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

1st João Simões Catulo
Instituto Superior Técnico
Universidade de Lisboa, Portugal
joao.catulo@tecnico.ulisboa.pt

***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.**

***Index Terms*—hidden Markov models, Bayesian inference, collision risk estimation, machine learning**

I. Introduction

Since the beginning of the space age, with the launch of Sputnik-1 in 1957, the amount of resident space objects in Earth orbit has been steadily increasing, as shown in Figure 1, which presents the evolution of the number of objects in space from 1957 until today. The space environment is becoming progressively crowded and space traffic is undergoing notable changes fuelled by the development of commercial and private space activities and the deployment of large constellations, especially in the Low Earth Orbit (LEO) region, which further adds to the growth of space population. This continuous growth of the number of objects in space can pose a great danger to all operational satellites since collisions between resident space objects create large amounts of fragments that are further released into orbit. These fragmentation events create numerous debris that are spread into different directions at different velocities and, over time, lead to a gradual pollution

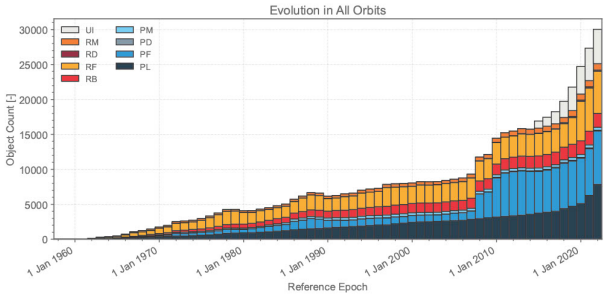

Fig. 1. Count evolution of objects in space from 1957 to 2022 [1].

of a vast volume of space [2], eventually contaminating entire orbital altitudes. If measures are not taken, collisions between space objects can reach a cascading point, in which collisions may cause a sequence of new impacts, due to the high density of objects in orbit, posing a real threat to space missions and endangering the whole space population. This effect is known as the *Kessler Syndrome* [3]. For these reasons, the need to consider collision avoidance as part of routine operations is evident.

When an event between two space objects meets the conditions for a close conjunction, in which the monitored space object is referred to as target and the other object as chaser, collision warnings in the form of conjunction data messages (CDMs) are created and sent to the operators of the satellites. These messages contain propagated information about the event to the time of closest approach (TCA). However, orbit determination and propagation cannot be modeled with desired precision and have associated uncertainties making it impossible to know for sure whether a collision will occur or not. Hence, during the time span of the conjunction event, both objects that generated the issue of warning messages are routinely tracked, leading to the creation of more CDMs that contain refined and more precise information about the conjunction. Typically, a LEO satellite receives hundreds of CDMs per week that, currently, require the analysis of human experts/analysts, generating high operational costs [4]. With the continuous growth of the space population, this approach may be an unfeasible task in the future, highlighting the

importance of automation in risk assessment and estimation.

In 2019, ESA launched the *Collision Avoidance Challenge* (CAC) [5] to study the feasibility of applying machine learning (ML) methods in collision risk estimation and released a dataset that contained sequences of CDMs received in support of real close encounters. The competition aimed to develop ML models capable of predicting the criticality of conjunction events by analyzing the time series of CDMs received up to 2 days before the predicted TCA, which is considered the cut-off time. The collision probability within the CDMs is computed through the Alfriend-Akella algorithm [6] and the final risk of each event is considered to be the risk contained in the last released CDM, which is considered the best knowledge about the outcome of the close approach. Figure 2 illustrates the concept of ML in collision avoidance.

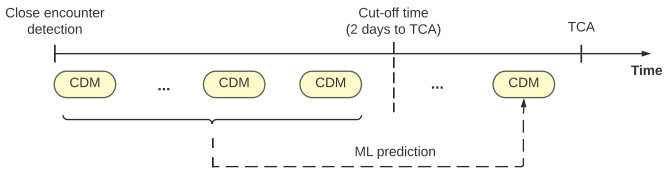

Fig. 2. Concept of the ML approach in collision avoidance.

The competition showed that the naive/baseline approach (using the risk contained in the last CDM received until the cut-off time as the risk prediction) is a strong predictor for this problem, with only 12 teams out of 97 managing to beat the benchmark solution [7]. 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. Manhattan LSTMs [8] and Gradient boosting trees showed good performance during the CAC.

After the competition, relevant work regarding the use of ML in collision avoidance has been conducted. Metz [9] 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 [6]. Acciarini et al. [10] built a physics-based generative model using probabilistic programming to simulate the generation of CDMs, based on real data. Pinto et al. [11] used Bayesian deep learning with recurrent neural network architectures to also study the possibility of generating CDMs. Abay et al. [12] benchmarked the results for the state-of-the-art ML 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 [7], 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 by implementing and benchmarking the use of hidden Markov model (HMM) in the risk prediction problem, using Bayesian statistics. For that, two datasets are used: the one that ESA released for the CAC challenge and a real-world raw dataset provided by *Neuraspace*.

II. Background

In this Section, the necessary theoretical concepts about Bayesian modeling and HMMs are provided. This section has been kept as brief as possible while giving all the necessary concepts. For more details, the reader is encouraged to read the references provided throughout this Section.

