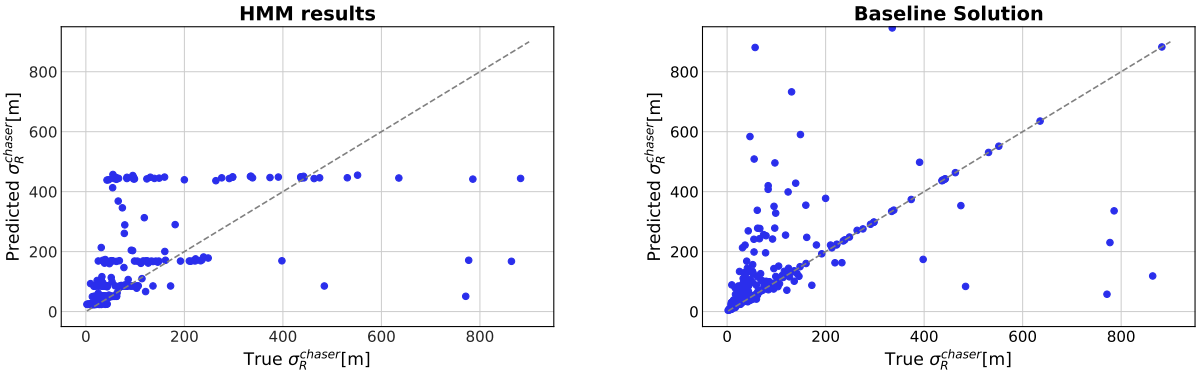


Figure 7.9: Prediction results for the radial position uncertainty σ_R^{chaser} (left) and for the baseline (right), over the true parameter values. The diagonal dashed line represents the values where the predictions are equal to the true values.

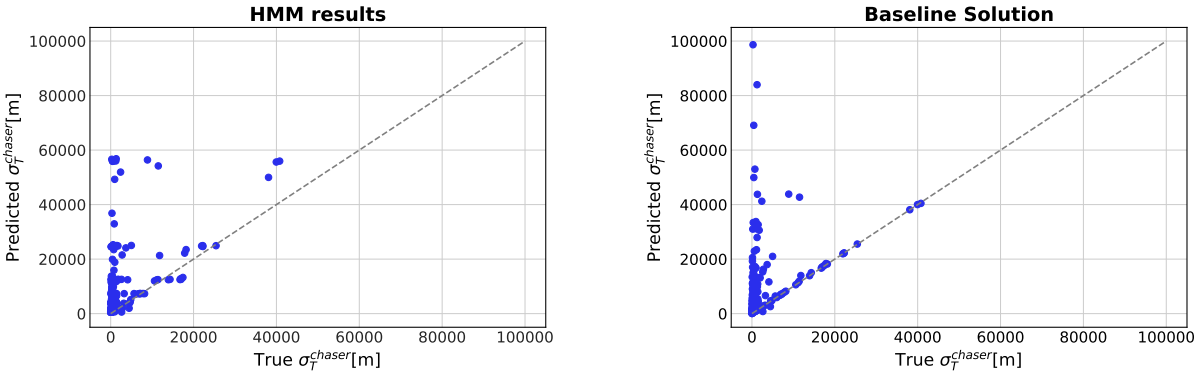


Figure 7.10: Prediction results for the along-track position uncertainty σ_T^{chaser} (left) and for the baseline (right), over the true parameter values. The diagonal dashed line represents the values where the predictions are equal to the true values.

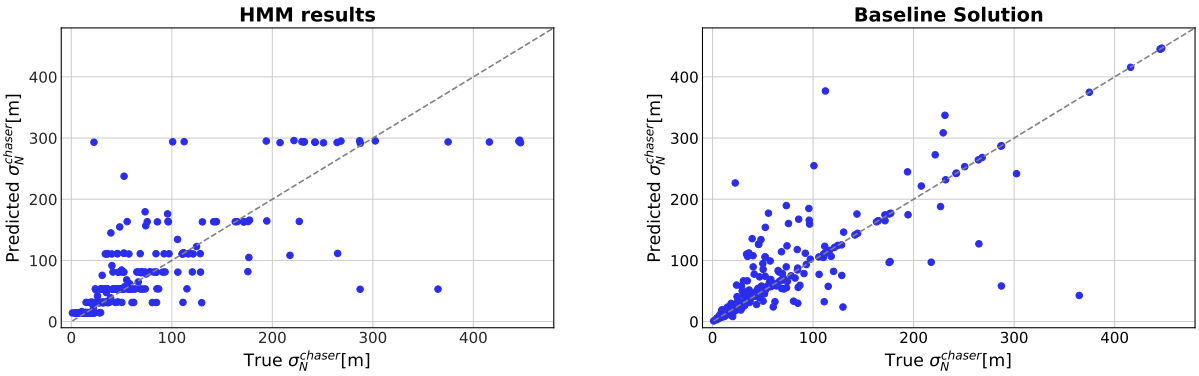


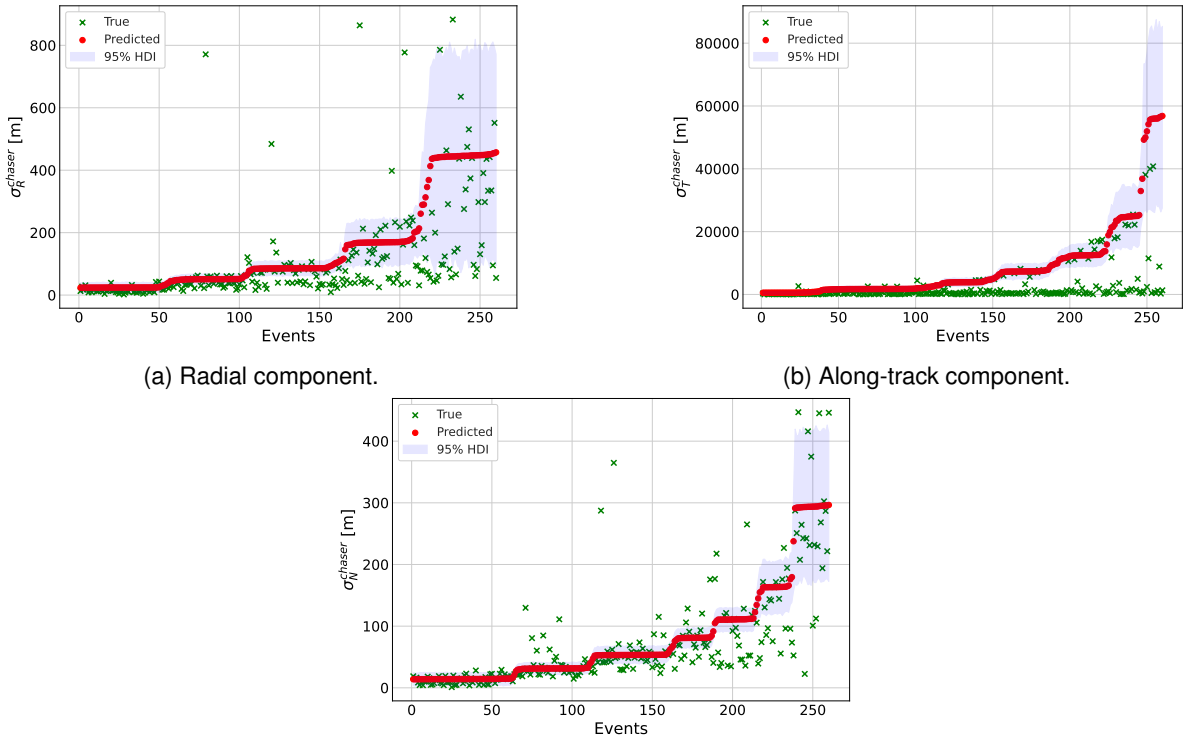
Figure 7.11: Prediction results for the normal position uncertainty σ_N^{chaser} (left) and for the baseline (right), over the true parameter values. The diagonal dashed line represents the values where the predictions are equal to the true values.

