Astrodynamics Vol. 6, No. 2, 121–140, 2022 https://doi.org/10.1007/s42064-021-0101-5

Spacecraft collision avoidance challenge: Design and results of a machine learning competition

Thomas Uriot[1], Dario Izzo[1](✉), Luís F. Simões[2], Rasit Abay[3], Nils Einecke[4], Sven Rebhan[4], Jose Martinez-Heras[5], Francesca Letizia[5], Jan Siminski[5], and Klaus Merz[5]

1. The European Space Agency, Noordwijk, 2201 AZ, the Netherlands
2. ML Analytics, Lisbon, Portugal
3. FuturifAI, Canberra, Australia
4. Honda Research Institute Europe GmbH, Offenbach 63073, Germany
5. ESOC, Space Debris Office, Darmstadt 64293, Germany

ABSTRACT

Spacecraft collision avoidance procedures have become an essential part of satellite operations. Complex and constantly updated estimates of the collision risk between orbiting objects inform various operators who can then plan risk mitigation measures. Such measures can be aided by the development of suitable machine learning (ML) models that predict, for example, the evolution of the collision risk over time. In October 2019, in an attempt to study this opportunity, the European Space Agency released a large curated dataset containing information about close approach events in the form of conjunction data messages (CDMs), which was collected from 2015 to 2019. This dataset was used in the Spacecraft Collision Avoidance Challenge, which was an ML competition where participants had to build models to predict the final collision risk between orbiting objects. This paper describes the design and results of the competition and discusses the challenges and lessons learned when applying ML methods to this problem domain.

KEYWORDS

space
debris
collision avoidance
competition
kelvins

Research Article

Received: 1 October 2020
Accepted: 27 January 2021

1 Introduction

The overcrowding of the low Earth orbit (LEO) has been extensively discussed in the scientific literature [1, 2]. More than 900,000 small debris objects with a radius of at least 1 cm have been estimated to be currently orbiting uncontrolled in the LEO①, posing a threat to operational satellites [3]. The consequences of an impact between orbiting objects can be dramatic, as the 2009 Iridium-33/Cosmos-2251 collision demonstrated [4]. While shielding a satellite may be effective for impacts with smaller objects [5], any impact of an active satellite with objects that have cross-sections larger than 10 cm is most likely to result in its complete destruction. Over the past decades, international institutions and agencies have become increasingly concerned with and contributed to defining guidelines to mitigate collision risk and preserve the space environment for future generations [6]. As a result, agencies, as well as operators and manufacturers, have been assessing a number of approaches and technologies in an attempt to alleviate this problem [7–9].

Despite all the efforts to actively control debris and satellite populations, this problem is still of increasing concern today. To illustrate the crowding of some areas of the LEO, we have visualized, as of 22 May 2020, the position of all 19,084 objects monitored by the radar and optical observations of the United States Space Surveillance Network (SSN) in Fig. 1. The figure clearly shows the density of objects at low altitudes, as well as the density drop around the northern and southern polar caps owing to the orbital dynamics being dominated by the main perturbations that, in LEO, act primarily on the argument of perigee and on the right ascension of the

✉ dario.izzo@esa.int
① Data from https://sdup.esoc.esa.int/discosweb/statistics/ (accessed on 3 June 2020).

ascending node [10].

To obtain a first assessment of the risk posed to an active satellite operating, for example, in a Sun-synchronous orbit, we computed the closest distance of a Sun-synchronous satellite to the LEO population and its distribution at random epochs and within a two-year window. Figure 2 shows the results for Sentinel-3B. In most of the epochs, the satellite was far from other objects, but in some rare scenarios, the closest distance approached values that were of concern. A Weibull distribution can be fitted to the obtained data, where results from extreme value statistics justify its use to make preliminary inferences on collision probabilities [11]. Such inferences are very sensitive to the Weibull distribution parameters and, in particular, to the behavior of its tail close to the origin.

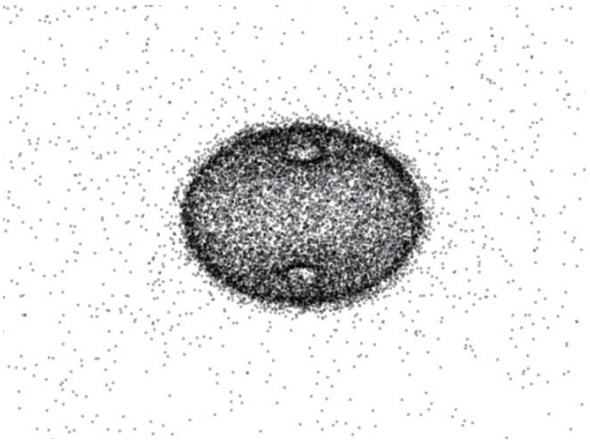


Fig. 1 Visualization of the density of objects orbiting the low Earth orbit as of 2020-May-22 (data from www.space-track.org).

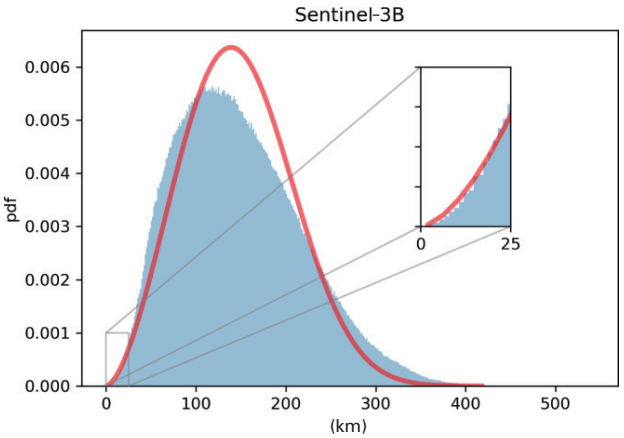


Fig. 2 Distribution of the distance between the closest object and Sentinel-3B, and a fitted Weibull distribution (fit skewed to represent the tail with higher accuracy).

This type of inference, as well as a series of resounding events, including the destruction of Fengyun-1C (2007), the Iridium-33/Cosmos-2251 collision (2009), and the Briz-M explosion (2012), convinced most satellite operators to include the possibility of collision avoidance maneuvers in the routine operation of their satellites [12].

In addition, the actual number of active satellites is steadily increasing, and plans for mega-constellations such as Starlink, OneWeb, and Project Kuiper [13] indicate that the population of active satellites is likely to increase in the coming decades. Thus, satellite collision avoidance systems are expected to be increasingly important, and their further improvement, in particular their full automation, will be a priority in the coming decades [14].

1.1 Spacecraft collision avoidance challenge

To advance the research on the automation of preventive collision avoidance maneuvers, the European Space Agency (ESA) released a unique real-world dataset containing a time series of events representing the evolution of collision risks related to several actively monitored satellites. The dataset was made available to the public as part of a machine learning (ML) challenge called the Collision Avoidance Challenge, which was hosted on the Kelvins online platform[①]. The challenge occurred over two months, with 96 teams participating, resulting in 862 submissions. It attracted a wide range of people, from students to ML practitioners and aerospace engineers, as well as academic institutions and companies. In this challenge, the participants were requested to predict the final risk of collision at the time of closest approach (TCA) between a satellite and a space object using data cropped at two days to the TCA.

In this paper, we analyze the competition's dataset and results, highlighting problems to be addressed by the scientific community to advantageously introduce ML in collision avoidance systems in the future. The paper is structured as follows: In Section 2, we describe the collision avoidance pipeline currently in place at ESA, introducing important concepts used throughout the paper and crucial to the understanding of the dataset. In Section 3, we describe the dataset and the details of its acquisition. Subsequently, in Section 4, we outline the competition design process and discuss some of the decisions made and their consequences. The

① Hosted at https://kelvins.esa.int/.

competition results, analysis of the received submissions, and challenges encountered when building statistical models of the collision avoidance decision-making process are the subjects of Section 5. In Section 6, we evaluate the generalization of ML models in this problem beyond their training data.

2 Collision avoidance at ESA

A detailed description of the collision avoidance process currently implemented at ESA is available in previous reports [15, 16]. In this section, we briefly outline several fundamental concepts.

The Space Debris Office of ESA supports operational collision avoidance activities. Its activities primarily encompass ESA's missions Aeolus, Cluster II, Cryosat-2, the constellation of Swarm-A/B/C, and the Copernicus Sentinel fleet composed of seven satellites, as well as the missions of third-party customers. The altitudes of these missions plotted against the background density of orbiting objects, as computed by the ESA MASTER①, are shown in Fig. 3.

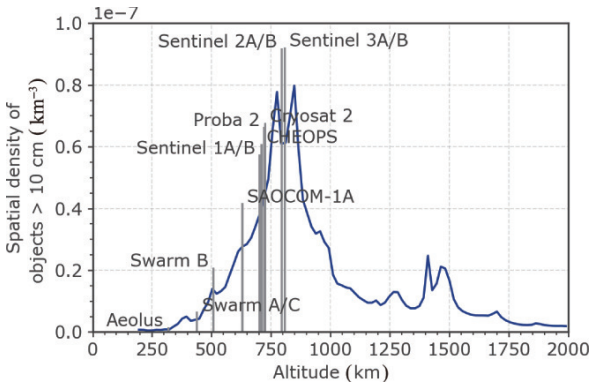


Fig. 3 Operational altitudes for the missions in LEO supported by ESA Space Debris Office, and the spatial density of objects with a cross-section of > 10 cm.

The main source of information of the collision avoidance process at ESA is based on conjunction data messages (CDMs). These are *ascii* files produced and distributed by the United States based Combined Space Operations Center (CSpOC). Each conjunction contains information on one close approach between a monitored space object (the "target satellite") and a second space object (the "chaser satellite"). The CDMs contain multiple attributes of the approach, such as the identity of the satellite in question, the object type of the potential collider, the TCA, the positions and velocities of the objects, and their associated uncertainties (i.e., covariances). The data contained in the CDMs are then processed to obtain risk estimates by applying algorithms such as the Alfriend–Akella algorithm [17].

In the days after the first CDM, regular CDM updates are received, and over time, the uncertainties of the object positions become smaller as the knowledge on the close encounter is refined. Typically, a time series of CDMs over one week is released for each unique close approach, with approximately three CDMs becoming available per day. For a particular close approach, the last obtained CDM can be assumed to be the best knowledge available on the potential collision and the state of the two objects in question. If the estimated collision risk for a particular event is close to or above the reaction threshold (e.g., 10^{-4}), the Space Debris Office will alarm control teams and begin planning a potential avoidance maneuver a few days prior to the close approach, as well as meeting the flight dynamics and mission operations teams. While the Space Debris Office at ESA provides a risk value associated with each CDM, to date, it has not attempted to propagate the risk value into the future. Therefore, a practical baseline that can be considered as the current best estimate would be to use the latest risk value as the final prediction. We introduce this estimate as the latest risk prediction (LRP) baseline in Section 4.4.