A. Bayesian Modeling

In probabilistic models, the set of parameters θ of a probabilistic model is typically obtained by finding the parameters that result in the best match between the model and the observed data $\mathbf{X}$, using e.g. the maximum likelihood estimation. In this work, rather than estimating a single set of parameters, an entire joint distribution for θ is inferred. This is possible by adopting a Bayesian approach, in which the unknown parameters are treated as random variables and probability theory is used to update its values conditioned on the observed data [13]. 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 some observed data $\mathbf{X}$.

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

$$p(\theta|\mathbf{X}) = p(\mathbf{X}|\theta)\, p(\theta)/p(\mathbf{X}), \tag{1}$$

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. Once the posterior is defined, it can be used to obtain predictions of the model for new input data.

However, computing the distribution $p(\theta|\mathbf{X})$ analytically is usually an unfeasible problem since it depends on the computation of the normalizing constant $p(\mathbf{X})$:

$$p(\mathbf{X}) = \int_\theta p(\mathbf{X}, \theta)\, d\theta = \int_\theta p(\mathbf{X}|\theta) p(\theta)\, d\theta, \tag{2}$$

where it is necessary to integrate over all the possible values of θ. To address this issue, Markov chain Monte Carlo (MCMC) methods are used. These methods approximate the posterior distribution using samples, by evaluating the likelihood and prior distributions at different parameter values. In this work, Bayesian statistical models are implemented using a probabilistic programming framework called *PyMC* [14] and, to sample from the posterior, the No-U-Turn Sampler (NUTS) [15] is used.

B. Hidden Markov Models

HMM is a type of directed graphical model and a tool for representing probability distributions over sequences of observations that are produced by an underlying stochastic process, whose states cannot be directly observed, i.e., are hidden [16]. This hidden process that generates the observations is a first-order finite state Markov chain and, hence, respects the Markov property that 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" [17]. Throughout this work, the number of possible states that each latent variable can take will be denoted as K, the sequences of hidden states as $\mathbf{Z} = \{\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_N\}$ and the sequences of observations as $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$, where each hidden state $\mathbf{z}_n$ generates the corresponding observation $\mathbf{x}_n$ (which may be of different type or dimension [18]) and N represents the number of observations.

A HMM with N observations is depicted as a graphical model in figure 3.

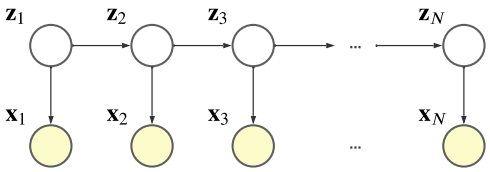

Fig. 3. Graphical representation of a HMM for a sequence of N observations.

In HMMs, the latent variables $\mathbf{Z}$ follow a Markov chain with transition matrix $\mathbf{A} \in \mathbb{R}^{K \times K} : \mathbf{A} \geq 0$, $\mathbf{A}\mathbb{1} = \mathbb{1}$ (where $\mathbb{1}$ denotes a K-dimensional vector with elements equal to 1) and initial distribution $\boldsymbol{\pi} \in \mathbb{R}^K : \boldsymbol{\pi} \geq 0$, $\boldsymbol{\pi}^T \mathbb{1} = 1$ that represent the probability of transitioning from one hidden state to another — $p(\mathbf{z}_n|\mathbf{z}_{n-1}, \mathbf{A})$ — and the hidden state initialization probability — $p(\mathbf{z}_1|\boldsymbol{\pi})$ —, respectively. The i-th row of $\mathbf{A}$, which 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 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 $\mathbf{A}_{ij}$ represents the probability of transitioning from state i to state j. The observations, that depend on the hidden states, are specified by the emission distributions $p(\mathbf{x}_n|\mathbf{z}_n, \boldsymbol{\phi})$, where $\boldsymbol{\phi}$ is the set of parameters that rule the distribution, that can be either discrete or continuous. Thus, a HMM is then completely specified by the set of components $\theta = (\mathbf{A}, \boldsymbol{\pi}, \boldsymbol{\phi})$ that must be learnt during the training phase.

As previously described, in this work, a Bayesian approach is adopted, in which current statistical procedures depend on the likelihood distribution of the models. Hence, the likelihood distribution $p(\mathbf{X}|\theta)$ of HMMs is needed.

1) Likelihood distribution: 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 [18]:

$$p(\mathbf{X}|\theta) = \sum_{\mathbf{z}_N} \alpha(\mathbf{z}_N), \tag{3}$$

where $\alpha(\mathbf{z}_n) = p(\mathbf{x}_n|\mathbf{z}_n, \boldsymbol{\phi}) \sum_{\mathbf{z}_{n-1}} \alpha(\mathbf{z}_{n-1}) p(\mathbf{z}_n|\mathbf{z}_{n-1}, \boldsymbol{A})$ and $\alpha(\mathbf{z}_1) = p(\mathbf{z}_1|\boldsymbol{\pi}) p(\mathbf{x}_1|\mathbf{z}_1, \boldsymbol{\phi})$.