The difference in scale between the figures highlights, again, that the along-track component of the position uncertainty of the chaser object is significantly higher than the radial and normal components.

All of the HMM prediction plots show that the predictions tend to be arranged near the sampled mean values of the emission distributions of the respective HMM (as can be verified by Tables B.3, B.4 and B.5 of Appendix B). Additionally, the plots of the chaser along-track position uncertainty predictions made by the HMM and by the baseline, presented in Figure 7.10, show that the along-track position errors get significantly over-predicted for true values of σ_T^{chaser} lower than, approximately, 10 000 m - the source of these over-predictions are addressed at the end of this Section. The results of the HMM for the along-track position uncertainty seem to show that the implemented model reduces the errors of those over predictions when compared to the baseline solution which justifies its lower RMSE since that metric penalizes large errors more. However, overall, the results of the baseline are closer to the dashed line that represents the line of the correct predictions, justifying its lower MAE. The same reasoning can be applied to the radial position uncertainty predictions: both models (HMM and baseline) over-predict the lower true values of σ_R^{chaser}, but the over-predictions made by the HMM are lower, which justifies it lower RMSE. However, the baseline predictions are closer to the true regression line, overall, justifying its lower MAE.

In addition, it can be seen from the HMM results presented in Figures 7.9, 7.10 and 7.11 that the predictions seem to be “truncated” at an upper threshold for all the HMMs. The values of these upper thresholds correspond, approximately, to the maximum values of the sampled mean of the emission distributions for each parameter, as can be seen by Tables B.3, B.4 and B.5 of Appendix B. These results suggest that more data containing higher values for the standard deviation would be needed.

Figures 7.12a, 7.12b and 7.12c show the 95% HDI (blue shaded area) associated with each prediction, the true parameter values (green markers) and the predictions made by the HMMs (red markers), for the chaser radial, along-track and normal components of the position uncertainties, respectively. In these plots, the events are organized in ascending order regarding the prediction values. In all of the figures, it can be seen, again, that within each prediction step the prediction intervals are very similar and correspond, approximately, to a deviation of two standard deviations from the mean values of the corresponding emission distributions. The 95% HDI of the HMM for the evolution of σ_T^{chaser}, σ_R^{chaser} and σ_N^{chaser} cover 27.3%, 61.2% and 66.9% of the corresponding true parameter values, respectively.



(a) Radial component.

(b) Along-track component.

(c) Normal component.

Figure 7.12: Representation of the true position error values of each event (green), the predictions (red) and the 95% HDI area (area shaded in blue), for the three position uncertainty components, following **Approach B**.

In addition, the plot of Figure 7.12b highlights the significant over-predictions of the events with a final value of σ_T^{chaser} lower than 10 000 m and shows that, in those cases, most of the HMM prediction intervals do not capture the true values. The evolution of the chaser along-track position uncertainty within the events that have been largely over-predicted by the HMM and the baseline is shown in the plot of Figure 7.13.

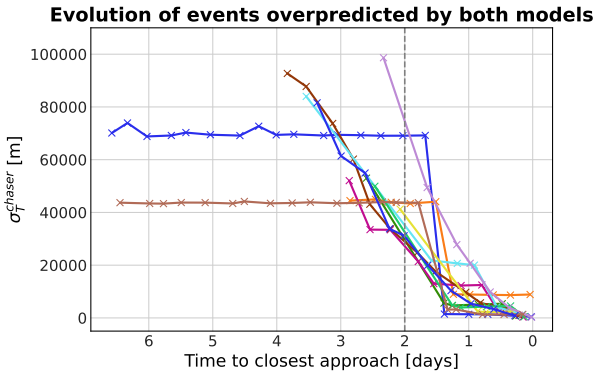


Figure 7.13: Time series evolution of the chaser along-track prediction errors for the largely over-predicted events by the complete model of **Approach B** and the baseline, using the Neuraspace dataset.

It is possible to see that all over-predicted events have a chaser along-track standard deviation greater than approximately 30 000 m before the cut-off time and, after this threshold, the position error evolves into much lower values, which could not be captured by the HMM. This is likely due to the fact that the Neuraspace dataset contains only a small percentage of CDMs with high along-track position errors, which shows that only 5% of the collision warnings have a value of σ_T^{chaser} greater than approximately 16 000 m. Since there are few events with a high value for σ_T^{chaser}, their position error updates cannot be captured by the HMM, suggesting that more data (with events containing chaser objects with high position errors) would be needed. In addition, it can be seen from the plot of Figure 7.13 that, in some events, the σ_T^{chaser} updates stay almost at a constant value and, after the cut-off threshold, experience a significant drop, which cannot be foreseen by the HMM that predicts the position error to stay at that value. The events with this type of σ_T^{chaser} evolution are rare within the Neuraspace dataset and, consequently, their behavior cannot be forecast by the implemented model. To account for these events, more data would be needed.

As explained in Chapter 4, the covariance predictions are used to forecast the risk of collision, using Akella's and Alfriend's equation. Although the HMM for the evolution of σ_N^{chaser} yields poor results in comparison to the baseline, it will still be used to obtain the risk of collision, in order to benchmark the performance of the complete model of **Approach B**. However, it is important to note that, in real-life applications, the predictions of this model would not be used to compute the risk, since it has shown worse performance than the baseline, and the values of σ_N^{chaser} would be taken from the last released CDM before the cut-off time.

In the next Section, the risk computation results obtained with the complete model of **Approach B** are presented.

7.2.3 Results of the risk computation

Recall from Chapter 4 that to obtain the risk predictions, in **Approach B**, the following steps are needed:

- if the baseline solution for the event is –30, then directly predict the final risk as –30;

- take target information and chaser position and correlation coefficients from last available CDM before cut-off time;
- predict chaser standard deviations and compute combined covariance matrix;
- give the previous parameters as input to (2.14) and compute final risk.

In each event, a distribution for the predictions of each component of the positional standard deviations of the chaser is obtained and the final predicted values for the three parameters are the mean of the corresponding distributions, which are used to build the chaser covariance matrix that is combined with the target covariance matrix and the result is given as input to the Akella's and Alfriend's equation. After computing the risk, in each event, the performance of the complete model is evaluated by comparing the obtained result with the risk value contained in the last released CDM.

Table 7.15 presents the performance metrics obtained for the risk prediction problem, with the complete model of **Approach B** (which is schematized in Figure 4.3a) and the baseline solution, on the test set. The confusion matrices obtained with the implemented model and the baseline are also shown in Tables 7.16 and 7.17, respectively. Note that the same split was performed on the Neuraspace dataset for **Approach A** and **Approach B** so that the results obtained with the different models could be easily compared.

Model	RMSE	MAE	Precision (%)	Recall (%)	$\mathbf{F}_1$	$\mathbf{F}_2$
Approach B	2.879	0.455	18.52	62.50	0.286	0.424
Baseline	2.660	0.412	29.41	62.50	0.400	0.510

Table 7.15: Summary of the performance metrics of the complete model of **Approach B** and baseline solution, for the risk prediction problem. The shaded cells highlight the best score for each metric.