3 Database of conjunction events

The CDMs collected by the ESA Space Debris Office in support of collision avoidance operations between 2015 and 2019 were assembled into a database of conjunction events. Two initial phases of data preparation were performed. First, the database of collected CDMs was queried to consider only events where the theoretical maximum collision probability (i.e., the maximum collision probability obtained by scaling the combined target-chaser covariance) was greater than 10^{-15}. Here, the target refers to the ESA satellites, while the chaser refers to the space debris or object to be avoided. In addition, events related to intra-constellation conjunctions (e.g., for the Cluster II mission) and anomalous entries, such as scenarios with null relative velocity between the target and chaser, were removed. Finally, some events may cover a period during which

① Available at https://sdup.esoc.esa.int/master/.

the spacecraft performs a maneuver. In these scenarios, the last estimation of the collision risk cannot be predicted from the evolution of the CDM data, as the propulsive maneuver is not described. These scenarios were addressed by removing all CDM data before the maneuver epoch.

The second step in the data preparation was the anonymization of the data. This involved transforming absolute time stamps and position/velocity values in relative values, respectively, in terms of time to the TCA and state with respect to the target. The names of the target mission were also removed, and a numerical mission identifier was introduced to group similar missions. A random event identifier was assigned to each event. The full list of the attributes extracted from the CDMs and released in the dataset, as well as their explanations, are available on the Kelvins competition website.

Here, we briefly describe only a few attributes relevant to later discussions:

- *time_to_tca*: time interval between the CDM creation and the TCA (day).
- *c_object_type*: type of the object at a collision risk with the satellite.
- *t_span*: size of the target satellite used by the collision risk computation algorithm (m).
- *miss_distance*: relative position between chaser and target.
- *mission_id*: identifier of the mission from which the CDMs are obtained.
- *risk*: self-computed value at the epoch of each CDM, using the attributes contained in the CDM, as described in Section 2.

Table 1 provides an overview of the resulting database, indicating the number of entries (i.e., CDMs) and unique close-approach events. The risk computed from the last available CDM is denoted as r.

Table 1 Database of conjunction events at a glance

Characteristics	Number
Events	15,321
High-risk events ($r \geqslant 10^{-4}$)	30
High-risk events ($r \geqslant 10^{-5}$)	131
High-risk events ($r \geqslant 10^{-6}$)	515
CDMs	199,082
Average CDMs per event	13
Maximum CDMs per event	23
Minimum CDMs per event	1
Attributes	103

4 Competition design

The database of conjunction events constitutes an important historical record of risky conjunction events that occurred in LEO and creates the opportunity to test the use of ML approaches in the collision avoidance process. The decision on whether to perform an avoidance maneuver is based on the best knowledge one has of the associated collision risk at the time when the maneuver cannot be further delayed, i.e., the risk reported in the latest CDM available. Such a decision would clearly benefit from a forecast of the collision risk, enabling past evolution and projected trends to be considered. During the design of the Spacecraft Collision Avoidance Challenge, it was natural to begin from a forecasting standpoint, seeking an answer to the question: can an ML model forecast the collision risk evolution from available CDMs?

Such a forecast could assist the decision of whether or not to perform an avoidance maneuver by providing a better estimate of the future collision risk before further CDMs are released. Forecasting competitions are widely recognized as an effective means of determining good predictive models and solutions for a particular problem [18]. The successful designing of such competitions requires a good balance to be determined between the desire to create an interesting and fair ML challenge, motivating and involving a large community of data scientists worldwide, and fulfills the objective of furthering the current understanding by answering a meaningful scientific question [19].

Designing a competition to forecast r from the database of conjunction events presents a few challenges. First, the distribution of the final risk r associated with all the conjunction events contained in the database is highly skewed (Fig. 4), revealing how most events eventually result in a negligible risk. Intuitively, the effect is due to the uncertainties being reduced as the objects get closer, which in most scenarios results in close approaches where a safe distance is maintained between the orbiting objects. Furthermore, events that already require an avoidance maneuver are removed from the data, thus reducing the number of high-risk events. This is particularly troublesome as the interesting events, the ones that are to be forecasted accurately, are the few ones for which the final risk is significant. Second, there is significant heterogeneity in the various time series associated with

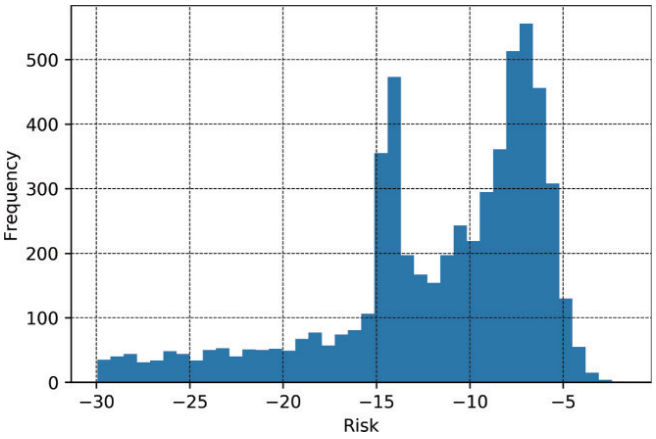


Fig. 4 Histogram of the latest known risk value (logarithmic scale) for the entire dataset (training and testing sets). Note that there are 9505 events with a final risk value of $\log_{10} r = -30$ or lower, which are not displayed in this figure.

different events, both in terms of the number of available CDMs and the actual *time_to_tca* at which CDMs are available, and most importantly, of the *time_to_tca* of the last available CDM that defines the variable r to be predicted. Therefore, the test and training sets and the competition metric were designed to alleviate these problems.

4.1 Definition of high-risk events

Many mission operators in LEO use 10^{-4} as a risk threshold to implement an avoidance maneuver. Over time, this value has been applied by default. However, the selection of a suitable reaction threshold for a particular mission depends on many different parameters (e.g., size of the chaser, target satellite), and its selection can be driven by considerations of the risk reduction that an operator seeks to achieve [20]. Therefore, ESA missions in LEO adopt reaction thresholds ranging between 10^{-5} and 10^{-4}. Events are monitored and highlighted when the collision risk is larger than a notification threshold, which is typically set to one order of magnitude lower than the reaction threshold. Note that in the remainder of this paper, the $\log_{10}$ of the risk value is used frequently, such that $\log_{10} r \geqslant -6$ defines high-risk events. Thus, we often omit writing log and simply refer to $r \geqslant -6$. For the objectives of the competition, a single notification threshold was used for all missions, and its value was set at 10^{-6}. The threshold value was selected to have a higher number of high-risk events while maintaining its value close to the more frequently used operational value of 10^{-5}. Figure 4 shows the risk computed from the last available CDM for all the close approach events in the database, revealing an abrupt increase in the risk value of -6. In particular, there were 30 events with $r > -4$, 131 events with $r > -5$, and 515 events with $r > -6$ (Table 1).

4.2 Test and training sets

ML algorithms learn relationships between inputs and outputs by maximizing a particular objective function. The aim is to automatically learn patterns from the training data that generalize to unseen data, known as the test set. Hence, the training and test sets must be obtained from similar data distributions. In addition, the data in the test set should reflect the type of data that we care about when deploying the ML model in the real world.

While releasing the raw database of conjunction events to the public was a priority, and thus provide the community with an unbiased set of information to learn from, the various models produced during the competition were tested primarily on predictions of events deemed particularly meaningful. Consequently, while the training and test sets originated from a split of the original database, they were not randomly sampled from it. Events corresponding to useful operational scenarios appeared in the test set.

In particular, for some events, the latest available CDM was days away from the (known) time to the closest approach, which made its prediction (also if correct) not a good proxy for the risk at the TCA. Furthermore, potential avoidance maneuvers were planned at least two days prior to the closest approach; thus, events that contain several CDMs at least two days prior to the TCA were more interesting. Overall, three constraints were imposed on the events to be eligible in the test set:

(1) The event had to contain at least two CDMs, one to infer from and one to use as the target.
(2) The last CDM released for the event had to be within a day (*time_to_tca* < 1) of the TCA.
(3) The first CDM released for the event had to be at least two days before the TCA (*time_to_tca* $\geqslant 2$) and all the CDMs that were within two days from the TCA (*time_to_tca* < 2) were removed.

Figure 5 depicts an example of an event that satisfied the requirements. Note that by permitting only events that satisfied the aforementioned requirements in the test set, the number of high-risk events was considerably

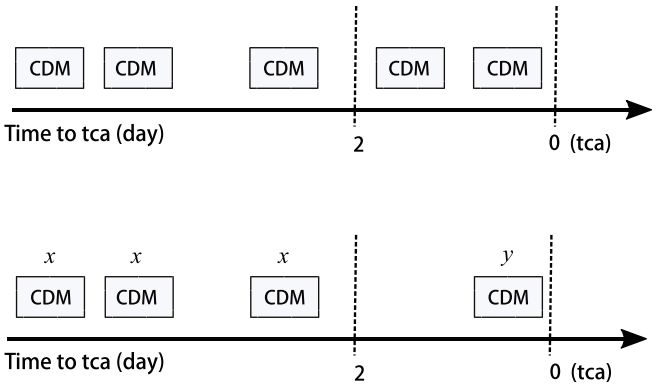


Fig. 5 Diagram depicting the raw CDMs time series for one event (top), and the same series if it was selected for the test set (bottom): only the CDMs prior to two days to TCA were made available (labeled as x) and the latest CDM was used as the target (labeled as y).

diminished. After enforcing the three requirements described above, only 216 high-risk events (out of 515) were eligible for the test set. Note that the remaining 299 high-risk events were maintained in the training set without being necessarily representative of the test events.

Because of the unbalanced nature of the dataset and the small number of high-risk events eligible for the test set, we decided to place most of the eligible events into the test set. Specifically, 150 eligible high-risk events were included in the test set and 66 in the training set. To alleviate the risk of directly probing the test set and thus overfitting, we limited the number of submissions per team to two per day during the first month of the competition and to a single submission per day during the second month.

4.3 Competition metric

In this section, we introduce the metric used to rank the participants and discuss its advantages and drawbacks. Several criteria were used to design a metric that could be fair and reward models of interest for operational objectives. The Spacecraft Collision Avoidance Challenge had two main objectives: (i) the correct classification of events into high- and low-risk events; (ii) the prediction of the risk value for high-risk events. In other words, whenever an event belonged to the low-risk class, the exact risk value was not important, and if an event belonged to the high-risk class, its exact value was of interest. Furthermore, because in the context of collision avoidance, false negatives were much more disastrous than false positives, their occurrences were to be penalized more. Finally, this was a highly unbalanced problem, where the proportion of low-risk events was much higher than that of high-risk events.