2) Predictive distribution: In this work, another quantity of interest is the predictive distribution $p(\mathbf{x}_{N+1}|\mathbf{X}, \theta)$, 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 observation $\mathbf{x}_{N+1}$. This distribution is given by [18]:

$$\begin{aligned} p(\mathbf{x}_{N+1}|\mathbf{X}, \theta) = & \frac{1}{p(\mathbf{X}|\theta)} \sum_{\mathbf{z}_{N+1}} p(\mathbf{x}_{N+1}|\mathbf{z}_{N+1}, \theta) \\ & \cdot \sum_{\mathbf{z}_N} p(\mathbf{z}_{N+1}|\mathbf{z}_N, \theta) \alpha(\mathbf{z}_N). \end{aligned} \tag{4}$$

However, during a conjunction event, 3 CDMs are received, on average, per day [7], and since the defined cut-off time is 2 days before the TCA, it could be advantageous to predict the information 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, and only $\mathbf{x}_{N+1}$ is predicted and is used to benchmark the performance of HMMs.

III. Data Analysis and Cleaning

In this work, two datasets are used: the one released by ESA during the CAC and one provided by Neuraspace, which contains real-world raw data regarding collision avoidance operations. The CAC dataset contains 162 634 samples/CDMs (with 103 parameters each) and 13 154 unique conjunction events. In the Neuraspace dataset there are 55 161 events and 1 586 152 CDMs (each containing 231 features). In both datasets, 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. Note that the values of all parameters contained in each CDM are propagated to the TCA.

However, not all events contained in the datasets are eligible for the ML approach, since spacecraft operators need time to make a decision regarding the performance of an avoidance maneuver. Thus, the events must follow some constraints [7]: (i) the events must have at least 2 CDMs, one to learn and one to use as label; (ii) the first CDM has to be released before the cut-off time (2 days until TCA); (iii) the last CDM has to be released within 1 day of the TCA.

Since the goal of ML models in the collision risk assessment is to analyze the sequence of the values of the collision risk contained in the CDMs received until the cut-off time and correctly identify whether an event is of high or low risk of collision, the data can be divided into two categorical classes, based on the risk that is present in the last CDM released in each event: if the risk is lower than −6, the event is considered of low risk, otherwise it is considered a high-risk event. The proportion of these classes is 97.23% of low-risk and 2.77% of high-risk, for the CAC dataset, and, for the Neuraspace dataset, the proportion is 99.30% of low-risk and only 0.74% of high-risk. The different proportions between the two datasets are due to the fact that CAC data was slightly manipulated by ESA before being released to the public in order to contain a slightly higher percentage of CDMs containing high-risk values, to be better explored during the competition [7]. The data imbalance problem poses to be the main challenge in collision risk estimation.

By performing an exploratory data analysis (EDA) on the datasets, some anomalies can be found. There are parameters that contain extreme outliers or even physically impossible values — for example, negative ballistic coefficients or energy dissipation rates. In addition, in some collision warnings, the position standard deviations (along-track, radial, and transverse) of the target and chaser take values larger than the Earth radius, which is unrealistic from a physical point of view and affects the value of the collision probability.

During the data preparation and cleaning phase, in this work, both datasets are kept as close to the original as possible, in order to benchmark the performance of HMMs with data that is representative of real collision avoidance missions. Hence, only the CDMs that contain unrealistic or physically impossible values (like negative ballistic coefficients or position errors larger than the radius of the Earth) are removed. In addition, the events that don't follow the previously described constraints are also discarded. In the Neuraspace dataset, two additional cleaning steps must be taken. The Neuraspace data contains events in which a collision avoidance maneuver was performed. An avoidance maneuver is an external factor that influences and alters the natural evolution of the values of the parameters, which can affect the training of the ML models. Thus, the maneuvered conjunctions are not considered in this work. In addition, the collision warnings within the Neuraspace dataset have different originators, but only the ones released by the 18th Space Control Squadron [19] are kept, because they represent the rawest version of the data.

IV. Methodology

In collision avoidance, it is extremely important to identify high-risk conjunctions in order to prevent catastrophic collisions between space objects that can damage and lead to the destruction of operational spacecraft. However, as presented in Section III, both datasets are extremely imbalanced with only a very small percentage of the events having a final risk higher than −6. In addition, the CAC results [7] have shown that the risk can have an unpredictable and complex evolution within the events, highlighting the difficulty of using the risk of collision as the prediction target. These problems encouraged Metz [9] to explore another approach: using the chaser position uncertainties as the prediction label and then recomputing the probability of collision with the predictions, using Akella's and Alfriend's equation [6]. 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 of the target and chaser, and can be visualized as an error ellipsoid centered on each object. The work done by Metz [9] showed that this method yields good results 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.

In real-life operations, there is a lot more available information regarding the target object (operating spacecraft), which leads to more accurate orbit determinations and propagated position estimates with lower uncertainties, meaning that it can be assumed that the last CDM released before the cut-off time offers a good estimation of the orbital state parameters of the target, at the TCA. The orbital states at the TCA of the chaser object (which can be of multiple types, sizes, and origins) are much more difficult to estimate, and, hence, it can be advantageous to predict them, using ML methods. Just like in Metz's work, only the three chaser standard deviations σ_R^{chaser}, σ_T^{chaser}, and σ_N^{chaser} that represent the position uncertainties in the radial, transverse and normal components of the chaser RTN reference frame are used as the prediction targets and the rest of the necessary parameters to compute the collision probability using Akella's and Alfriend's equation (relative position vector, the other components to compute the combined covariance matrix and the object sizes) are taken from the last released CDM before the cut-off time. The three components of the position errors of the chaser define the diagonal entries of the chaser covariance matrix and express the "length" of each component of the error ellipsoid, so these parameters are considered the main contributors to the collision probability.