		Predicted	
		Low-risk	**High-risk**
True	**Low-risk**	6840 (TN)	22 (FP)
	High-risk	3 (FN)	5 (TP)

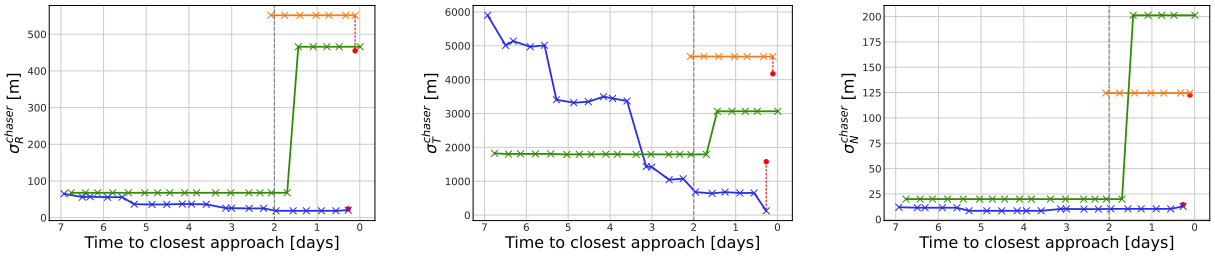
Table 7.16: Confusion matrix for the risk predictions of the implemented model following **Approach B**.

		Predicted	
		Low-risk	**High-risk**
True	**Low-risk**	6850 (TN)	12 (FP)
	High-risk	3 (FN)	5 (TP)

Table 7.17: Confusion matrix for the risk predictions of the baseline solution, using the Neuraspace dataset.

Table 7.15 shows that the complete model of **Approach B** is outperformed by the baseline solution in all metrics and, consequently, it is also outperformed by the model of **Approach A**, whose results are presented in Table 7.6. This is due to the fact that the baseline predictions for the position errors are closer to the true values, when compared to the predictions of the HMMs, as shown by Figures 7.9, 7.10 and 7.11. Thus, the risk predictions made by the baseline are also closer to the true risk values. To improve these results, it could be useful to train the models with more data and test the performance of the HMMs with a higher number of possible hidden states. In future work, it could also be advantageous to use more informative priors that take into consideration the physics of the problem.

The confusion matrices presented in Tables 7.16 and 7.17 show that the complete model of **Approach B** has 10 more FPs than the baseline, which justifies its lower precision metric. The events that generated the FN predictions are the same for both models and are also wrongly predicted as low-risk conjunctions by the model of **Approach A**. The true evolution of the radial, along-track and normal position errors of the FN events is represented in Figures 7.14a, 7.14b and 7.14c, respectively. In the figures, the red markers identify the HMM predictions for each event and the dashed red line connects the predicted values to the corresponding true position errors. Notice that the event represented in green does



(a) Radial component. (b) Along-track component. (c) Normal component.

Figure 7.14: True evolution of the radial, along-track and normal position errors of the FN events. The red markers identify the HMM predictions for each event and the dashed red line connects the predicted value to the corresponding true position error.

not have any associated prediction because the event has a baseline solution of –30 and, hence, it does not go through the prediction process of the HMM. Analyzing the plots of Figures 7.14a, 7.14b and 7.14c, there is not any visible trend within the evolution of the position errors of the FN events. However, three events are not a representative sample to take conclusions from, suggesting that future research would benefit from using more data containing more high-risk conjunctions (if possible). By comparing these plots to the true risk evolution of the events wrongly classified as low-risk by the model of **Approach A** (which is shown in Figure 7.5a), using the Neuraspace dataset, some interesting observations can be made. The lines with the same color correspond to the same event, in all plots, and it can be seen that the risk of the event in orange remains almost constant due to the fact that the position errors of the chaser (which are considered the main contributors for the risk of collision) also remain with the same value in all updates of that event. The evolution of the three position error components of the event in green suffers a significant transition to higher values, after the cut-off time, which justifies the extreme transition in the risk value.

The plots of Figure 7.15 show the predicted risk values over the true final risk values for the events in the test set, using the complete model of **Approach B** and the baseline solution.

It can be seen that the baseline predictions are closer to the true regression line, especially for true values of risk smaller than –15. It is also possible to verify that, as in the results of **Approach A**, the –30 events are largely over-predicted.

Since that for the predictions of the parameters σ_T^{chaser}, σ_R^{chaser} and σ_N^{chaser} of each event (that has a baseline solution different from –30) a distribution of possible final values is obtained, it is possible to provide an interval of possible risk values that expresses the trustworthiness of the predictions made for the risk prediction. In order to obtain a risk distribution, the following steps are performed, in each

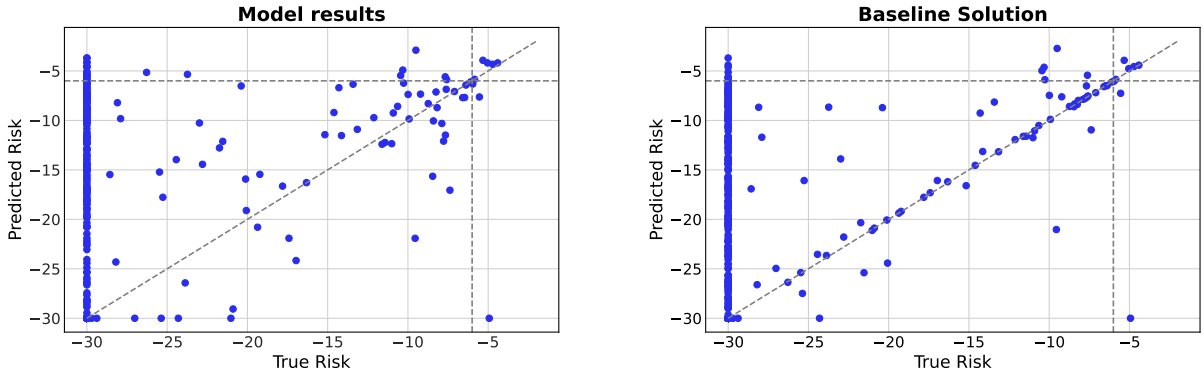
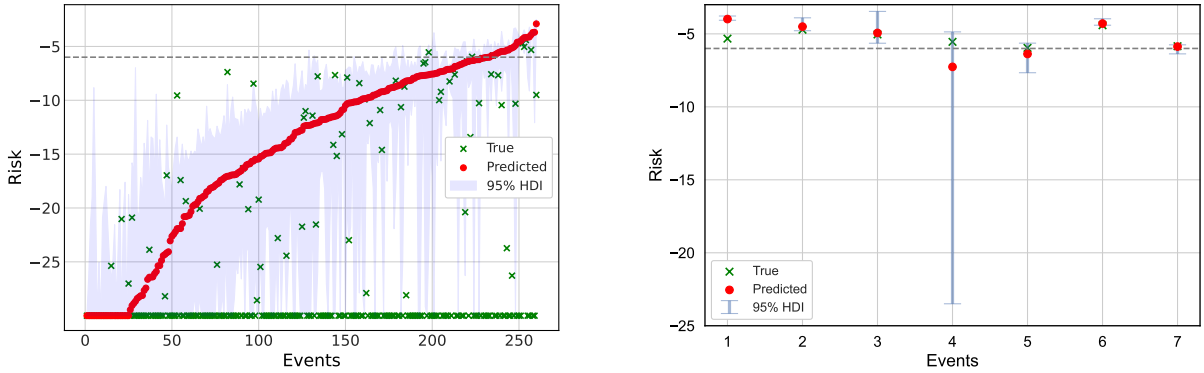