The final metric used considered these requirements and summarized them into one overall value to rank competitors. Eventually, the Spacecraft Collision Avoidance Challenge metric included both the classification and regression parts. Denoting the final risk as r and the corresponding prediction as $\hat{r}$, the metric can be defined as

$$L(\hat{r}) = \frac{1}{F_2}\mathrm{MSE}_{\mathrm{HR}}(r, \hat{r}) \tag{1}$$

where F_2 is computed over the entire test set using two classes (high final risk: $r \geqslant -6$, low final risk: $r < -6$) and $\mathrm{MSE}_{\mathrm{HR}}(r, \cdot)$ is only computed for high-risk events. More formally, we obtain

$$\mathrm{MSE}_{\mathrm{HR}}(r, \hat{r}) = \frac{1}{N^*}\sum_{i=1}^{N} \mathbb{1}_i (r_i - \hat{r}_i)^2 \tag{2}$$

where N is the total number of events, $N^* = \sum_{i=1}^{N} \mathbb{1}_i$ is the number of high-risk events, r_i and $\hat{r}_i$ are the true and predicted risks for the ith event, respectively, and

$$\mathbb{1}_i = \begin{cases} 1, & \text{if } r_i \geqslant -6 \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

Finally, the F score is defined as

$$F_\beta = (1 + \beta^2)\frac{p \times q}{(\beta^2 \times p) + q} \tag{4}$$

where β essentially controls the trade-off between precision and recall, denoted as p and q, respectively. A higher value of β means that a recall has more weight than precision; thus, more emphasis is placed on false negatives. To penalize false negatives more, we set $\beta = 2$.

While the metric encourages participants to have a higher F_2 score and a lower mean squared error, it introduces many layers of subjectivity. This is because the metric contains multiple sub-objectives that are combined into one meta-objective. In the denominator, the F_2 score is already an implicit multiobjective metric, where precision and recall are maximized to 1. Thus, there is a trade-off between precision and recall, which is controlled by β. In the numerator, the mean squared error penalizes erroneous predictions for high-risk events. The squaring is justified by the desire to penalize large errors.

In the metric defined in Eq. (1), F_2 functions as a scaling factor for $\mathrm{MSE}_{\mathrm{HR}}$, where F_2 assumes values in $[0, 1]$ and $\mathrm{MSE}_{\mathrm{HR}}$ in R^+, which means that the metric is largely dominated by $\mathrm{MSE}_{\mathrm{HR}}$ in the numerator.

Nonetheless, as reported in Section 5, even the highest-ranked models achieved a relatively small MSE_{HR}; thus, the F_2 scaling factor is appropriate.

In conclusion, several objectives were combined into one metric, which introduced some level of complexity and subjectivity. An alternative to the metric used in Eq. (1) would have a simple weighted average for each sub-metric (F_2 and MSE_{HR}). This scoring scheme is routinely used in public benchmarks such as the popular GLUE [21] score used in natural language processing, and it presents similar problems in the selection of weights that function as scaling factors.

Note that according to Eq. (1), as soon as an event is predicted to be low-risk ($\hat{r} < -6$), the optimal prediction to assign to the event is $\hat{r} = -6 - \epsilon$, where $\epsilon > 0$. Thus, for a false negative, we minimize MSE_{HR}, and for a true negative, the actual value does not matter, as long as $\hat{r} < -6$. Consequently, all risk predictions can be clipped at a value slightly lower than 10^{-6} to improve the overall score (or at least produce an equivalent score). In the remainder of this paper, we utilize this clipping, and the scores of the various teams are reported after the clipping has been applied, using $\epsilon = 0.001$.

4.4 Baselines

To have a sense of the effectiveness of a proposed solution, baseline solutions should be introduced. For the Spacecraft Collision Avoidance Challenge, two simple concepts can be used to build such baselines. Let us denote $\hat{r}_i$ and r_{-2_i} as the predicted risk and the latest known risk for the ith event, respectively (the subscript -2 reminds us that the latest known risk for a close-approach event is associated with a CDM released at least two days before the TCA, as shown in Fig. 5). The first baseline solution, called the constant risk prediction (CRP) baseline, is then defined as

$$\hat{r}_i = -5 \tag{5}$$

and has an overall score of $L = 2.5$. It constantly predicts the same value for the risk, and it was highlighted during the competition as a persistent entry in the leaderboard. Of the 97 teams, 38 managed to produce a better model.

One of the simplest approaches in time series prediction is the naive forecast [22], i.e., forecasting with the last known observation. This is known to be optimal for random walk data, and it operates well on economic and financial time series. Based on this fact, a second baseline solution, called latest risk prediction (LRP) baseline is defined as the clipped naive forecast:

$$\hat{r}_i = \begin{cases} r_{-2_i}, & \text{if } r_{-2_i} \geqslant -6 \\ -6.001, & \text{otherwise} \end{cases} \tag{6}$$

and has a score of $L = 0.694$ when evaluated on the complete test set. Of the 97 teams, 12 managed to submit better solutions. A few different teams obtained and utilized this baseline (or equivalent variants) in their submissions. The score achieved by the LRP is also reported in Table 3 and is plotted as a horizontal line in Fig. 9, along with the proposed solutions from the top ten teams. The LRP was of interest in this competition, as in any forecasting competition, because it provides a simple and yet surprisingly effective benchmark to improve upon.

4.5 Data split

This section discusses the splitting of the original dataset into training and test sets. First, principal component analysis (PCA) is applied to the data and it demonstrates that the attributes depend on the mission identifier. In other words, attributes recorded during different missions are not obtained from the same distribution, making it difficult to generalize from one mission to another. Next, we study the effect of different splits of the test data on the leaderboard scores (evaluated on a portion of the test set) and in the final ranking (evaluated on the full test set), using the LRP baseline solution.

In Fig. 6, the PCA projection of the original data is shown by maintaining only the first two principal components. While the first two principal components only account for 20% of the total variance, the projected data can still be distinguished and crudely clustered by

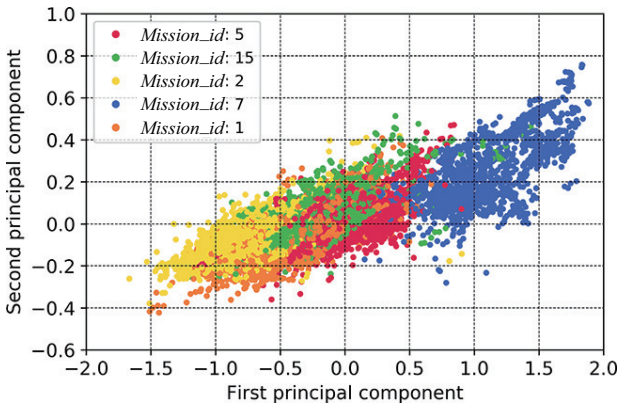


Fig. 6 Projection of the original CDMs, from the test set, onto the first two principal components, colored according to *mission_id*.

mission_id, in particular *mission_id*: 7 and *mission_id*: 2. This unsurprisingly implies that the attributes from the CDMs do not come from the same distribution, making it potentially difficult to generalize from one mission to another. Thus, each *mission_id* refers to a different satellite orbiting at different altitudes in regions of space with varying space debris density (Fig. 3).

Therefore, imbalances in mission type should not be created when splitting the data into training and test sets. Figure 7 shows that, for low-risk events, the missions are proportionally represented in both the training and test sets. However, when we examine only high-risk events (Fig. 8), we observe that the missions are not well distributed. In particular, *mission_id*: 2, *mission_id*: 15, and *mission_id*: 19 are over-represented in the test set, whereas *mission_id*: 1, *mission_id*: 6, and *mission_id*: 20 are under-represented. This is because the dataset was randomly split into training and testing, considering only the risk value and not the mission type. For future research, we recommended that mission type should be considered during the splitting of the dataset, or datasets with a higher homogeneity should be created with respect to the mission type. Note that further analysis of the dataset split and the correlation between the training and test sets are presented in Section 6.

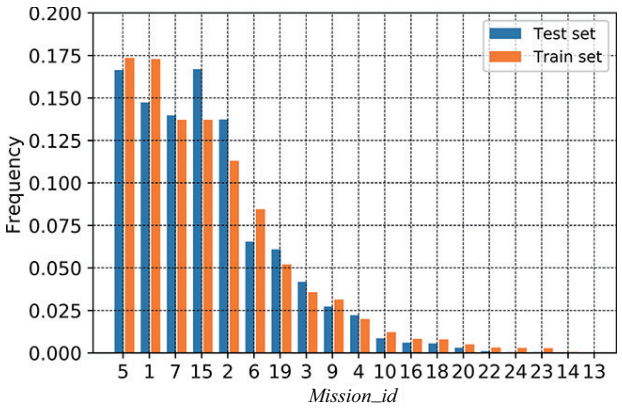


Fig. 7 Distribution of the mission type for the test and training sets for low-risk events.

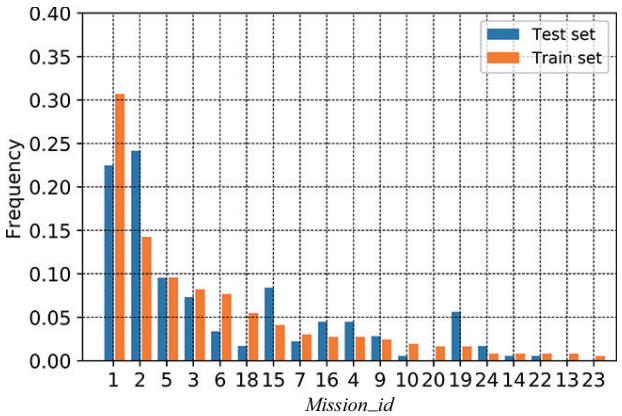


Fig. 8 Distribution of the mission type for the test and training sets for high-risk events.

5 Competition results

After the competition ended and the final rankings were made public, a survey was distributed to all the participating teams in the form of a questionnaire. The results of the survey, the final rankings, the methods of a few of the best-ranking teams, and a brief meta-analysis of all the solutions are reported in this section.

5.1 Survey

A total of 22 teams participated in the survey, including all top-ranked teams. Some questions from the survey were targeted to gather more information on the background of the participants. The questions were phrased as follows: "How would you describe your knowledge in space debris/astrodynamics?" A similar approach was conducted for ML and data science. The possible answers were limited to "professional", "studying", and "amateur". The answers are reported in Table 2, which shows that most participants had a background in ML and less in orbital mechanics. Note that the top three teams all identified themselves as ML professionals and two as studying orbital mechanics.