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, taking the other necessary parameters from the last released CDM, before the cut-off time.

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

However, it is important to note that the CAC dataset cannot be used in **Approach B**. 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. To combine the covariance matrices it is necessary to transform them into the same reference frame, but, for that, the position and velocity vectors of the objects are needed, which are parameters that are not available in the CAC dataset.

When the naive forecast predicts the final risk value of an event as −30 (the risk values are truncated at that lower bound), 99.34% of the predictions are correct, in the CAC dataset, and, in the Neuraspace dataset, 99.93% of the predictions are accurate. 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. So, those events are not used for training and, during the test phase, are directly predicted with a final risk of −30. With this approach, a significant amount of low-risk conjunctions are removed, which can help deal with the data imbalance problem, and the volume of training data is reduced, resulting in lower memory requirements and a lower computational time. However, it is important to note

that this method is not perfect, because, in the Neuraspace dataset, two high-risk conjunctions are predicted as −30 by the naive forecast, suggesting that future work should focus on the development of a better and more robust model that can detect the −30 events without any miss classifications of high-risk conjunctions.

Furthermore, since the goal of this work is to benchmark the performance of Bayesian HMMs in risk estimation, each HMM will only learn the evolution of one single feature of the dataset. In other words, 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. This way, this work provides a foundation for future research regarding the implementation of HMMs, with Bayesian statistics, in collision risk estimation.

V. Data Preparation and Setup

Taking into consideration the methodology followed in this work, it is necessary to prepare the data in order to be analyzed by the models. After the data cleaning described in Section III, the CDMs of each event are arranged in descending order regarding the time to the predicted TCA, and the parameters of each collision warning that are not necessary for the risk prediction process of each approach are removed. 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 are considered. Then, a stratified split is performed in both datasets to preserve the same proportion of samples of each class, and the data is divided 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 of the TCA. The training set is used to infer the parameters of the HMMs and, as previously described, only the events that have a baseline risk different from −30 are used for training.

However, at this point, a challenge arises. To infer the parameters of each model, 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 equation (3) 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 is to vectorize the sequences of CDMs and compute the log-likelihood density for each sequence at once and then sum the result. To vectorize the sequences of collision warnings, an approximation must be done regarding the data setup of the training set. Typically, in real collision events, 3 CDMs are released per day [7] 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 less than 3 collision warnings are received, the latest CDM received is repeated until there are 3 on that day. If there are no CDMs received prior 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 4.

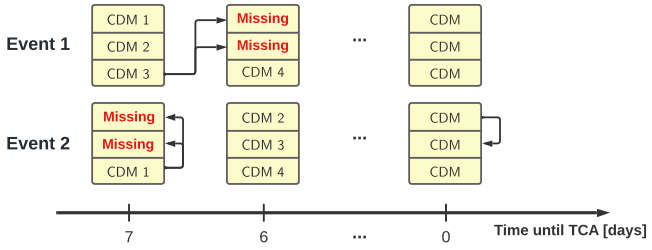

Fig. 4. Data Setup schematization.

VI. Bayesian Models

After the data preparation, the training sequences of observations are used to infer the parameters of the models. In this work, it is believed that the risk/position errors generated by each latent variable of the HMM should be near a specific value and the occurrence of risk/position errors far from that value is less frequent. However, it is important to note that the variables to model must follow some constraints: the risk is truncated at a lower bound of −30 and cannot be greater than 0, because the risk is defined as the $\log_{10}$ of the collision probability; and the positional standard deviations are restricted to be greater than zero because they define the diagonal entries of the covariance matrix, which has to be positive semi-definite. To take these constraints into account, univariate Truncated Normal distributions are used as the emission distributions of the HMMs of both **Approach A** and **Approach B**, with lower and upper bounds of −30 and 0 for the HMM of the risk evolution, and lower bound of 0.01 m for the HMMs of the standard deviation evolution (where it is to assumed that the position uncertainties caused by measurement and/or propagation errors have a minimum value of 1 cm). Therefore, the parameters that must be inferred for each implemented HMM are the following, where K represents the number of possible hidden states: (i) the transition probabilities represented by the matrix $\mathbf{A} \in \mathbb{R}^{K\times K} : \mathbf{A} \geq 0, \mathbf{A}\mathbb{1} = \mathbb{1}$; (ii) the initial probability distribution represented by the vector $\boldsymbol{\pi} \in \mathbb{R}^K : \boldsymbol{\pi} \geq 0, \boldsymbol{\pi}^T\mathbb{1} = 1$; (iii) the mean values $\boldsymbol{\mu} \in \mathbb{R}^K$

of the emission distributions; (iv) the standard deviations $\boldsymbol{\sigma} \in \mathbb{R}_+^K$ of the emission distributions. The parameters $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ denote the set of mean values and standard deviations of the emissions, respectively, as $\boldsymbol{\mu} = [\,\mu_1, \mu_2, ..., \mu_K\,]$ and $\boldsymbol{\sigma} = [\,\sigma_1, \sigma_2, ..., \sigma_K\,]$, in which μ_k and σ_k represent the mean and standard deviations of the Truncated Normal emission generated by the hidden state k. To find the best value for K, in each approach, a stratified cross-validation with 5 folds is performed.