Figure 7.15: Prediction results for the model of **Approach B** (left) and for the baseline (right), using the Neuraspace dataset. The diagonal dashed line represents the values where predictions of the risk are equal to the true values and the horizontal and vertical lines mark the high-risk threshold for the predictions and for the true risk, respectively.

event of the test set: (i) from each prediction distribution of σ_T^{chaser}, σ_R^{chaser} and σ_N^{chaser}, take 2 000 random samples and build 2 000 covariance matrices for the chaser position uncertainties; (ii) take the target information and chaser position and correlation coefficients from the last available CDM before cut-off time; (iii) compute the final risk using Akella's and Alfriend's equation for the 2 000 possible chaser covariance matrices. After obtaining a risk distribution for each event, it is possible to compute a credible interval for each prediction. Figure 7.16 shows the 95% HDI (blue area) associated with each risk prediction (red markers) as well as the true risk value for all the events that went through the prediction process of the HMMs (Figure 7.16a) and for the high-risk events (Figure 7.16b). In the plot of Figure 7.16a the events are arranged in ascending order regarding the value of the predicted risk.



(a) All events.

(b) True high-risk events.

Figure 7.16: Representation of the true risk values of each event (green), the predictions (red) and the 95% HDI area (area shaded in blue), for all the events that went through the prediction process of the HMM (left) and for the high-risk events (right), following **Approach B** and using the Neuraspace dataset.

Figure 7.16a highlights the large over-predictions of the –30 events, in which only 58% of those events are covered by the risk intervals. In addition, it can be seen that there is a lot of uncertainty associated with the predictions. Figure 7.16b shows that the majority of the high-risk events are captured by the 95% HDI and the high true risk values that are not covered by the risk intervals are predicted as high-risk, anyway.

Notice that, like in the results of the model of **Approach A** using the Neuraspace dataset, there

are 8 true high-risk conjunctions in the test set, but in the plot of Figure 7.16b only 7 are represented. This, again, is due to the fact that one of the high-risk conjunctions has a baseline solution of –30 and, consequently, did not go through the prediction process of the HMM, meaning that a prediction interval could not be provided.

The results obtained with **Approach B** have shown a poorer performance when compared to the baseline solution and the complete model of **Approach A**. These results show that, for this class of models, the evolution of the risk is easier to predict.

To improve these results, future research may focus on the inference of HMMs with more informative priors and with a larger number of possible hidden states. The training of the models with a larger dataset could also improve the obtained results. In addition, it can also be advantageous to implement multivariate HMMs that take into account other features that have a strong statistical relationship with the position errors.

Chapter 8

Conclusions

The major contribution of this work was the development of Bayesian HMMs that evaluate the time-series evolution of some parameters contained in the CDMs and allow for the investigation of one of the conclusions taken from the CAC results: the time-series of the collision warnings in a given conjunction event approximately follows the Markov property. In this thesis, the performance of HMMs was established by following two different approaches: directly model and predict the risk of collision (**Approach A**), and predict the chaser position errors and compute the risk using Akella's and Alfriend's formula (**Approach B**). To provide benchmark solutions, each implemented HMM only learned the evolution of one single feature of the dataset, i.e., in **Approach A**, only the risk feature was analyzed by the HMM and, in **Approach B**, three separate HMMs were used to learn the evolution of the three components of the position errors of the chaser object. Thus, this work provides a foundation for future research regarding the implementation of Bayesian HMMs in collision risk estimation.

Although **Approach B** yields poor results when compared to the baseline solution, **Approach A** outperformed the naive forecast (which is considered a very strong predictor for this problem and is used as the risk estimator in most collision avoidance operations) in all metrics, for both the CAC and private Neuraspace datasets. The data imbalance problem and the complex behavior of the risk updates within the events pose to be the main difficulty of the collision risk estimation. Additionally, the fact that real data was used, further adds to the complexity of the problem. Despite those challenges, the implemented model improved the baseline solution, which highlights the good results obtained in this thesis. The fact that some approximations were made in this work (e.g., manipulating the training events during the data setup and only predicting the next observation of each event) and that only one feature was used (the risk, which has an extremely imbalanced distribution) further emphasizes the good performance of Bayesian HMMs in the risk prediction problem. The promising results further add to the idea that the CDMs may follow the Markov property and suggest that this is a powerful method that should be further explored.

By using Bayesian statistics, prediction intervals that reflect the uncertainty of the sampled posterior can be provided. It was verified that, in most cases, the risk intervals could not capture the true risk values, due to the complexity of the problem and due to the fact that simple univariate HMMs were used.

8.1 Future Work

In this work, the events that showed a baseline solution for the risk of collision equal to –30 were not used for training and were directly predicted as having a final value of –30. As already mentioned, this approach is not perfect since, in the Neuraspace dataset, it misclassified two high-risk events (read Section 4.2.1). Future work should tackle this problem, in which a more robust model that is capable of correctly identifying the –30 events without any miss-classification of high-risk conjunctions should be developed.

Furthermore, in the implemented HMMs, only a small range of possible states was tested, due to the fact that, for a higher number of possible states, the posterior is multimodal and the MCMC sampler used in this thesis cannot explore all the density areas of multimodal posterior distributions with sharp density curvatures. However, some of the cross-validation results suggested that it could be beneficial to test the performance of HMMs with a higher number of possible hidden states, so, as future work, it could be advantageous to use/develop an efficient sampler that is based on Hamiltonian dynamics (in order for the sampled values to have low autocorrelation and for the proposed values to have a high acceptance rate) and is capable of handling multimodality.

In addition, this thesis benchmarked the performance of HMMs in the collision risk estimation problem by only predicting the values contained in the next observation/CDM of the events, after the cut-off time. However, as previously mentioned, in real-life operations, 3 CDMs are received, on average, per day, and the last CDM received is assumed to be the best knowledge about the outcome of that event. Therefore, since the defined cut-off time is 2 days before the TCA, instead of just predicting the next observation/CDM received after that threshold — $\mathbf{x}_{N+1}$ — it can be advantageous to study the possibility of predicting k observations ahead — $\mathbf{x}_{N+k}$. Additionally, in this work, the predictive distribution shown in (2.23) was approximated by a truncated Normal distribution with mean value and variance equal to the first and second momentum of the original distribution, respectively. Future work should focus on the computation of the true predictive distribution for the HMM.

Finally, it would also be interesting to use multivariate emission distributions, in order to explore the relations between the features of the dataset in one single HMM. For that, it could be considered multivariate Normal emission distributions, in which the mean values could be specified by Normal (or Truncated normal) priors and the prior for the covariance matrix could be given by the Lewandowski-Kurowicka-Joe (LKJ) distribution [71] (combined with priors on the standard deviations of each component). The LKJ is a distribution for positive definite correlation matrices that, combined with priors on the standard deviations of each component, can induce a prior for the covariance matrix of the multivariate Normal distributions. With these priors, HMMs with multivariate Normal emission distributions can be inferred and, hence, the relation between different features can be explored in one single HMM. In addition, the performance of HMMs with Bayesian regressions as the emission distributions should also be tested.

Bibliography

[1] ESA's Space Debris Office at ESOC, Darmstadt. Space debris by the numbers. https://www.esa.int/Safety_Security/Space_Debris/Space_debris_by_the_numbers, 2022.