Table 2 Background of the participants, out of 22 respondents to the end of the competition questionnaire

Discipline	Proficiency		
	Professional	Student	Amateur
Machine learning	10	10	4
Orbital mechanics	4	5	15

As mentioned in Section 4.2 and reported in Table 1, the dataset for the collision avoidance challenge is highly unbalanced, with the training and test sets not randomly sampled from the dataset. A question from the survey probed whether the participants explicitly attempted to address class imbalance (e.g., by artificially balancing the classes, assigning importance weighing to samples) by asking, "Did you apply any approach to compensate for the imbalances in the dataset?" A total of 65%

of the participants answered positively. Furthermore, half of the participants reported attempting to build a validation set with similar properties and risk distribution as the test set, albeit failing since most surveyed teams lamented a poor correlation between training and test set performances.

One of the main scientific questions that this challenge aimed at addressing was whether the temporal information contained in the time series of CDMs was used to infer the future risk of collision between the target and chaser. A specific question from the survey asked participants if they found the evolution of the attributes over time useful to predict the final risk value. Surprisingly, 65% of the teams framed the learning problem as a static one, summarizing the information contained in the time series as an aggregation of attributes (e.g., using summary statistics, or simply the latest available CDM). This may have been a direct consequence of the great predictive strength of a naive forecast for this dataset, as outlined in the approaches implemented by the top teams in Section 5.3.

Finally, because of the small number of high-risk events in the test set and the emphasis placed on false negatives induced by the F_2 score, it is natural to ask whether teams probed the test set through a trial-and-error process. Overall, 30% of the participants (including the top-ranked team sesc, see Section 5.3) reported utilizing a trial-and-error method to identify high-risk events, suggesting that the difference between the test and training sets posed a significant problem for many teams, a fact that deserves some further insight, which we provide in Section 6.

5.2 Final rankings

96 teams participated to the challenge and produced a total of 862 different submissions during the competition timeframe. The scores on the leaderboard changed frequently, and the final ranking remained uncertain until the end of the competition. The evolution of the scores for the top ten teams throughout the competition is shown in Fig. 9. Note how the top four teams closely competed for first place until the very last days. Another observation is that while all the top teams managed to beat the LRP baseline, most of the teams required approximately 20 days to do so, implying that the LRP baseline was fairly strong. This was further supported by the fact that the scores did not improve much below the LRP baseline, suggesting that the naive forecast is

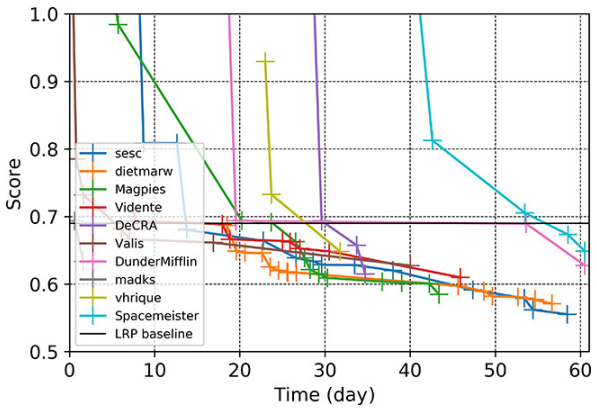


Fig. 9 Evolution of the scores of the submission of various top teams.

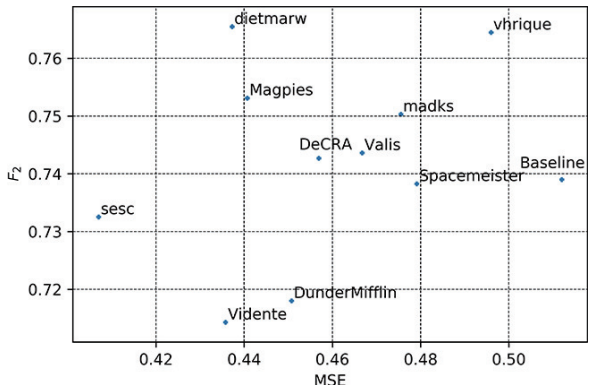
an important predictor of the risk value at the closest approach.

The final results, broken down into MSE_{HR}, with the risk clipped at -6.001 and the F_2 components, are shown in Table 3 for the top ten teams. All teams managed to improve upon the LRP baseline score by obtaining a better MSE_{HR}. However, many teams failed to obtain a better F_2 value than the LRP baseline.

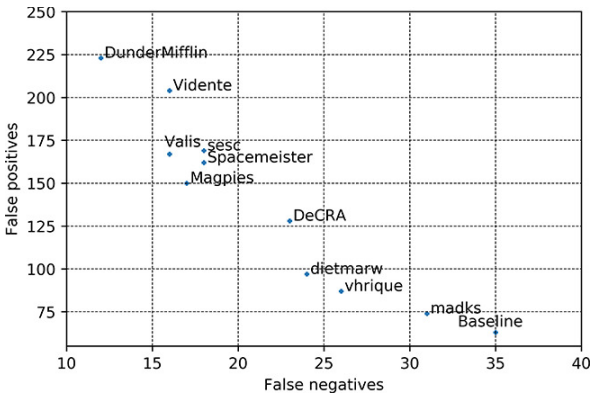
Table 3 Final rankings (from best to worst) evaluated on the test set, for the top ten teams. The best results are shown in bold

Team	Score	MSE_{HR}	F_2
sesc	**0.556**	**0.407**	0.733
dietmarw	0.571	0.437	**0.765**
Magpies	0.585	0.441	0.753
Vidente	0.610	0.436	0.714
DeCRA	0.615	0.457	0.743
Valis	0.628	0.467	0.744
DunderMifflin	0.628	0.451	0.718
madks	0.634	0.476	0.750
vhrique	0.649	0.496	0.764
Spacemeister	0.649	0.479	0.738
LRP baseline	0.694	0.513	0.739

To further investigate the differences between the F_2 score achieved by the teams and the LRP baseline solution, it is useful to examine the false positives and false negatives of each returned model (Fig. 10(b)). The Pareto front is very heterogeneous and consists of several teams: DunderMifflin, Valis, Magpies, DeCRA, dietmarw, vhrique, madks, and the baseline solution, denoted as Baseline. Although the baseline solution is in the Pareto front, we can observe that the resulting F_2 score in Fig. 10(a) is dominated by several teams. This is because the F_2 score places more emphasis on penalizing false



(a)



(b)

Fig. 10 On the top, in (a), the F_2 score and the MSE_{HR} are plotted for the top ten teams. On the bottom, in (b), the F_2 scores are broken into two components: false negatives (out of 150 positive events) and false positives (out of 2017 negative events).

negatives, which the baseline solution has the most of. In Fig. 10(a), only two teams remain in the Pareto front: sesc and dietmarw. Interestingly, dietmarw has the highest F_2 score, and sesc has the lowest MSE_{HR}, suggesting that their methods can be combined to achieve a better overall score.

5.3 Methods used by top teams

5.3.1 Team sesc

The highest-ranking team was composed of scientists from diverse domains of expertise: evolutionary optimization, ML, computer vision, data science, and energy management. In the early stages of the competition, the team attempted to use different methods, including extracting time series features [23], constructing an automated ML pipeline via Genetic Programming [24, 25], and using random forests. All these approaches were reported to have a score of $L \in [0.83, 1.0]$ on the test set, but they performed radically better on the training set. Such a difference was considered an indication that an automated, off-the-shelf ML pipeline was unlikely to be the appropriate way of learning from this dataset.

Instead, the team resorted to a step-by-step approach informed by statistical analysis, utilizing the metric and the constitution of the test set. Thus, the F_2 score is biased toward false negatives, and there is a relatively higher proportion of high-risk events in the test set than in the training set. Furthermore, we can observe that, in the training set, most of the high-risk events misclassified by the naive forecast have the latest risk r_{-2} only slightly below the threshold. A simple strategy is to promote borderline low-risk events to high-risk ones, thus improving the recall (at the cost of penalizing precision), which is what the F_2 score puts emphasis on. In practice, this strategy was implemented by introducing three thresholds, referred to as step 0, step 1, and step 2, as shown in Table 4 and Eq. (7).

Additional incremental improvements were achieved by assigning events to low risk whenever either the chaser type (*c_object_type* attribute) was identified as a payload and the diameter of the satellite (*t_span* attribute) was small (below 0.5) or the *miss_distance* was greater than 30,000 m. These steps are referred to as step 3, step 4, and step 5, respectively, in Table 4 and Eq. (7):

Finally, the risk value for high-risk events was clipped to a slightly lower risk value to enforce the general trend of risk decrease over time, thus improving the MSE_{HR} while preserving the F_2 score. This step is referred to as step 6 in Table 4, and Eq. (7).

In summary, the aforementioned observations resulted in the introduction of a cascade of thresholds:

$$\hat{r} = \begin{cases} -5.95, & \text{if } -6.04 \leqslant r_{-2} < -6.00 \quad \text{(step 0)} \\ -5.60, & \text{if } -6.40 \leqslant r_{-2} < -6.04 \quad \text{(step 1)} \\ -5.00, & \text{if } -7.30 \leqslant r_{-2} < -6.40 \quad \text{(step 2)} \\ -6.00001, & \text{if } \textit{c\_object\_type} \text{ is “payload”} \quad \text{(step 3)} \\ -6.00001, & \text{if } \textit{t\_span} < 0.5 \quad \text{(step 4)} \\ -6.00001, & \text{if } \textit{miss\_distance} > 30000 \quad \text{(step 5)} \\ -4.00, & \text{if } -4.00 \leqslant r_{-2} < -3.50 \quad \text{(step 6)} \\ -3.50, & \text{if } r_{-2} \geqslant -3.50 \quad \text{(step 6)} \end{cases} \tag{7}$$

5.3.2 Team Magpies

The third-ranked team was composed of a space situational awareness (SSA) researcher and Learning (ML) engineer. The team achieved its final score

Table 4 Evaluation of team sesc's approach, as additional steps were added

Combinations of steps	Training set			Test set			
	MSE_{HR}	F_2	Loss	MSE_{HR}	F_2	Loss	Leaderboard
LRP baseline	0.330	0.411	0.804	0.513	0.739	0.694	0.718
Steps: 0	0.330	0.430	0.768	0.512	0.753	0.680	0.703
Steps: 0 + 1	0.305	0.392	0.779	0.498	0.764	0.653	0.670
Steps: 0 + 1 + 2	0.290	0.296	0.982	0.445	0.738	0.603	0.612
Steps: 0 + 1 + 2 + 3	0.290	0.301	0.966	0.426	0.735	0.579	0.587
Steps: 0 + 1 + 2 + 4	0.290	0.298	0.974	0.447	0.735	0.608	0.611
Steps: 0 + 1 + 2 + 5	0.325	0.304	1.070	0.444	0.733	0.607	0.613
Steps: 0 + 1 + 2 + 6	0.293	0.296	0.990	0.424	0.738	0.575	0.581
Steps: 0 + 1 + 2 + 3 + 4 + 5 + 6	0.327	0.311	1.050	0.414	0.728	0.569	0.564
Steps: 0 + 1 + 2 + 5 + 6	0.327	0.304	1.077	0.424	0.733	0.578	0.581
Steps: 0 + 1 + 2 + 5 + 6 + 7	0.327	0.304	1.077	0.407	0.733	0.555	0.555

by leveraging Manhattan-LSTMs [26] and a siamese architecture based on recurrent neural networks. Team Magpies began by analyzing the dataset and filtering the training data according to the test set requirements described in Section 4.2. Subsequently, they selected seven out of 103 features (*time_to_tca*, *max_risk_estimate*, *max_risk_scaling*, *mahalanobis_distance*, *miss_distance*, *c_position_covariance_det*, and *c_obs_used*) by comparing the distribution difference of the non-anomalous event (last available collision risk is low and ends up low at close approach, and vice-versa for high-risk events) and anomalous events (last available collision risk is low and ends up high at the close approach, and vice-versa for high-risk events). Figure 11 shows the number of anomalous and non-anomalous scenarios. In addition to these seven attributes, three new features were included: the number of CDMs (*number_CDMs*) issued before two days, the mean (*mean_risk_CDMs*) and standard deviation (*std_risk_CDMs*) of the risk values of the CDMs.