To perform Bayesian inference on the HMMs, it is essential to define the prior distributions for each of the parameters $\theta = (\mathbf{A}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma})$ of the models of **Approach A** and **Approach B**.

Notice that the parameters of the priors presented in this section were chosen via a trial-and-error process (taking into consideration the constraints of the variables), so they are not unique and can be further improved.

A. Priors for HMM of Approach A

Regarding the parameters $\boldsymbol{\pi}$ and $\mathbf{A}_i \in \mathbb{R}^K : \mathbf{A}_i \geq 0$, $\mathbf{A}_i^T \mathbb{1} = 1$, that define the initial state distribution and the rows of the transition matrix (with $i \in \{1, ..., K\}$) a natural choice of priors is the Dirichlet distribution, that is is confined to a simplex, i.e., all elements of the random variable belong to the interval $[0, 1]$ and sum up to one. The Dirichlet distribution is parameterized by the vector $\boldsymbol{\alpha} \in \mathbb{R}^K : \boldsymbol{\alpha} > 0$ whose elements must be positive real numbers and, in the case where all elements of $\boldsymbol{\alpha}$ are equal to one, the 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 nor about the hidden state transitions, so a Dirichlet distribution with elements of $\boldsymbol{\alpha}$ equal to one is used as prior for $\boldsymbol{\pi}$ and $\mathbf{A}_i$.

Since the risk can only take values between −30 and 0, the mean values of the emission distributions are also 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 $K = 3$, the priors for the elements of $\boldsymbol{\mu}$ will be: $\mu_1 \sim \mathcal{TN}(\mu = -30, \sigma = 4)$, $\mu_2 \sim \mathcal{TN}(\mu = -15, \sigma = 4)$ and $\mu_3 \sim \mathcal{TN}(\mu = 0, \sigma = 4)$, in which the values of the lower and upper bounds of the Truncated Normal distribution are not shown, because these are fixed throughout.

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. The chosen distribution for the priors of the elements of $\boldsymbol{\sigma}$ is the inverse gamma distribution with parameters α and β equal to 40 and 80, respectively (these values were chosen through a trial and error process).

In summary, the priors for the HMM of **Approach A** are given by:

$$
\begin{aligned}
\boldsymbol{\pi} &\sim \text{Dir}(\boldsymbol{\alpha} = \mathbb{1}); \\
\mathbf{A}_i &\sim \text{Dir}(\boldsymbol{\alpha} = \mathbb{1}), \quad \forall\, i \in \{1, ..., K\}; \\
\boldsymbol{\mu} &\sim \mathcal{TN}\big(\,\mu = m,\, \sigma = 4,\, \text{L} = -30,\, \text{U} = 0\,\big); \\
\sigma_i &\sim \mathcal{IG}(\alpha = 40, \beta = 80), \quad \forall\, i \in \{1, ..., K\},
\end{aligned}
$$

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, L and U denote the lower and upper bounds of the distributions, respectively, and $\mathbb{1}$ denotes a K-dimensional vector with all the elements equal to one.

B. Priors for HMMs of Approach B

The priors for the parameters of the three HMMs of **Approach B** are very similar to the ones presented in Section VI-A, only changing the parameters of some prior distributions.

As in the HMM of **Approach A**, the chosen priors for $\boldsymbol{\pi}$ and the rows of $\boldsymbol{A}$ of the three HMMs of **Approach B** are Dirichlet distributions with elements of $\boldsymbol{\alpha}$ equal to one, since there is neither any prior knowledge regarding the first state that generates the first observation of any of the three position uncertainties nor any belief about the probability of transitioning from one hidden state to another.

For $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ of the emissions of each HMM, the chosen priors are Truncated Normal distributions (with lower bound of 0.01 m) and Inverse Gamma distributions, respectively. The CDMs containing values above 100 000 m, 1 000 m and 450 m for σ_T^{chaser}, σ_R^{chaser}, and σ_N^{chaser} are considered errors [20]. Hence, the mean values of the prior distributions for the elements of $\boldsymbol{\mu}$ (Truncated Normals) of the three HMMs are equally spaced within the range from 0.01 m to the respective upper threshold of the variable to model. The standard deviations of those priors are defined as 10 000 m, 400 m and 100 m for the HMMs of σ_T^{chaser}, σ_R^{chaser}, and σ_N^{chaser}, respectively. The parameters α and β of the Inverse Gamma priors for $\boldsymbol{\sigma}$ are defined such that the density areas of the distribution can cover a large number of possible values. For the HMM of σ_T^{chaser} the parameters of the Inverse Gamma are $\alpha = 4.5$ and $\beta = 28\,480$, for σ_R^{chaser} are $\alpha = 7$ and $\beta = 545$, and for σ_N^{chaser} are $\alpha = 6$ and $\beta = 500$.

C. Inferences

With the likelihood and prior distributions, it is possible to infer the parameters of the implemented models, using the NUTS. As previously described, a stratified cross-validation with five folds is performed in order to find the best value for K of each HMM and the best models for **Approach A** and **Approach B** are then trained using the entire training set. During cross-validation, 3 chains of 2 000 iterations are sampled 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 set to 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.