[2] ESA Space Debris Office. ESA'S ANNUAL SPACE ENVIRONMENT REPORT. LOG GEN-DB-LOG-00288-OPS-SD, ESA, April 2022.

[3] T. Kelso. Analysis of the Iridium 33-Cosmos 2251 Collision. *Proceedings of the 19th AIAA/AAS Astrodynamics Specialist Conference*, 135, 2009.

[4] J. Radtke, C. Kebschull, and E. Stoll. Interactions of the space debris environment with mega constellations—using the example of the oneweb constellation. *Acta Astronautica*, 131:55–68, 2017. ISSN 0094-5765. doi: 10.1016/j.actaastro.2016.11.021.

[5] World Economic Forum. The Global Risks Report 2022 - 17th Edition, 2022.

[6] NASA. Space debris and human spacecraft. https://www.nasa.gov/mission_pages/station/news/orbital_debris.htm, May 2021.

[7] European Space Agency. The history of space debris creation. https://www.esa.int/ESA_Multimedia/Images/2021/03/The_history_of_space_debris_creation, March 2021.

[8] C. Pardini and L. Anselmo. Assessment of the consequences of the Fengyun-1C breakup in low Earth orbit. *Advances in Space Research 44 (2009) 545–557*, 2009. doi: 10.1016/j.asr.2009.04.014.

[9] European Space Agency. Spatial density of objects by orbital altitude. https://www.esa.int/ESA_Multimedia/Images/2019/10/Spatial_density_of_objects_by_orbital_altitude, October 2019.

[10] European Space Agency. The current state of space debris. https://www.esa.int/Space_Safety/Space_Debris/The_current_state_of_space_debris, October 2020.

[11] European Space Agency. Hypervelocity impacts and protecting spacecraft. https://www.esa.int/Space_Safety/Space_Debris/Hypervelocity_impacts_and_protecting_spacecraft#.Y0se3TtXe-U.link, n.d..

[12] D. Kessler and B. Cour-Palais. Collision frequency of artificial satellites: The creation of a debris belt. *Journal of Geophysical Research: Space Physics*, 83(A6):2637–2646, 1978. doi: 10.1029/JA083iA06p02637.

[13] European Space Agency. The Kessler Effect and how to stop it. `https://www.esa.int/Enabling_Support/Space_Engineering_Technology/The_Kessler_Effect_and_how_to_stop_it`, n.d..

[14] R. Jehn. Dispersion of debris clouds from on-orbit fragmentation events. *ESA Journal*, 15(1):63–77, 1991.

[15] D. Wright. Space Debris. *Physics Today*, 60(10):35–40, 2007. doi: 10.1063/1.2800252.

[16] Q. Funke, B. B. Virgili, V. Braun, T. Flohrer, H. Krag, S. Lemmens, F. Letizia, K. Merz, and J. Siminski. Operational support to collision avoidance activities by ESA's space debris office. *CEAS Space Journal*, 8(3):177–189, 2016. doi: 10.1007/s12567-016-0119-3.

[17] T. Uriot, D. Izzo, L. Simões, R. Abay, N. Einecke, S. Rebhan, J. Martinez-Heras, F. Letizia, J. Siminski, and K. Merz. Spacecraft collision avoidance challenge: Design and results of a machine learning competition. *Astrodyn*, 6:121–140, 2022. doi: 10.1007/s42064-021-0101-5.

[18] European Space Agency. AI challenged to stave off collisions in space. `https://www.esa.int/Enabling_Support/Space_Engineering_Technology/AI_challenged_to_stave_off_collisions_in_space`, October 2019.

[19] B. B. Virgili, T. Flohrer, H. Krag, K. Merz, and S. Lemmens. CREAM - ESA's Proposal for Collision Risk Estimation and Automated Mitigation. In *Advanced Maui Optical and Space Surveillance Technologies Conference*, page 57, 2019.

[20] ESA Advanced Concepts Team. Kelvins Collision Avoidance Challenge. `https://kelvins.esa.int/collision-avoidance-challenge`, 2019.

[21] M. Akella and K. Alfriend. Probability of Collision Between Space Objects. *Journal of Guidance, Control, and Dynamics*, 23(5):769–772, 2000. doi: 10.2514/2.4611.

[22] J. Mueller and A. Thyagarajan. Siamese Recurrent Architectures for Learning Sentence Similarity. *Proceedings of the AAAI Conference on Artificial Intelligence*, 30(1), 2016. doi: 10.1609/aaai.v30i1.10350.

[23] S. Metz. Implementation and comparison of data-based methods for collision avoidance in satellite operations. Master's thesis, Technische Universität Darmstadt, Germany, 2020.

[24] G. Acciarini, F. Pinto, S. Metz, S. Boufelja, S. Kaczmarek, K. Merz, J. A. Martinez-Heras, F. Letizia, C. Bridges, and A. G. Baydin. Spacecraft Collision Risk Assessment with Probabilistic Programming. *Third Workshop on Machine Learning and the Physical Sciences (NeurIPS 2020)*, 2020. doi: 10.48550/ARXIV.2012.10260.

[25] F. Pinto, G. Acciarini, S. Metz, S. Boufelja, S. Kaczmarek, K. Merz, J. A. Martinez-Heras, F. Letizia, C. Bridges, and A. G. Baydin. Towards Automated Satellite Conjunction Management with Bayesian Deep Learning. *AI for Earth Sciences Workshop at NeurIPS 2020*, 2020. doi: 10.48550/arxiv.2012.12450.

[26] R. Abay, F. Caldas, M. Filipe, and M. Guimarães. Benchmarking machine learning models for collision risk prediction in low-earth orbit. *Proc. 8th European Conference on Space Debris*, 8, 2021.

[27] J. Subirana, J. Zornoza, and M. Hernández-Pajares. Conventional Celestial Reference System. `https://gssc.esa.int/navipedia/index.php/Conventional_Celestial_Reference_System`, 2011.

[28] R. Bate, D. Mueller, and J. White. *Fundamentals of Astrodynamics*. Dover Publications, first edition, 1971. ISBN 978-0486600611.

[29] D. Vallado. *Fundamentals of Astrodynamics and Applications*. Microcosm Press, fourth edition, 2013. ISBN 978-1881883180.

[30] D. Florijn. Collision Analysis and Mitigation for Distributed Space Systems. Master's thesis, Delft University of Technology, 2015.

[31] E. Kerr and N. Sanchez. State of the Art and Future Needs in Conjunction Analysis Methods, Processes and Software. *8th European Conference on Space Debris*, 8(1), 2021.

[32] E. Stoll, B. D'Souza, B. B. Virgili, K. Merz, and H. Krag. Operational collision avoidance of small satellite missions. In *2013 IEEE Aerospace Conference*, pages 1–11, 2013. doi: 10.1109/AERO.2013.6496955.

[33] 18th Space Control Squadron, Joint Force Space Component Command. Spaceflight Safety Handbook for Satellite Operations: 18 SPCS Processes for On-Orbit Conjunction Assessment & Collision Avoidance, 2020.

[34] H. Klinkrad. *Space Debris: Models and Risk Analysis*. Springer Berlin, Heidelberg, first edition, 2006. ISBN 978-3-540-25448-5. doi: 10.1007/3-540-37674-7.

[35] J. Woodburn, V. Coppola, and F. Stoner. A description of filters for minimizing the time required for orbital conjunction computations. *AAS/AIAA Astrodynamics Specialist Conference, AAS 09-372*, 2009.

[36] F. R. Hoots, L. L. Crawford, and R. L. Roehrich. An analytic method to determine future close approaches between satellites. *Celestial Mechanics*, 33:143–158, 1984. doi: 10.1007/BF01234152.