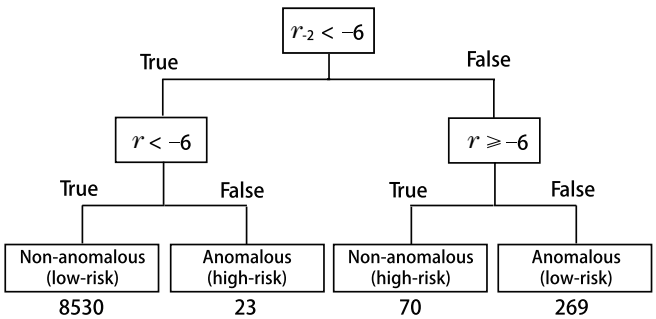


Fig. 11 Number of anomalous and non-anomalous events from the training set.

Hyperbolic tangents were used as activation functions, and Adam was used as the gradient descent optimizer [27]. The training data were split using a three-fold cross-validation (eight events were selected in each validation fold, from the 23 anomalous events). Subsequently, {non-anomalous, non-anomalous}, and {non-anomalous, anomalous} pairs were generated for the siamese network to learn similar and dissimilar pairs, respectively.

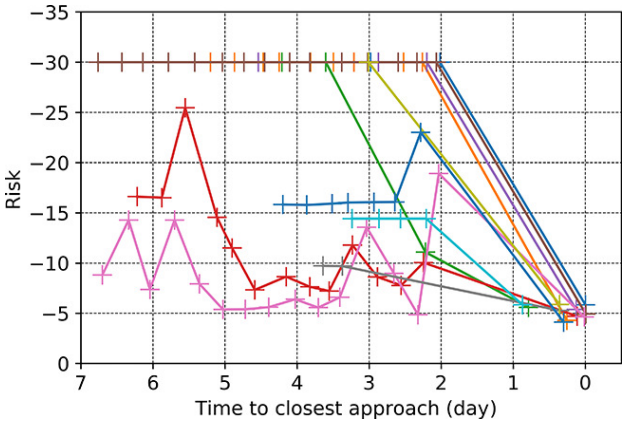
For each validation fold, several networks were trained using different hyperparameters. Networks that attained a reasonably high performance were then used in a majority voting ensemble scheme with equal weights. The majority vote outcome was denoted as f, used ten features as inputs, denoted as x, and predicted whether a low-risk event was anomalous or not. Then, the final predictions of the test set were expressed as

$$\hat{r} = \begin{cases} -6.001, & \text{if } r_{-2} < -6 \text{ and } f(x) = \text{non-anomalous} \\ -5.35, & \text{if } r_{-2} < -6 \text{ and } f(x) = \text{anomalous} \\ r_{-2}, & \text{if } r_{-2} \geqslant -6 \end{cases} \tag{8}$$

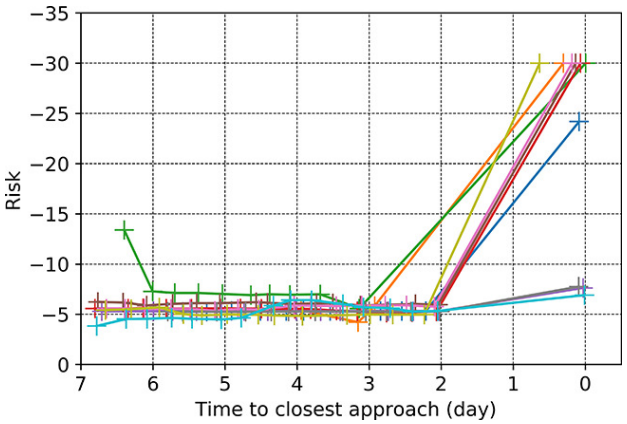
where -5.35 is the average risk value of all high-risk events in the training set.

5.4 Difficulty of samples

In this section, we investigate the events in the test set that were consistently misclassified by all the top ten teams. These events can be separated into two groups: false positives and false negatives. The false negatives correspond to events that were incorrectly classified as low risk and false positives correspond to events incorrectly classified as high risk. Figure 12 shows the evolution of the risk of events that were consistently misclassified. The figure shows that these events all experience a significant change in their risk value, as they progressed to the closest approach, thus rendering the use of the latest risk



(a)



(b)

Fig. 12 Events consistently misclassified by all the top ten teams: (a) false negatives, (b) false positives. In the top panel, we show all false negatives (11 out of 150 high-risk events). Each event is represented as a line, and the CDMs are marked with crosses. The evolution of the risk between two CDMs is plotted as a linear interpolation. In the bottom panel, we show ten randomly sampled events out of 62 false positives in total. These events were particularly difficult to classify because of the big leap in risk closer to the TCA, ranging from low risk to high risk in (a) and vice-versa in (b).

value misleading. Furthermore, as shown in Fig. 12, the temporal information is likely to be of little use to make good inferences in these scenarios: there is no visible trend and the risk value jumped from one extreme to the other (from very low to very high risk in (a) and vice-versa in (b)). One characteristic that all these events have in common is high uncertainties in their associated measurements (e.g., position and velocity), resulting in very uncertain risk estimates, susceptible to large jumps close to the TCA. Figure 13 shows the evolution of the uncertainty in the radial velocity of the target spacecraft (*t_sigma_rdot*) for the 150 high-risk events in the test set.

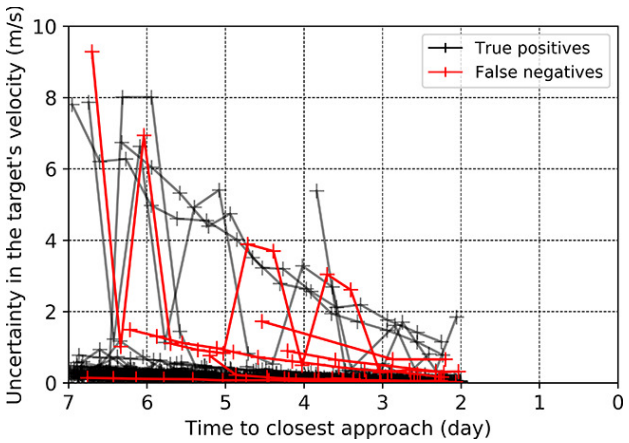


Fig. 13 Evolution of the uncertainty in the radial velocity of the target spacecraft (*t_sigma_rdot*) over time, up until two days to the TCA. The evolution of the uncertainty of the 11 false negative events (Fig. 12) is indicated in red. The evolution of the uncertainty of the 139 remaining true positive events is indicated in black.

The uncertainty values were generally higher for misclassified events. Note that many more uncertainty attributes were recorded in the dataset, and Fig. 13 shows only one of them. Higher uncertainties for the misclassified events suggests that there may be value in building a model which takes these uncertainties into account at inference time, for instance, by outputting a risk distribution instead of a point prediction.

6 Post competition ML

Further ML experiments were conducted on the dataset, both to analyze the competition and to further investigate the use of predictive models for collision avoidance. The aim of these experiments was to understand the difficulties experienced by competitors in this challenge and to gain deeper insights into the ability of ML models to learn generalizable knowledge from these data.

6.1 Training/test set loss correlation

The first experiment was designed to analyze the correlation between the performance of ML models on the training and test sets used during the competition. Only the training set events conforming to the test set specifications (Section 4.2) were considered: final CDM within a day of the TCA, and all other CDMs at least two days away from the TCA. The last CDM that is at least two days away from the TCA, the most recent CDM available to operators when making the final planning decisions, was used here as the sole input to the model.

In other words, temporal information from the CDM time series was not considered. From that CDM, only the numerical features were used; the two categorical features (*mission_id* and *object_type*) were discarded. Thus, for models to learn mission or object-specific rules, they would have to utilize features encoding relevant properties of that feature or object. It was hoped that this would force the model to learn more generalizable rules. In addition to this step, no other transformations were applied to the CDM raw values (such as scaling or embedding). Similarly, no steps were implemented to impute the missing values that occurred at times in many of the CDM variables. We left these to be addressed by the ML algorithm (LightGBM in our case) through its own internal mechanisms. Most importantly, the model's target was defined as the change in risk value between the input CDM and the event's final CDM ($r - r_{-2}$), rather than the final risk r itself. This facilitated the learning task as it implicitly reduced the bias toward the most represented final risk (i.e., -30). Furthermore, it enabled a direct comparison to the LRP baseline, as the various models were de facto tasked to predict a new estimator (h) such that $r = \text{LRP} + h$. The quantity h was further encoded through a quantile transformer to assume a uniform distribution.

Eventually, the training data consisted of a table of 8293 CDMs from as many events, each described by 100 features. Each CDM was assigned a single numerical value that was to be predicted. Overall, these steps resulted in a simplified data pipeline. Note the absence of any steps to address the existing class imbalance and the absence of any focus on high-risk events during the training process. Models were requested to learn the evolution of risk across the full range of risk values, although they were mostly assessed on their performance at one end of the range of risk values during evaluation. The competition's MSE_{HR} metric was obtained only over the true high-risk events, and the use of clipping at a risk of -6.001 further ignored where the final predicted risk lay if it fell below this value. In addition, F_2, a classification metric, cared only for where the risk values lay with respect to the -6 threshold.

For the type of regression problem outlined above, with tabular data representation, gradient boosting methods [28] offer state-of-the-art performance. Thus, we selected the LightGBM gradient boosting framework [29] to train many models. To attain both training speed and model diversity, we changed the hyperparameters as follows (with respect to the default in the LGBMRegressor of LightGBM 2.2.3): the n_estimators was set to 25, feature_fraction to 0.25, and learning_rate to 0.05. Together, these settings resulted in an ensemble with fewer decision trees (the default is 100), and each tree was trained exclusively on a small random subset of 25% of the available features (the default is 100%), and each successive tree had a reduced capability to overrule what previous trees had learned (the default learning rate, also known as shrinkage in the literature, is 0.1).