In this work, after sampling, it is necessary to deal with the *label switching problem* [21] — the label of the parameters switch between or within chains, due to the invariance of the likelihood and priors in the permutations of θ. In this work, this is solved by relabelling the chains according to statistical analysis. Then, the convergence and autocorrelation of the sampled chains of each model are checked by visualizing the trace plots and by analyzing some of the convergence diagnostics criteria provided by *PyMC*, such as the Potential Scale Reduction ($\hat{R}$) [22] and the Effective Sample Size (ESS) [23]. If the inferences pass all the requirements, the samples of the posterior distribution can be used to obtain predictions.

VII. Results

This section presents the results obtained with **Approach A** and **Approach B**, comparing the predictions with the naive solution. In this section, only part of the results of the complete work are presented. For more details, the reader is encouraged to read the full document of the thesis.

To obtain predictions, in all HMMs, 400 draws are randomly taken from each sampled chain of the posterior distribution and are given as input to equation 4 (in this work, the performance of HMMs is benchmarked by predicting only $\mathbf{x}_{N+1}$), outputting a distribution that reflects the prediction uncertainty. The final predicted value of the desired parameter of each event is the mean of the corresponding distribution. The metrics used to evaluate the models are the root mean squared error (RMSE), the mean absolute error (MAE), precision, recall, and F_1 and F_2 scores. In addition, confusion matrices are also used.

A. Results of Approach A

In this approach, two datasets are used: the CAC and the Neuraspace data. Thus, the results are divided into two parts.

1) CAC dataset: To choose the best number of possible hidden states (K), cross-validation is performed and K is iterated between 4 and 10 states. For a lower number of K, it is considered that the HMMs have poor coverage of all the possible values of the desired parameter, and, for a higher number of K, the chains start converging into different values, indicating that the posterior distribution is multimodal with sharp density regions, and the NUTS cannot explore the density areas of the desired distribution. Future work may tackle this issue by using/developing an efficient sampler that can handle multimodality, but, in this work, this step is simplified. 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 cross-validation, the performance of the best model of **Approach A** (in this case, a HMM with 8 states is chosen) is then tested using the test set (recall that the events with a naive forecast of −30 are directly predicted as having a final risk of −30). Table I shows the performance metrics for both the complete model of **Approach A** and the baseline predictions.

Metrics						
Model	RMSE	MAE	Precision	Recall	F_1	F_2
Approach A	**8.40**	**4.32**	**17.5%**	**70.0%**	**0.280**	**0.438**
Baseline	8.88	4.43	14.6%	**70.0%**	0.241	0.398
Confusion Matrix (Model | Baseline)						
	Pred. Low-Risk		Pred. High-Risk			
True Low-Risk	1406 | 1398		33 | 41			
True High-Risk	3 | 3		7 | 7			

TABLE I: Performance metrics and confusion matrix for the model of **Approach A** and baseline solution, using the CAC dataset.

The implemented model outperforms the baseline solution in all metrics (with the exception of the recall) and significantly reduces the number of miss-classified low-risk events. The results show that, despite the approximations made to build the model, the complex behavior of the risk updates within the events, the data imbalance problem, and the fact that only one feature is used, the implemented model manages to outperform the naive forecast, which is considered a very strong predictor for this problem. The miss-classified low-risk events are the same for both models, but, to analyze the source of these errors, more data would be needed, since that three events are not a sufficiently large sample to take conclusions from.

Figure 5 shows the predicted values of the risk against the true risk values. The predictions tend to be arranged in 8

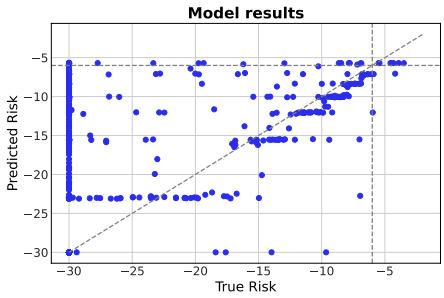

Fig. 5. Predicted vs True risk values, using **Approach A** and the CAC data.

steps, which correspond to the sampled mean of the emission distributions of the HMM. It can also be seen a large over-prediction of the −30 events. All of these over-predicted events share the same behavior: the risk updates evolve at high-risk values, but, after the cut-off time, there is a big risk transition, from high-risk values to −30.

As previously mentioned, an entire distribution is obtained for each prediction, so prediction intervals can be provided for each event. Figure 6 shows the 95% Highest Density

Interval (HDI) associated with each prediction for the true high-risk events of the test set. It can be seen that the prediction intervals have poor coverage of the true high-risk values and that the predictions seem to be "truncated" at an upper bound. It is important to highlight that this is a simple model that only analyzes one feature of the dataset - the risk, which is extremely imbalanced. To improve the results, it could be beneficial to train the model with a larger dataset containing more high-risk conjunctions and explore the impact of other features.

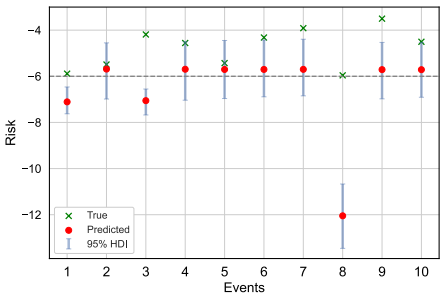

Fig. 6. Representation of the true risk values of each event (green), the HMM predictions (red) and the 95% HDI area (area shaded in blue), for all the high-risk events of the CAC dataset, following **Approach A**.