[37] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006. ISBN 978-0-387-31073-2.

[38] M. Mohri, A. Rostamizadeh, and A. Talwalkar. *Foundations of machine learning*. MIT Press, second edition, 2018. ISBN 0262039400.

[39] A. Müller and S. Guido. *Introduction to Machine Learning with Python: A guide for data scientists*. O'Reilly Media, first edition, 2016. ISBN 978-1449369415.

[40] W. Li and C. Zhang. Markov Chain Analysis. In *International Encyclopedia of Human Geography (Second Edition)*, pages 407–412. Elsevier, Oxford, second edition, 2009. ISBN 978-0-08-102296-2. doi: 10.1016/B978-0-08-102295-5.10403-2.

[41] V. N. Gudivada, D. Rao, and V. V. Raghavan. Chapter 9 - Big Data Driven Natural Language Processing Research and Applications. In *Big Data Analytics*, volume 33 of *Handbook of Statistics*, pages 203–238. Elsevier, 2015. doi: 10.1016/B978-0-444-63492-4.00009-5.

[42] Z. Ghahramani. An Introduction to Hidden Markov Models and Bayesian Networks. *International Journal of Pattern Recognition and Artificial Intelligence*, 15(01):9–42, 2001. doi: 10.1142/S0218001401000836.

[43] L. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989. doi: 10.1109/5.18626.

[44] O. Martin, R. Kumar, and J. Lao. *Bayesian Modeling and Computation in Python*. Chapman and Hall/CRC, Boca Raton, 2021. ISBN 978-0-367-89436-8.

[45] L. Jospin, H. Laga, F. Boussaid, W. Buntine, and M. Bennamoun. Hands-On Bayesian Neural Networks - A Tutorial for Deep Learning Users. *IEEE Computational Intelligence Magazine*, 17(2): 29–48, 2022. doi: 10.1109/MCI.2022.3155327.

[46] S. Theodoridis. Chapter 12 - Bayesian Learning: Inference and the EM Algorithm. In *Machine Learning (Second Edition)*, pages 595–646. Academic Press, second edition, 2020. ISBN 978-0-12-818803-3. doi: 10.1016/B978-0-12-818803-3.00023-4.

[47] A. Gelman, J. Carlin, H. Stern, D. Dunson, A. Vehtari, and D. Rubin. *Bayesian Data Analysis*. Taylor & Francis, third edition, 2013. ISBN 9780429113079. doi: https://doi.org/10.1201/b16018.

[48] J. Salvatier, T. V. Wiecki, and C. Fonnesbeck. Probabilistic programming in Python using PyMC3. *PeerJ Computer Science*, 2:e55, 2016. doi: 10.7717/peerj-cs.55.

[49] H. A. Saadi, F. Ykhlef, and A. Guessoum. MCMC for parameters estimation by Bayesian approach. In *Eighth International Multi-Conference on Systems, Signals & Devices*, pages 1–6, 2011. doi: 10.1109/SSD.2011.5767395.

[50] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of State Calculations by Fast Computing Machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953. doi: 10.1063/1.1699114.

[51] W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970. ISSN 0006-3444. doi: 10.1093/biomet/57.1.97.

[52] S. Duane, A. Kennedy, B. J. Pendleton, and D. Roweth. Hybrid Monte Carlo. *Physics Letters B*, 195(2):216–222, 1987. ISSN 0370-2693. doi: 10.1016/0370-2693(87)91197-X.

[53] M. D. Hoffman and A. Gelman. The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo, 2011.

[54] A. Vehtari, A. Gelman, D. Simpson, B. Carpenter, and P. Bürkner. Rank-Normalization, Folding, and Localization: An Improved $\widehat{R}$ for Assessing Convergence of MCMC (with Discussion). *Bayesian Analysis*, 16(2), 2021. doi: 10.1214/20-ba1221.

[55] Stan Development Team. Stan Reference Manual, 2022. URL `http://mc-stan.org`.

[56] L. Martino, V. Elvira, and F. Louzada. Effective sample size for importance sampling based on discrepancy measures. *Signal Processing*, 131:386–401, 2017. ISSN 0165-1684. doi: 10.1016/j.sigpro.2016.08.025.

[57] A. Jasra, C. C. Holmes, and D. A. Stephens. Markov Chain Monte Carlo Methods and the Label Switching Problem in Bayesian Mixture Modeling. *Statistical Science*, 20(1):50–67, 2005. doi: 10.1214/088342305000000016.

[58] M. Stephens. Dealing with Label Switching in Mixture Models. *Journal of the Royal Statistical Society Series B (Statistical Methodology)*, 62(4):795–809, 2000. doi: 10.1111/1467-9868.00265.

[59] P. Papastamoulis and G. Iliopoulos. An Artificial Allocations Based Solution to the Label Switching Problem in Bayesian Analysis of Mixtures of Distributions. *Journal of Computational and Graphical Statistics*, 19(2):313–331, 2010. doi: 10.1198/jcgs.2010.09008.

[60] C. E. Rodríguez and S. G. Walker. Label Switching in Bayesian Mixture Models: Deterministic Relabeling Strategies. *Journal of Computational and Graphical Statistics*, 23(1):25–45, 2014. doi: 10.1080/10618600.2012.735624.

[61] J. W. Tukey. *Exploratory Data Analysis*. Pearson, first edition, 1977. ISBN 978-0201076165.

[62] N. Andrienko and G. Andrienko. *Exploratory Analysis of Spatial and Temporal Data*. Springer, first edition, 2006. ISBN 978-3-662-49996-2. doi: 10.1007/3-540-31190-4.

[63] European Space Agency. Polar and Sun-synchronous orbit. `https://www.esa.int/ESA_Multimedia/Images/2020/03/Polar_and_Sun-synchronous_orbit`, March 2020.

[64] European Space Agency. Low Earth orbit. `https://www.esa.int/ESA_Multimedia/Images/2020/03/Low_Earth_orbit`, March 2020.

[65] Z. LU and W. HU. Estimation of ballistic coefficients of space debris using the ratios between different objects. *Chinese Journal of Aeronautics*, 30(3):1204–1216, 2017. ISSN 1000-9361. doi: 10.1016/j.cja.2017.03.009.

[66] G. Acciarini, F. Pinto, F. Letizia, J. A. Martinez-heras, K. Merz, C. Bridges, and A. G. Baydin. Kessler: a Machine Learning Library for Spacecraft Collision Avoidance. *8th European Conference on Space Debris*, 8(1), 2021.

[67] B. Frigyik, A. Kapila, and M. Gupta. Introduction to the Dirichlet Distribution and Related Processes. Technical Report UWEETR-2010-0006, University of Washington, 2010.

[68] V. Sundarapandian. *Probability, Statistics and Queuing Theory*. Prentice Hall India Pvt., Limited, 2009. ISBN 978-81-203-3844-9.

[69] SAS Institute Inc. Sas/stat® 15.1 user's guide, 2018.

[70] J. K. Kruschke. Chapter 4 - What is This Stuff Called Probability? In *Doing Bayesian Data Analysis (Second Edition)*, pages 71–97. Academic Press, second edition, 2015. ISBN 978-0-12-405888-0. doi: 10.1016/B978-0-12-405888-0.00004-0.

[71] D. Lewandowski, D. Kurowicka, and H. Joe. Generating random correlation matrices based on vines and extended onion method. *Journal of Multivariate Analysis*, 100(9):1989–2001, 2009. ISSN 0047-259X. doi: 10.1016/j.jmva.2009.04.008.