Figure 14 shows the evaluations of 5000 models, on the training and test sets, on the MSE_{HR} and $1/F_2$ metrics, as well as their product (the competition's score, or loss metric). The risk values were clipped at -6.001 prior to measuring the MSE_{HR}. We compared the performance of the models on the training (x-axis) and test sets (y-axis).

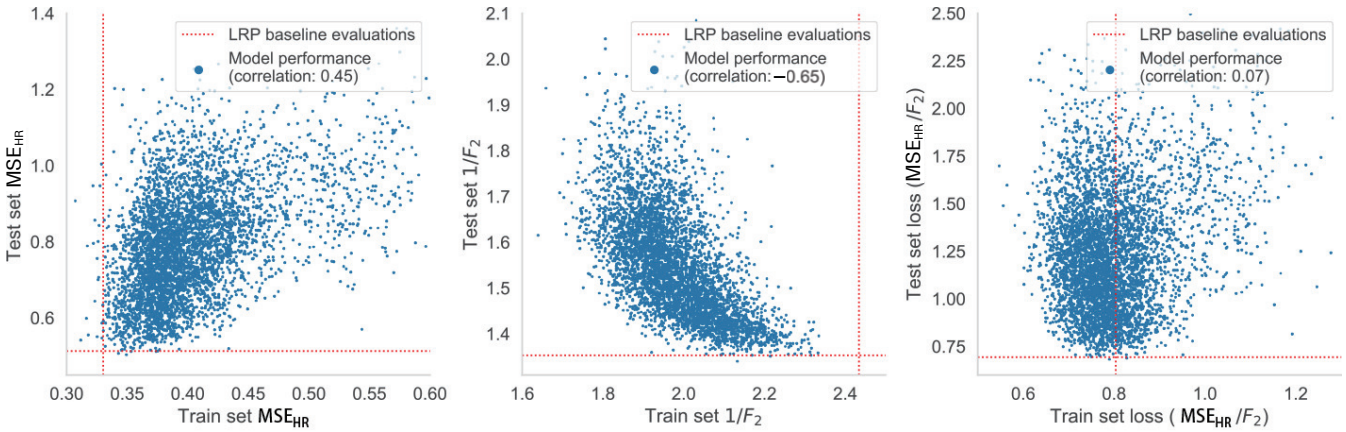


Fig. 14 Performance levels achieved by 5000 gradient boosting regression models trained using the competition's dataset split.

As a reference, the dotted lines show the performance of the LRP baseline (Section 4.4). The Spearman rank-order correlation coefficient was computed as an indicator of the extent to which performance in the training set generalizes to the test set.

Only one model (0.02% of the trained models) outperformed the LRP baseline loss in both the training and test sets. With a test set loss of 0.684 (1.4% gain over LRP), that model would have ranked 11 th in the official competition.

Overall, Fig. 14 shows several undesirable trends. The $\mathrm{MSE}_{\mathrm{HR}}$ plot exhibited a positive correlation: the training set performance was predictive of the performance on the test set. However, the models struggled to improve the LRP in both sets. Most models degraded the risk estimate available in the most recent CDM. In $1/F_2$, we observed a strong negative correlation: the better the performance on the training set, the worse it was on the test set. This was a clear sign of overfitting. When aggregated, we remained with a loss metric displaying essentially no correlation. This observation, while bound to the modeling choices made, offers a possible explanation for competitors' sentiment of disappointment over models that were good in their local training setups evaluating poorly on the leaderboard.

6.2 Simulating 10,000 virtual competitions

To further our understanding of the absence of a significant Spearman rank correlation between training and test set performances, as highlighted in Fig. 14, we simulated 10,000 possible competitions that differed in the data split. In each, a test size was randomly selected from a set of 19 options, containing the values from 0.05 through 0.95 in steps of 0.05. This setting indicates the fraction of events that should be randomly selected to be moved to the test set. The full dataset being partitioned was composed solely of the 10,460 events that conformed to the official competition's test set specifications. We adopted a different splitting procedure from that reported in Section 4.5. A stratified shuffle splitter was used, so the proportions of final high-risk events in both the training and test sets would always match the proportion observed in the dataset being partitioned (2.07%) as closely as possible. For reference, a test size of 0.2 results in 172.8 high-risk events on average in the training set, and 43.2 in the test set (and 8195.2 and 2048.8 low-risk events, respectively). In the training and test sets, no allowances were made to preserve the event distributions of mission ids and chaser object types present in the full dataset being partitioned. As shown in Figs. 7 and 8, the fraction of events from the different missions had such an imbalance that many of these generated splits likely either resulted in some missions being entirely unrepresented in either the training or test set or having such low volumes as to render the learning of their properties unlikely. Similarly, the object type attribute had an identical imbalance, which led to similar challenges. Although this partitioning process made achieving a higher score on the performance metrics more difficult, it served the current aim of evaluating the generalization capability.

In each of the 10,000 virtual competitions, 100 regression models were trained using the same data pipeline and model settings as described in the previous section. On average, 526 competitions were simulated for each of the 19 different test size settings, each with its own distinct data split. In total, 1 million models were trained. Although framed here as virtual recreations of the Kelvins competition, this process implemented, per test size setting, a Monte Carlo cross-validation or repeated random sub-sampling validation [30]. If the number of random data splits approached infinity, the results would tend toward those of leave-p-out cross-validation.

The experiment's results are shown in Figs. 15–19. Figure 15 shows statistics on the Spearman rank-order correlation coefficients between model evaluations in the training and test sets per evaluation metric. A positive Spearman correlation signals the ability to use the metric for model selection. The better a model is on the training set, the better we expect it to be on the unseen events of the test set. A negative correlation is a sign of overfitting or inability to generalize beyond the training set data. Figure 16 complements the analysis in Fig. 15 by showing the statistics on the percentage of the models per simulated competition that outperformed the LRP baseline in their respective training and test sets. The curves show the mean performance as a function of the test size, and the shaded areas represent the region within one standard deviation. Figure 17 also shows the correlations between the training and test set evaluations, but now matches $\mathrm{MSE}_{\mathrm{HR}}$ correlations to $1/F_2$ correlations. Thus, an overview of the effect of the same data split on models' capabilities to learn

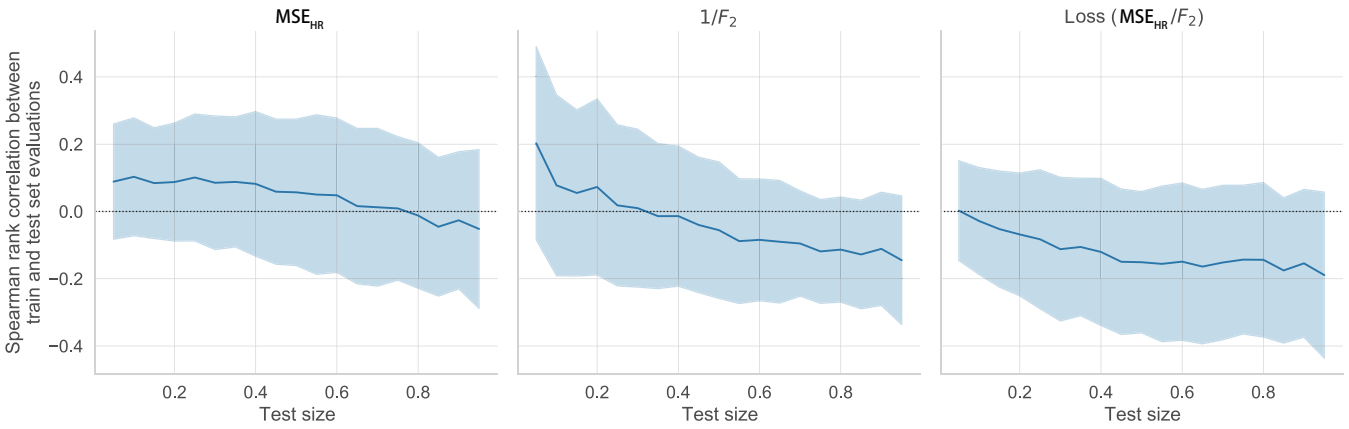


Fig. 15 Extent to which different data splits affected the ability to infer test set performance from the training set performance. Expected Spearman rank-order correlation coefficients between training and test set evaluations, as data sets vary in the fraction of events assigned to both (shown in the x-axis). Correlations measured in the $\mathrm{MSE}_{\mathrm{HR}}$ and $1/F_2$ metrics, as well as their product.

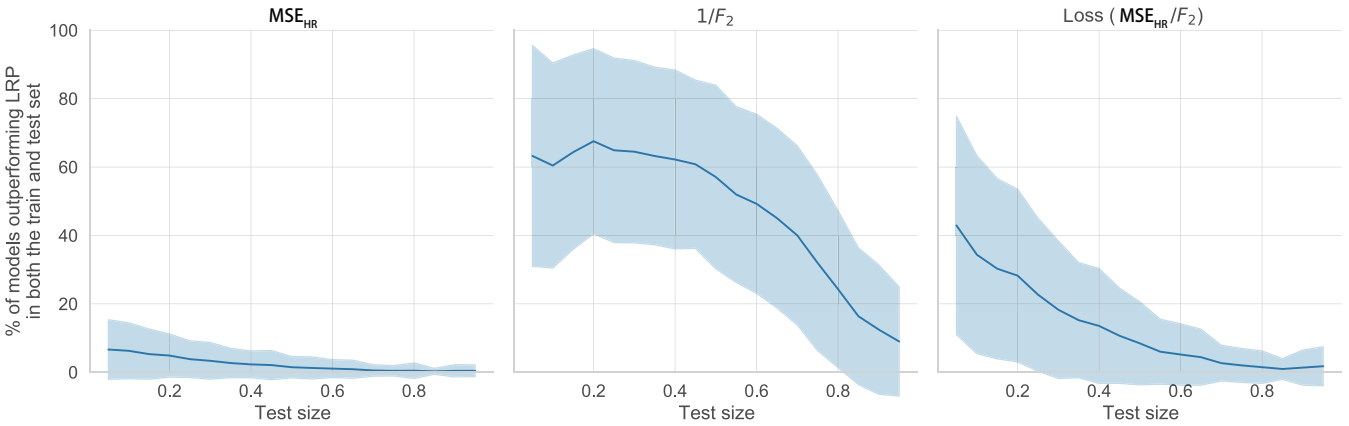


Fig. 16 Expected percentage of ML models that would outperform the LRP baseline in both the training and test set, as data sets varied in the fraction of events assigned to both (shown in the x-axis). Performance measured in the $\mathrm{MSE}_{\mathrm{HR}}$ and $1/F_2$ metrics, as well as their product.

generalizable knowledge simultaneously with respect to regression and classification objectives can be obtained. The red star places the Kelvins competition's unstratified (with respect to high-risk events) data split in the context of 10,000 stratified splits, indicating how much of an outlier it turned out to be.