2) Neuraspace dataset: For the Neuraspace dataset, the number of possible hidden states is iterated between 4 and 9 states in each cross-validation fold. A higher number of states 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 data was prepared and slightly manipulated by ESA in order to be publicly released, 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. After cross-validation, the HMM with K equal to 9 is chosen, because it has better scores in all metrics. This result suggests that future work should tackle the problem of sampling from multimodal posterior distributions, in order to test the performance of HMMs with a higher number of possible hidden states. The performance metrics of the complete model with the final HMM are presented in Table II.

Metrics						
Model	RMSE	MAE	Precision	Recall	F_1	F_2
Approach A	**2.54**	**0.40**	**31.5%**	**62.5%**	**0.417**	**0.521**
Baseline	2.66	0.41	29.4%	**62.5%**	0.400	0.510
Confusion Matrix (Model | Baseline)						
	Pred. Low-Risk		Pred. High-Risk			
True Low-Risk	6851 | 6850		11 | 12			
True High-Risk	3 | 3		5 | 5			

TABLE II: Performance metrics and confusion matrix for the models, using the Neuraspace dataset.

It can be verified that the complete model of **Approach A** outperforms the baseline in all metrics, also using the Neuraspace dataset. It is important to highlight that the Neuraspace dataset contains real data, which further increases the difficulty of the problem. Even so, the implemented model manages to outperform the naive forecast, 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 plot of Figure 7 presents the predicted values of the risk against the true risk values. It can be seen, again, that the

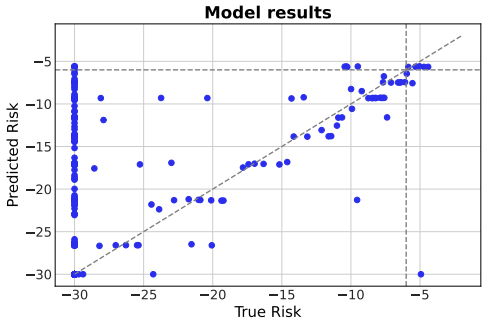

Fig. 7. Predicted vs True risk values, using **Approach A** and the Neuraspace data.

predictions tend to be arranged in 9 steps that correspond to the mean values of the emission distributions. The predictions are not completely aligned with the dashed diagonal line but tend to be close to the corresponding true risk values. It can also be concluded that the HMM would benefit from more data containing a more high-risk values since the HMM predictions seem to be truncated at an upper threshold. Figure 7 also shows that the implemented model largely over-predicts the −30 events. As in the CAC dataset, these over-predicted events have a risk evolution in high-risk values, but after the cut-off threshold experience a big risk transition to −30.

Figure 8 presents the 95% HDI associated with each prediction and the true risk, for the true high-risk events in the dataset. The prediction intervals capture the large majority

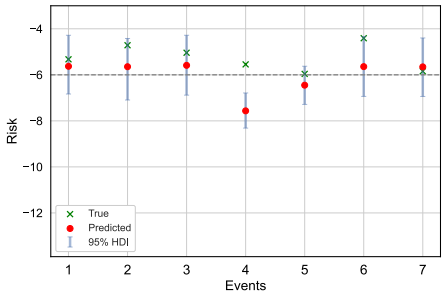

Fig. 8. Representation of the true risk values of each event (green), the HMM predictions (red) and the 95% HDI area (area shaded in blue), for all the high-risk events in the Neuraspace dataset, following **Approach A**.

of the high-risk values and, in one of the two cases where

the HMM fails to identify the high-risk conjunction, 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 8 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. This emphasizes that future work should focus on the development of a robust model that can identify −30 conjunctions without any miss-classifications of high-risk events.

B. Results of Approach B

In this approach, the chaser positional standard deviations are modeled and predicted, and risk is computed using Akella's and Alfriend's equation. Hence, this Subsection is divided into two parts: covariance prediction and risk computation. As previously explained, to test the performance of **Approach B**, only the Neuraspace dataset can be used.

1) Covariance prediction: To find the best number of hidden states for the three separate HMMs, K is iterated between 4 and 8 states, for each HMM. After cross-validation, the performance of the final HMMs for the covariance prediction is tested using the test set. In **Approach B**, the three components of the chaser position errors are predicted, but only the prediction results for σ_T^{chaser} are presented in the following, as an example.

Figure 9 shows the predicted along track position errors of the chaser against the true values for that parameter.

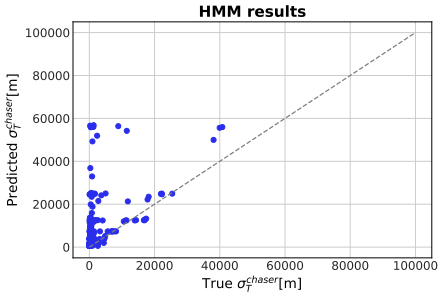

Fig. 9. Prediction results for the along-track position uncertainty σ_T^{chaser} against the true values.

The results show that the along-track position errors get largely over-predicted for events with final values of σ_T^{chaser} lower than, approximately, 10 000 m. The evolution of the chaser along-track position uncertainty for those events is depicted in Figure 10. All 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. In addition, it can be seen 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.