[72] M. Betancourt. A Conceptual Introduction to Hamiltonian Monte Carlo. *arXiv*, 2017. doi: 10.48550/ARXIV.1701.02434.

[73] R. M. Neal. *Handbook of Markov Chain Monte Carlo*. Chapman and Hall/CRC, first edition, 2011. ISBN 9780429138508. doi: 10.1201/b10905.

Appendix A

Hamiltonian Monte Carlo

This Section explains the HMC algorithm by, first, presenting the formal definitions and equations and, then, providing a physical intuition about how it works.

A Hamiltonian describes the total energy of a physical system and can be decomposed into the sum of the kinetic energy and potential energy, which are completely defined by the momentum and position of an object. In HMC, the problem of sampling from the posterior distribution can be interpreted as recording the position of a particle that is rolling in a valley with a frictionless surface, where it is assumed the total energy of the system is preserved, i.e., if the system loses kinetic energy then it gains the same amount of potential energy. The potential energy of the particle is given by its position θ and the kinetic energy is defined by its momentum $\mathbf{r}$. The Hamiltonian of this system can be written as:

$$H(\theta, \mathbf{r}) = K(\mathbf{r}) + V(\theta), \tag{A.1}$$

in which $K(\mathbf{r})$ and $V(\theta)$ represent the kinetic and potential energy, respectively. However, the main goal of HMC is not to model a particle rolling down an idealized valley, but to simulate a fictitious particle moving along the density regions of the posterior distribution. So, with the addition of an auxiliary momentum variable $\mathbf{r} \in \mathbb{R}^D$ (a momentum for each parameter of the model), the following joint distribution is obtained:

$$p(\mathbf{r}, \theta|\mathbf{X}) = p(\mathbf{r}|\theta, \mathbf{X})\, p(\theta|\mathbf{X}) = p(\mathbf{r})\, p(\theta|\mathbf{X}), \tag{A.2}$$

where the dependencies of the distribution of $\mathbf{r}$ on θ and $\mathbf{X}$ were dropped because, generally, the momentum is an independent random variable. By definition, the joint distribution of the variables θ and $\mathbf{r}$ defines the Hamiltonian [72]:

$$\begin{aligned} H(\theta, \mathbf{r}) &= -\log p(\theta, \mathbf{r}|\mathbf{X}) \\ &= -\log p(\mathbf{r}) - \log p(\theta|\mathbf{X}). \end{aligned} \tag{A.3}$$

Through the comparison of (A.1) and (A.3), it is possible to conclude that the kinetic energy is given as $K(\mathbf{r}) = -\log p(\mathbf{r})$ and the potential energy is given as $V(\theta) = -\log p(\theta|\mathbf{X})$. Note that, by defining

the posterior distribution as the potential energy, one is imposing that the "height" of the particle is described by the density values of the target distribution and, by taking the minus of log-posterior, the density regions are inverted, i.e., the potential energy is the lowest at the highest density values of the posterior distribution and vice-versa, which turns the region to explore into a bowl-like hyper-surface. The distribution $p(\mathbf{r})$ can be freely specified [72] and, in most applications of HMC, the momentum variable $\mathbf{r}$ is drawn independently from a zero mean multivariate normal distribution $p(\mathbf{r}) = \mathcal{N}(0, M)$, in which the covariance matrix M is commonly referred to as the mass matrix. With this definition, the kinetic energy can be written as:

$$K(\mathbf{r}, \theta) = \frac{1}{2}\mathbf{r}^T M^{-1}\mathbf{r} + \log|M| + \text{cte}, \tag{A.4}$$

where $|M|$ is the determinant of matrix M.

Each iteration of the HMC algorithm has three steps. In the first step, the momentum $\mathbf{r}$ is randomly drawn from $p(\mathbf{r})$, which is independent of θ. In the second step, a new sample is proposed by simulating the system through Hamiltonian equations [73]:

$$\frac{d\,\theta}{d\,t} = \frac{\partial H}{\partial \mathbf{r}} = \frac{\partial K}{\partial \mathbf{r}} + \frac{\partial V}{\partial \mathbf{r}} = M\,\mathbf{r} \tag{A.5}$$

$$\frac{d\,\mathbf{r}}{d\,t} = -\frac{\partial H}{\partial \theta} = -\frac{\partial K}{\partial \theta} - \frac{\partial V}{\partial \theta} = -\frac{\partial V}{\partial \theta}, \tag{A.6}$$

where $\frac{\partial K}{\partial \theta} = \frac{\partial V}{\partial \mathbf{r}} = 0$ and $\frac{\partial V}{\partial \theta}$ is the gradient of the logarithm of the target distribution. In this step, the simulated trajectory of the particle has to be computed, i.e., all the points between one sampled value and the next, which leaves a set of differential equations to solve. Most HMC implementations use the leapfrog method that discretizes the Hamiltonian equations into L steps of size/time interval ε. Starting at time $t = 0$ in each HMC iteration, the leapfrog method iteratively computes the position and momentum at times ε, 2ε, 3ε,..., $L\varepsilon$, alternating between half-step updates of the momentum and full-step updates of θ. The leapfrog method proceeds according to the following updates [53, 73]:

$$\mathbf{r}\left(t + \frac{\varepsilon}{2}\right) = \mathbf{r}\,(t) - \frac{\varepsilon}{2}\frac{\partial V}{\partial \theta} \tag{A.7}$$

$$\theta\,(t + \varepsilon) = \theta\,(t) + \varepsilon M \mathbf{r}\left(t + \frac{\varepsilon}{2}\right) \tag{A.8}$$

$$\mathbf{r}\,(t + \varepsilon) = \mathbf{r}\left(t + \frac{\varepsilon}{2}\right) - \frac{\varepsilon}{2}\frac{\partial V}{\partial \theta}, \tag{A.9}$$

where $\mathbf{r}(t)$ and $\theta(t)$ denote the values of $\mathbf{r}$ and θ at time t. The proposed sample at the end of the leapfrog algorithm will be denoted as $\mathbf{r}^*$ and θ^*.

In the final step, the Metropolis acceptance criterion is applied, in which the acceptance probability is given as:

$$\alpha = \min\left[1, \frac{p(\theta^*, \mathbf{r}^*)}{p(\theta, \mathbf{r})}\right] = \min\left[1,\, \exp\Big(-H(\theta^*, \mathbf{r}^*) + H(\theta, \mathbf{r})\Big)\right]. \tag{A.10}$$

If the acceptance probability is lower than a random number drawn from a uniform distribution between 0 and 1, the proposed sample is rejected, otherwise, it is accepted. One might wonder why the Metropolis acceptance criterion is needed, since the energy of the system is preserved, making $H(\theta^*, \mathbf{r}^*) = H(\theta, \mathbf{r})$ and, consequently, $\alpha = 1$. Ideally, if Hamiltonian dynamics were exactly simulated, this would be true,

however, by applying numerical methods in which the Hamiltonian equations are discretized, errors are introduced, leading to a discretized trajectory that may slightly differ from the true trajectory. Thus, to account for these errors that depend on ε, the Metropolis acceptance step is needed.

The procedure for obtaining N posterior samples using the HMC method is represented in Algorithm 1.