The first conclusion to be drawn from these figures is that the aggregated loss metric, $\mathrm{MSE}_{\mathrm{HR}}/F_2$, was decidedly uninformative in terms of identifying models that were likely to generalize. It required two metrics that by themselves displayed a low correlation between the training and test set and aggregated them into a single value, which was even less correlated. Furthermore, as shown in Fig. 17, the highest loss correlations tended to occur when the $\mathrm{MSE}_{\mathrm{HR}}$ was highly correlated. The $\mathrm{MSE}_{\mathrm{HR}}$ was of the three metrics the one that tended to display a higher rank correlation. However, as shown in Fig. 16, few models outperformed the $\mathrm{MSE}_{\mathrm{HR}}$ obtained by the LRP baseline on both sets. This indicated an identical scenario to that shown in Fig. 14, in which we obtained a high positive correlation, but the models were not particularly successful. Predicting the actual final risk value was difficult; therefore, the further our predictions moved away from the most recent risk estimate in high-risk events (the only events scored by this metric), the worse we were likely to perform, both in the training and test sets. Nonetheless, as shown in the $1/F_2$ plots in Figs. 15 and 16, even if the predicted final risk values were not accurate, those perturbations moved the values across the -6 risk threshold to result in improved capability to forecast the final risk class. With a test size of 0.2, 67.55% of the trained models outperformed the LRP baseline

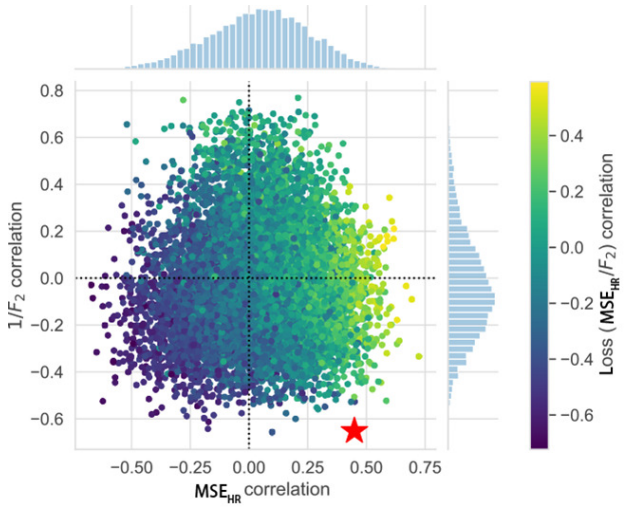


Fig. 17 Extent to which different data splits affected the ability to infer test set performance from the training set performance, as observed through simultaneous evaluations of regression and classification metrics. Spearman rank-order correlation coefficients between training and test set evaluations of the three performance metrics in 10,000 different data splits using different test size fractions. The red star corresponds to the data split of the official competition (Fig. 14).

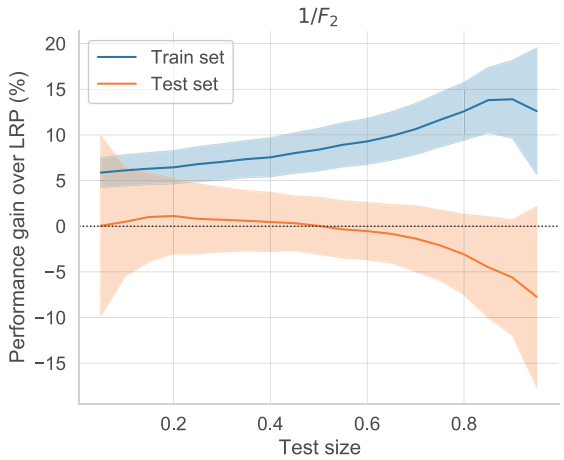


Fig. 18 Expected $1/F_2$ performance gain (%) over the LRP baseline as data splits varied in the fraction of events assigned to either the training or test set.

on both their training and test sets. This was in stark contrast to Fig. 14, where only 0.04% of the models (two out of 5000) outperformed the LRP baseline for the $1/F_2$ metric.

By normalizing models' $1/F_2$ evaluations with respect to the LRP baseline $1/F_2$ values, performance became comparable across different data splits. Figure 18 shows the mean and standard deviation of models' percentage gains in performance over the LRP baseline in the training and test sets. Statistics were calculated across all models trained over all the data splits that used the same test size setting. Figure 19 shows models' $1/F_2$ evaluations in the training and test sets, normalized against their respective LRP $1/F_2$ baseline, for selected test size settings. Over 50,000 ML models are shown in each subplot, trained over 500 data splits on average with that test size setting.

At one end, with a test size of 0.95, training sets had merely 523.0 events to learn from (10.8 of which are of high risk). With insufficient data to learn from, the models quickly overfit and failed to learn generalizable rules. This was indicated by a mean gain in performance of 12.61% on the training set, but a 7.73% mean loss in performance on the test set, both with respect to the LRP baseline. As we increased the amount of data available for training models, the training set performance decreased (more data patterns to learn from and harder to incorporate individual event idiosyncrasies into the model), but test performance increased. At the other end, with a test size of 0.05, most data were available for training, but the small test set was no longer representative (523.0 events to evaluate models on, 10.8 of high risk). Depending on the "predictability" of events that ended up on the test set, we either obtained a very high or very low performance: a mean gain of 0.04% on the test set, with a standard deviation of 9.83%. Here, the optimal trade-off lay in a test size of 0.2, where a 6.45% gain in the training set performance over the LRP baseline translated into a 1.12% gain in the test set. It is common for data scientists to use 80/20 splits of the dataset, as a rule of thumb inspired by the Pareto principle. Note we experimentally converged as this being the ideal setting.

To establish an ML performance baseline, we now turn directly to the F_2 score rather than its inverse (see the discussion in Section 4.3). F_2, which ranges in $[0, 1]$, is the harmonic mean of precision and recall, where recall (ability to identify all high-risk events) is valued two times higher than precision (ability to ensure all events predicted as being of high-risk indeed are). A Monte Carlo cross-validation with a test size of 0.2 and 505 stratified random data splits evaluated the LRP baseline (the direct use of an event's latest CDM's risk value as prediction) to a mean F_2 score of 0.59967 over the test set (standard deviation: 0.04391). Over the same data splits, a LightGBM regressor, acting over the same CDM raw values (see data and model configurations in

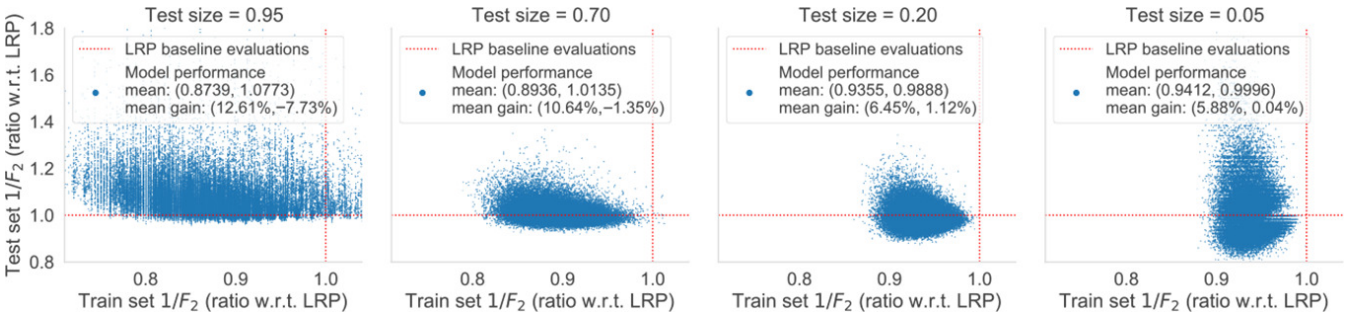


Fig. 19 Variation in the training and test set performance of the ML models ($1/F_2$) as a function of data availability. Performance normalized with respect to LRP baseline performance in the same datasets.

Section 6.1), evaluated to a mean F_2 score of 0.60732 over the test set (standard deviation: 0.04895; statistics over 50500 trained models). Therefore, a gain of 1.2743% over the LRP baseline[①]. The difference in performance between both approaches is statistically significant: a paired Student's t-test rejects the null hypothesis of equal averages (t-statistic: -10.23, two-sided p-value: 1.83×10^{-22}; per data split, the LRP F_2 score was paired with the mean LightGBM model F_2 score to ensure independence across pairs).

This is the strongest evidence yet that ML models can indeed learn generalizable knowledge in this problem. In a domain that safeguards assets valued in millions, a 1% gain in risk classification can already be transformative. Furthermore, this would be a 1% gain on top of approaches for risk determination that have been developed for decades. Note that these results, obtained using a classification metric, were achieved through a regression modeling approach. Furthermore, it had an intentionally limited modeling capability, was trained over a basic data preparation process, and was evaluated under adverse conditions (owing to imbalances in the mission id and type of chaser object). Thus, we expect it will be possible to significantly surpass these performance levels with more extensive research in data preparation and modeling.

6.3 Feature relevance

The experiment described in the previous section provides a basis on which to quantify feature relevance that is independent from the specifics of any particular data split or the decision-making of any individual model. We provide that information here, to illustrate what signal ML models use to arrive at their predictions, and to direct future research towards the more important features to train models on.

Of the 1 million models trained in the previous section's experiment, 47.75%, from across different test size settings, surpassed the $1/F_2$ LRP baseline on both their training and test sets. We selected all those models and used LightGBM to quantify their "gain" feature relevance. This process did not measure how frequently a feature was used across a model's decision trees, but rather the gains in loss it enabled when that feature was used (loss here refers to the objective function optimized by the algorithm while building the model, not to the competition's $\mathrm{MSE_{HR}}/F_2$ scoring function). For each model, relevance values were normalized over the features' total gains and converted to percentages. Subsequently, the values were aggregated through weighted statistics across the selected models, resulting in the relevance assessments shown in Table 5 (only the top twenty features are shown, out of the 100 used). The models' fractional gains in performance over the test sets' LRP $1/F_2$ baseline were used as weights.

The LRP is a strong predictor, as previously discussed. However, relevance measurements indicated that in the ML models, features directly related to risk (*risk*, *max_risk_scaling*, *max_risk_estimate*) together accounted for only half (54.44%) of the models' gains in loss. Models widely used the information available to them, with the top twenty features in Table 5 accounting for 78.32% of the gains in loss, and only two of the 100 features having a relevance of 0.0.