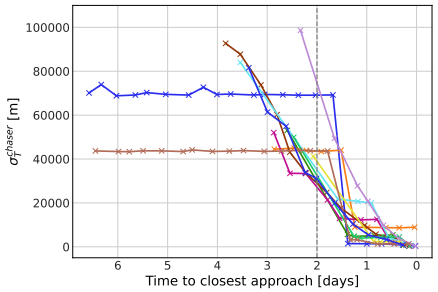

Fig. 10. Time series evolution of the chaser along-track prediction errors for the largely over-predicted events by the complete model of **Approach B**.

2) Risk Computation: After predicting the chaser positional standard deviations, the risk of collision is computed using Alfriend's and Akella's equation [6].

Table III presents the obtained performance metrics and the confusion matrix.

Metrics						
Model	RMSE	MAE	Precision	Recall	F_1	F_2
Approach B	2.88	0.46	18.5%	**62.5%**	0.286	0.424
Baseline	**2.66**	**0.41**	**29.4%**	**62.5%**	**0.400**	**0.510**
Confusion Matrix (Model | Baseline)						
		Pred. Low-Risk		Pred. High-Risk		
True Low-Risk		6840 | 6850		22 | 12	
True High-Risk		3 | 3		5 | 5	

TABLE III: Performance metrics and confusion matrix for the risk predictions of **Approach B** and baseline, using the Neuraspace dataset.

Table III shows that the complete model of **Approach B** is outperformed by the baseline solution in all metrics (with the exception of the recall). 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. As future work, it could also be advantageous to use more informative priors that take into consideration the physics of the problem.

Figure 11 shows the prediction results against the true risk values and it is possible to verify that with this method there is also a large over-prediction of the −30 events. In addition, for true risk values smaller than −15, the predictions tend to be far from the true regression line.

By taking random draws from the positional standard deviation prediction distributions, it is possible to obtain a credible interval for the final risk prediction of each event. Figure 12 represents the 95% HDI associated with each risk prediction, as well as the true risk values, for the high-risk events on the test set. The Figure 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, again, that there are 8 true high-risk conjunctions in the Neuraspace test set, but in the plot of

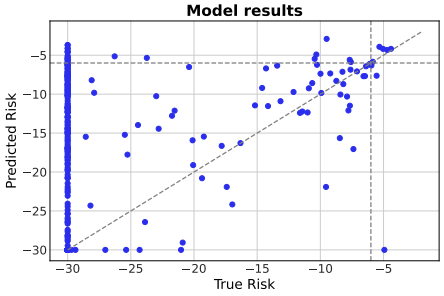

Fig. 11. Prediction results of the model of **Approach B** against the true values.

Figure 12 only 7 are represented, which can be justified by 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.

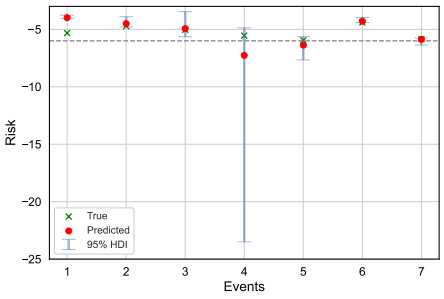

Fig. 12. Representation of the true risk values of each event (green), the HMM predictions (red) and the 95% HDI area (area shaded in blue), for the high-risk events in the Neuraspace test set, following **Approach B**.

VIII. Conclusions

One of the conclusions taken from the CAC results was that the naive forecast is a very strong predictor for the collision risk, indicating that the CDMs may follow the Markov property. This work tested this theory by benchmarking the performance of Bayesian HMMs, following two 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**). On the one hand, the second approach yields poor results in comparison with the baseline solution, indicating that the models should be trained with more data and that, in future work, more informative priors that take the physics of the problem into account should be used. On the other hand, the model of **Approach A** managed to outperform the baseline solution in all metrics, despite all the approximations made, the data imbalance problem, the fact that only the risk feature was used, and the complex behavior of the risk updates within the events. These promising results further add to the idea that the CDMs may follow the Markov property and suggest that this method should be further explored. In addition, this work provides a foundation for future research regarding the implementation of Bayesian HMMs to the challenge of applying ML in collision avoidance.

References

[1] ESA Space Debris Office. ESA's Annual Space Environment Report, 2022.

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

[3] 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.

[4] B. Bastida 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.

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

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

[7] 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. *Astrodynamics*, 6:121–140, 2022.

[8] J. Mueller and A. Thyagarajan. Siamese Recurrent Architectures for Learning Sentence Similarity. *Proceedings of the AAAI Conference on Artificial Intelligence*, 30(1), 2016.

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

[10] 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.

[11] 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.

[12] 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.

[13] O. Martin, R. Kumar, and J. Lao. *Bayesian Modeling and Computation in Python*. Chapman and Hall/CRC, Boca Raton, 2021.

[14] J. Salvatier, T. V. Wiecki, and C. Fonnesbeck. Probabilistic programming in Python using PyMC3. *PeerJ Computer Science*, 2:e55, 2016.

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

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

[17] 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.

[18] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[19] 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.

[20] 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.

[21] 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.

[22] A. Gelman and D. B. Rubin. Inference from Iterative Simulation Using Multiple Sequences. *Statistical Science*, 7(4):457–472, 1992.

[23] H. J. Thiébaux and F. W. Zwiers. The Interpretation and Estimation of Effective Sample Size. *Journal of Climate and Applied Meteorology*, 23(5):800–811, 1984.