Algorithm 1 Hamiltonian Monte Carlo algorithm

Initialize θ_0
for $n = 1$ to N **do** ▷ Sampling iterations of HMC
  $\mathbf{r} \sim \mathcal{N}(0, M)$
  $\theta^* \leftarrow \theta_{n-1}$, $\mathbf{r}^* \leftarrow \mathbf{r}$
  for $l = 1$ to L **do** ▷ Leapfrog algorithm
    $\mathbf{r}^* \leftarrow \mathbf{r}^* - \frac{\varepsilon}{2}\frac{\partial V}{\partial \theta}$
    $\theta^* \leftarrow \theta^* + \varepsilon M \mathbf{r}^*$
    $\mathbf{r}^* \leftarrow \mathbf{r}^* - \frac{\varepsilon}{2}\frac{\partial V}{\partial \theta}$
  end for
  $\alpha \leftarrow \min\left[1, \exp\{-H(\theta^*, \mathbf{r}^*) + H(\theta_{n-1}, \mathbf{r})\}\right]$ ▷ MH acceptance criterion
  $u \sim \mathcal{U}(0, 1)$
  if $\alpha > u$ **then**
    $\theta_n \leftarrow \theta^*$
  else
    $\theta_n \leftarrow \theta_{n-1}$
  end if
end for

To get a better understanding and intuition about this algorithm, Neal [73] visualizes the exploration process of the posterior distribution as a puck that slides over a frictionless ice surface of varying height, in which the position of the puck is given by θ and its momentum given by $\mathbf{r}$. The puck moves when it is pushed towards an arbitrary direction and, after sliding for a defined time of T, it stops and its position is noted. This process is repeated N times. The pushes that set the puck towards an initial direction, at every iteration, can be seen as the momentum variable that is randomly drawn from a Multivariate Gaussian distribution. Since the surface is frictionless, the puck will move at constant velocity (with constant momentum) in a flat region. In those cases of a flat density region, the analysis of the leapfrog algorithm shows that $\mathbf{r}$ remains constant, because $\frac{\partial V}{\partial \theta} = 0$ (there is no change in the potential energy), which physically makes sense. If the puck encounters a rising slope on the surface, i.e., $\frac{\partial V}{\partial \theta} > 0$, the puck's momentum allows it to climb, increasing the potential energy, but, at the same time, reducing the kinetic energy, in other words, decreasing its velocity/momentum. The puck will climb until the kinetic energy is zero, at which point it starts to slide down. Through the analysis of the leapfrog algorithm equations (A.7) and (A.9), it can be seen that the momentum decreases as the potential energy increases until $\mathbf{r}$ reaches zero. Then, $\mathbf{r}$ changes the sign (the puck movement changes direction), affecting (A.8), causing a reduction in θ — the puck starts to slide down.

Appendix B

Inferences

In this chapter, a descriptive analysis of the inferences for the parameters of the mean and the standard deviations of the emission distributions of the final HMMs are presented.

B.1 HMM of Approach A

This Section presents the inferred parameters for the HMM of **Approach A**, using both the CAC and Neuraspace datasets.

B.1.1 CAC dataset

Parameter	Mean	Standard deviation	HDI 3%	HDI 97%
μ_1	-11.006	0.019	-11.041	-10.972
μ_2	-14.352	0.037	-14.423	-14.285
μ_3	-4.963	0.011	-4.983	-4.941
μ_4	-30.000	0.000	-30.000	-30.000
μ_5	-22.426	0.075	-22.574	-22.291
μ_6	-8.871	0.015	-8.897	-8.841
μ_7	-6.343	0.008	-6.358	-6.328
μ_8	-7.406	0.010	-7.424	-7.387
σ_1	0.803	0.011	0.783	0.823
σ_2	1.478	0.027	1.431	1.532
σ_3	0.679	0.007	0.667	0.692
σ_4	0.011	0.000	0.011	0.011
σ_5	3.776	0.044	3.694	3.857
σ_6	0.588	0.008	0.572	0.602
σ_7	0.337	0.005	0.328	0.346
σ_8	0.364	0.006	0.353	0.374

Table B.1: Descripitive analysis of the inferences for the parameters of the mean and the standard deviations of the emission distributions, for the HMM of Approach A, using the CAC dataset.

The column *HDI 3%* represents the value below which 3% of the sampled posterior falls and the *HDI 97%* column reports the values above which 3% of the posterior falls.

B.1.2 Neuraspace dataset

Parameter	Mean	Standard deviation	HDI 3%	HDI 97%
μ_1	-12.719	0.031	-12.777	-12.662
μ_2	-30.000	0.000	-30.000	-29.999
μ_3	-10.451	0.021	-10.491	-10.411
μ_4	-6.778	0.017	-6.814	-6.748
μ_5	-26.530	0.055	-26.631	-26.423
μ_6	-20.590	0.059	-20.702	-20.483
μ_7	-15.938	0.056	-16.043	-15.831
μ_8	-8.342	0.021	-8.384	-8.304
μ_9	-5.092	0.019	-5.126	-5.053
σ_1	0.781	0.022	0.740	0.821
σ_2	0.027	0.000	0.026	0.028
σ_3	0.629	0.016	0.601	0.660
σ_4	0.453	0.011	0.433	0.473
σ_5	1.811	0.037	1.738	1.877
σ_6	1.703	0.039	1.630	1.775
σ_7	1.241	0.035	1.178	1.310
σ_8	0.590	0.014	0.564	0.618
σ_9	0.714	0.011	0.693	0.735

Table B.2: Descripitive analysis of the inferences for the parameters of the mean and the standard deviations of the emission distributions, for the HMM of Approach A, using the Neuraspace dataset.

As in Section B.1, the column *HDI 3%* represents the value below which 3% of the sampled posterior falls and the *HDI 97%* column reports the values above which 3% of the posterior falls.

B.2 HMMs of Approach B

Tables B.3, B.4 and B.5 present an analysis of the inferred parameters of the mean and the standard deviations of the emission distributions of the final HMMs for the σ_T^{chaser}, σ_N^{chaser} and σ_R^{chaser} evolution, respectively, where the column *Std* represents the standard deviation of the sampled values.

Parameter	Mean	Std
μ_1	489.467	8.394
μ_2	1865.318	18.144
μ_3	60071.534	458.006
μ_4	7846.198	53.300
μ_5	26546.870	170.780
μ_6	11497.633	3670.471
μ_7	5993.068	3669.840
σ_1	288.916	6.072
σ_2	549.451	10.198
σ_3	16406.806	305.190
σ_4	1437.904	26.045
σ_5	5641.217	113.456
σ_6	1848.929	529.276
σ_7	1056.932	527.970

Table B.3: Descriptive analysis of the inferences for the HMM for σ_T^{chaser}.

Parameter	Mean	Std
μ_1	80.816	0.254
μ_2	30.434	0.141
μ_3	111.894	0.556
μ_4	299.641	1.764
μ_5	165.097	1.164
μ_6	52.169	0.185
μ_7	13.508	0.109
σ_1	8.830	0.173
σ_2	5.337	0.084
σ_3	10.663	0.383
σ_4	64.577	1.171
σ_5	23.681	0.554
σ_6	7.741	0.123
σ_7	5.641	0.069

Table B.4: Descriptive analysis of the inferences for the HMM for σ_N^{chaser}.

Parameter	Mean	Std
μ_1	51.203	0.224
μ_2	171.369	1.119
μ_3	86.687	0.478
μ_4	456.960	3.777
μ_5	23.424	0.188
σ_1	9.198	0.144
σ_2	40.360	0.630
σ_3	13.152	0.373
σ_4	187.783	2.327
σ_5	8.723	0.117

Table B.5: Descriptive analysis of the inferences for the HMM for σ_R^{chaser}.