A set of 40 features had values for both the "target" (the ESA satellite – prefix *t_*), and "chaser" that should be avoided (space debris/object – prefix *c_*), for a total

① For comparison, cross-validation of team sesc's method (Section 5.3.1), over the same 505 data splits with test size of 0.2, evaluated to a mean F_2 score of 0.51563 over the test set. This was a performance loss of 14% with respect to the LRP baseline.

Table 5 Feature relevance estimates. In the prediction of near-term changes in risk, percentage of the reduction in error attributable to the feature. A description of the features is available on the Kelvins website

Feature	Rank	Mean	Std. dev.
risk	1	29.275	9.557
max_risk_scaling	2	22.544	8.979
mahalanobis_distance	3	3.261	1.675
c_sigma_t	4	3.000	1.715
max_risk_estimate	5	2.624	1.367
c_sigma_rdot	6	2.191	1.369
miss_distance	7	2.089	1.112
c_position_covariance_det	8	1.778	1.066
c_sigma_n	9	1.312	0.625
time_to_tca	10	1.236	0.517
c_sigma_r	11	1.177	0.739
c_obs_used	12	1.164	0.554
c_sigma_ndot	13	0.964	0.437
relative_position_n	14	0.954	0.754
c_recommended_od_span	15	0.945	0.423
relative_position_r	16	0.835	0.440
c_sedr	17	0.779	0.486
SSN	18	0.773	0.372
c_crdot_t	19	0.718	0.468
relative_speed	20	0.699	0.400

of 80 of the 100 features. Note the absence of "target" features in Table 5. The relevance of "target" features summed to a total of 9.41%, while "chaser" features summed to 23.49%. If the models were to rely too much on the properties of the "target", they would be learning mission-specific rules. Instead, we observed a greater reliance on properties of the "chaser", and in features with relative values, thus enabling better generalization across missions.

The mean relevance estimates were very stable. The unweighted aggregation of normalized relevance values in the remaining 52.25% of trained models not included in the selection above had a total of 10.51% absolute difference across features. The higher-performing models from which the statistics in Table 5 were drawn exhibited by comparison a greater reliance on *risk* and *max_risk_scaling* (+4.62%). The SSN, the Wolf sunspot number, at a rank of 18, was one of the most relevant features. It was also one of the features with a greater increase with respect to the alternate ranking, climbing three positions, and increasing relevance by 0.07%.

Note that models under consideration used CDM raw values as inputs. After some feature engineering, the attributes presented in Table 5 may follow a different ranking. A result of their information content with respect to the prediction target becoming clearer to identify and use by the ML algorithms. Note also that correlated features may have split relevance values between them, causing them to appear lower in this ranking.

7 Conclusions

The Spacecraft Collision Avoidance Challenge enabled, for the first time, the study of the use of ML methods in the domain of spacecraft collision avoidance owing to the public release of a unique dataset collected by the ESA Space Debris Office over more than four years of operations. Several challenges, mostly derived from the unavoidable unbalanced nature of the dataset, had to be accounted for to release the dataset in the form of a competition and the use of automated, off-the-shelf ML pipelines were limited. Nevertheless, the competition results and further experiments presented here clearly demonstrated two things. On one hand, naive forecasting models have surprisingly good performances and thus are established as an unavoidable benchmark for any future work in this subject; on the other hand, ML models can improve upon such a benchmark, hinting at the possibility of using ML to improve the decision-making process in collision avoidance systems.

Acknowledgements

The ESA would like to thank the Unite States Space Surveillance Network for the agreement that enabled the public release of the dataset for the objectives of the competition.

The authors would like to thank all the scientists that participated in the Spacecraft Collision Avoidance Challenge and that dedicated their time and knowledge to an important element of ESA's operated satellites.

In particular, we would like to acknowledge all members of team sesc, whose methodology is briefly described in this paper: Steffen Limmer, Sebastian Schmitt, Viktor Losing, Sven Rebhan, and Nils Einecke.

Declaration of competing interest

The authors have no competing interests to declare that are relevant to the content of this article.

References

[1] Liou, J. C., Johnson, N. L. Instability of the present LEO satellite populations. *Advances in Space Research*, **2008**, 41(7): 1046–1053.

[2] Krag, H. Consideration of space debris mitigation requirements in the operation of LEO missions. In: Proceedings of the SpaceOps 2012 Conference, **2012**.

[3] Klinkrad, H. *Space Debris*. Springer-Verlag Berlin Heidelberg, **2006**.

[4] Anselmo, L., Pardini, C. Analysis of the consequences in low Earth orbit of the collision between Cosmos 2251 and Iridium 33. In: Proceedings of the 21st International Symposium on Space Flight Dynamics, **2009**: 2009-294.

[5] Ryan, S., Christiansen, E. L. Hypervelocity impact testing of advanced materials and structures for micrometeoroid and orbital debris shielding. *Acta Astronautica*, **2013**, 83: 216–231.

[6] IADC. IADC space debris mitigation guidelines. Available at https://www.iadc-home.org/ (cited in 2007).

[7] Walker, R., Martin, C. E. Cost-effective and robust mitigation of space debris in low earth orbit. *Advances in Space Research*, **2004**, 34(5): 1233–1240.

[8] Biesbroek, R., Innocenti, L., Wolahan, A., Serrano, S. M. e. Deorbit-ESA's active debris removal mission. In: Proceedings of the 7th European Conference on Space Debris, **2017**: 10.

[9] Liou, J. C., Johnson, N. L., Hill, N. M. Controlling the growth of future LEO debris populations with active debris removal. *Acta Astronautica*, **2010**, 66(5–6): 2288–2294.

[10] Izzo, D. Effects of orbital parameter uncertainties. *Journal of Guidance, Control, and Dynamics*, **2005**, 28(2): 298–305.

[11] Smirnov, N. N. *Space Debris: Hazard Evaluation and Debris*. CRC Press, **2001**.

[12] Flohrer, T., Braun, V., Krag, H., Merz, K., Lemmens, S., Virgili, B. B., Funke, Q. Operational collision avoidance at ESOC. In: Proceedings of the Deutscher Luft-und Raumfahrtkongress, **2015**.

[13] Logue, T. J., Pelton, J. Overview of commercial small satellite systems in the "New Space" age. In: *Handbook of Small Satellites*. Pelton J. Ed. Springer, Cham, **2019**: 1–18.

[14] Flohrer, T., Krag, H., Merz, K., Lemmens, S. CREAM-ESA's proposal for collision risk estimation and automated mitigation. In: Proceedings of the Advanced Maui Optical and Space Surveillance Technologies Conference, **2019**.

[15] Merz, K., Bastida Virgili, B., Braun, V., Flohrer, T., Funke, Q., Krag, H., Lemmens, S., Siminski, J. Current collision avoidance service by ESA's Space Debris Office. In: Proceedings of the 7th European Conference on Space Debris, **2017**.

[16] Braun, V., Flohrer, T., Krag, H., Merz, K., Lemmens, S., Bastida Virgili, B., Funke, Q. Operational support to collision avoidance activities by ESA's space debris office. *CEAS Space Journal*, **2016**, 8(3): 177–189.

[17] Alfriend, K. T., Akella, M. R., Frisbee, J., Foster, J. L., Lee, D. J., Wilkins, M. Probability of collision error analysis. *Space Debris*, **1999**, 1(1): 21–35.

[18] Hyndman, R. J. A brief history of forecasting competitions. *International Journal of Forecasting*, **2020**, 36(1): 7–14.

[19] Kisantal, M., Sharma, S., Park, T. H., Izzo, D., Märtens, M., D'Amico, S. Satellite pose estimation challenge: Dataset, competition design, and results. *IEEE Transactions on Aerospace and Electronic Systems*, **2020**, 56(5): 4083–4098.

[20] Merz, K., Virgili, B. B., Braun, V. Risk reduction and collision risk thresholds for missions operated at ESA. In: Proceedings of the 27th International Symposium on Space Flight Dynamics, **2019**.

[21] Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., Bowman, S. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In: Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, **2018**: 353–355.

[22] Hyndman, R. J., Athanasopoulos, G. *Forecasting: Principles and Practice*, 2nd edn. OTexts, **2018**.

[23] Christ, M., Braun, N., Neuffer, J. tsfresh a python package. Available at https://tsfresh.readthedocs.io.

[24] Olson, R. S., Bartley, N., Urbanowicz, R. J., Moore, J. H. Evaluation of a tree-based pipeline optimization tool for automating data science. In: Proceedings of the Genetic and Evolutionary Computation Conference, **2016**: 485–492.

[25] Wang, C., Bäck, T., Hoos, H. H., Baratchi, M., Limmer, S., Olhofer, M. Automated machine learning for short-term electric load forecasting. In: Proceedings of the 2019 IEEE Symposium Series on Computational Intelligence, **2019**: 314–321.

[26] Mueller, J., Thyagarajan, A. Siamese recurrent architectures for learning sentence similarity. In: Proceedings of the 30th AAAI conference on artificial intelligence, **2016**.

[27] Kingma, D. P., Ba, L. J. Adam: A method for stochastic optimization. *arXiv preprint*, **2014**: arXiv:1412.6980. https://arxiv.org/abs/1412.6980.

[28] Hastie, T., Tibshirani, R., Friedman, J. Boosting and additive trees. In: *The Elements of Statistical Learning*, 2nd edn. New York: Springer New York, **2008**: 337–387.

[29] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., Liu, T.-Y. Light-GBM: A highly efficient gradient boosting decision tree. In: Proceedings

of the 31st Annual Conference on Neural Information Processing Systems, **2017**: 3146–3154.

[30] Kuhn, M. Johnson, K. *Applied Predictive Modeling*. New York: Springer-Verlag New York, **2013**.

Thomas Uriot graduated from the University of Oxford in the UK, where he obtained his master degree in statistics and mathematics. Thomas worked as a researcher at the ESA in the Advanced Concepts Team, where he conducted research on evolutionary machine learning and spacecraft collision avoidance. E-mail: uriot.thomas@gmail.com.

Dario Izzo graduated as a doctor of aeronautical engineering from the University Sapienza of Rome (Italy). He then took a second master degree in satellite platforms at the University of Cranfield in the UK and completed his Ph.D. degree in mathematical modelling at the University Sapienza of Rome where he lectured classical mechanics and space flight mechanics. Dario Izzo later joined the European Space Agency (ESA) and became the scientific coordinator of its Advanced Concepts Team. He devised and managed the Global Trajectory Optimization Competitions, the ESA's Summer of Code in Space, and the Kelvins innovation and competition platform for space problems. He published more than 180 papers in international journals and conferences making key contributions to the understanding of flight mechanics and spacecraft control and pioneering techniques based on evolutionary and machine learning approaches. Dario Izzo received the Humies Gold Medal and led the team winning the 8th edition of the Global Trajectory Optimization Competition. E-mail: Dario.izzo@esa.int.