

IMPLEMENTATION AND COMPARISON OF DATA-BASED METHODS FOR COLLISION AVOIDANCE IN SATELLITE OPERATIONS

S. Metz⁽¹⁾, F. Letizia⁽²⁾, and H. Simon⁽³⁾

⁽¹⁾*Institute for Flight Systems and Automatic Control, Technische Universität Darmstadt, Otto-Berndt-Straße 2, 64287 Darmstadt, Germany, Email: metz.sascha@gmx.de*

⁽²⁾*IMS Space Consultancy for the European Space Operations Center (ESOC)-ESA, Robert-Bosch-Straße 5, 64293 Darmstadt, Germany, Email: francesca.letizia@esa.int*

⁽³⁾*Institute for Flight Systems and Automatic Control, Technische Universität Darmstadt, Otto-Berndt-Straße 2, 64287 Darmstadt, Germany, Email: simon@fsr.tu-darmstadt.de*

ABSTRACT

In an increasingly dense space environment, collision avoidance has become an essential task in satellite operations. With the trend going towards large constellations, automated procedures may relieve the burden on human operators. This work presents the application of Machine Learning to collision risk assessment in satellite operations. In particular, Random Forest decision tree ensembles and recurrent neural network architectures have been deployed to predict the position uncertainties (i.e. the positional covariances) from the orbit determination of Space Debris objects at the time of a close approach, as reported in Conjunction Data Messages (CDMs), and retrieve a probability of collision from the predictions. An extensive feature importance analysis investigates how given parameters have influenced the predictions. Via sampling from covariance prediction distributions, collision probability intervals can be provided.

1. INTRODUCTION

1.1. The current Space Debris environment in the context of collision avoidance

Since the launch of Sputnik-1 has started the space age in 1957, about 6020 rocket launches have been carried out, sending around 10680 satellites into Earth orbit. Around 6250 of these satellites are still on orbit, with about 3900 active and operating [1]. With each launch activity, not only the intended payload, but multiple objects are being released into space, such as spent rocket stages, launch adapters, and fairings. Fragmentations of these objects due to on-orbit explosions and collisions lead to further release of orbiting objects. As a result, space, and especially the Low Earth Orbit (LEO) region ($h \leq 2.000$

km), becomes more and more dense with objects, most of which are Space Debris. Currently, approx. 34000 objects larger than 10 cm (size of an apple) are estimated to be in orbit, with around 28210 of these objects regularly tracked [1]. In the lower size regime, statistical estimations assume approx. 1 million objects of sizes between 1mm and 1cm, and 130 million objects smaller than 1mm to be currently in Earth orbit [1]. Figure 1 shows the count evolution of objects in space since the launch of Sputnik until today [2].

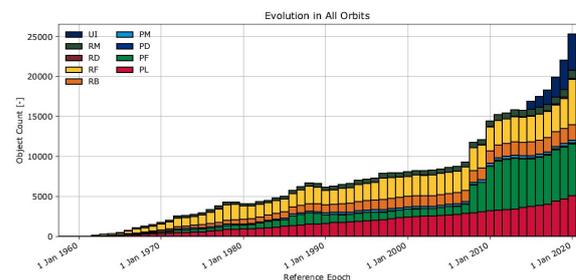


Figure 1. Count evolution by object type [2]

A large share of the current space debris population is due to fragmentations, such as explosions or collisions. The cloud of fragments resulting from a collision eventually contaminates entire orbital shells. In order to protect spacecrafts in this dangerous environment, ESA's Space Debris Office supports operational collision avoidance mainly for the ESA missions Aeolus, Cryosat-2, the constellations SWARM-A/B/C, CLUSTER-II, and the Copernicus Sentinel fleet, as well as spacecrafts of partner agencies [3]. The space debris objects currently tracked by the 18th SPCS via their Space Surveillance Network lead to hundreds of collision warnings per week for a typical spacecraft, and in the case of ESA missions to on average 2 collision avoidance manoeuvres per satellite per year. However, the assessment of conjunction events is currently mostly based on the expertise and experience of human analysts.

1.2. AI for collision avoidance in satellite operations

With the trend going towards large satellite constellations of thousands of satellites in LEO, the assessment of hundreds of collision warnings for only one single satellite within a constellation might become unfeasible in the future [6]. The main challenge in analyzing conjunction events, where human analysts need a good understanding and experience from past conjunction events, is to assess when a next observation update will arrive, and what information it will contain. The ability to estimate how a conjunction event unfolds in the future is necessary because corresponding collision avoidance manoeuvres have to be planned and designed ahead of time, with the process being initiated typically around 2 days before a predicted close approach. Machine Learning methods might be suitable to learn underlying patterns in the evolution of past conjunction events and assist in predicting ahead of time, how a given event unfolds. The lack of publicly shared data has been restraining these efforts, until ESA’s Space Debris Office has published an anonymised dataset of their collected collision warnings [16], received in support of real mission operations, for use in a Machine Learning Challenge (the *Kelvins Collision Avoidance Challenge* [4] [15]), where the main goal was to predict the final probability of collision (PoC, computed via the Alfriend-Akella algorithm [5] and in context of the challenge labeled “collision risk”) of an event, by learning from the time series evolution until 2 days before a predicted close approach.

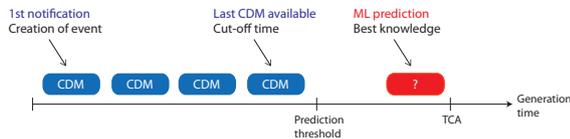


Figure 2. The concept of AI applied to collision avoidance: select a cut-off time (i.e. prediction threshold), collect information about the event and predict the final information.

1.3. The objective of this work

The objective of this work is to explore alternative scenarios to the Kelvins Collision Avoidance challenge, where an additional task was to reliably capture high risk (HR) events ($\text{PoC} > 10^{-6}$), which occur only very rarely in the dataset (about 500 HR events out of around 15.000 total events, which adds up to about 3% of the data). In collision avoidance, a *probability* of collision is introduced because the determined positions of objects are bound to uncertainties. The idea behind this work is to predict not the PoC, but the position uncertainties, also known as the positional covariances, which are contained

in collision warning data and (next to the object’s size and the relative distance) a factor in computing the PoC [7]. This approach presents mainly two advantages: we are not solely dependant on high risk events, but being able to exploit the whole dataset, and since the uncertainties partly come from physical conditions of the given event, we might even be able to take advantage of these relationships and a direct influence of some parameters (e.g. the orbital altitude), on the covariance evolution. ESA’s active satellites (in collision avoidance terminology: the “targets”) are assumed to be very well tracked and known by the owner/operator (O/O), on the other hand, debris objects (the “chasers”), may be more difficult to observe and characterise. Therefore, the focus of this work is to predict the three position uncertainties of the chaser objects in RTN-coordinates: σ_R , σ_T , σ_N . To compare the performance of different methods, Random Forest decision tree ensembles [10] and recurrent neural networks (in particular Long Short-Term Memory (LSTM) cells [9]) will be applied in different setups to predict the target variable. To obtain collision risk estimates as used in operations, the covariance predictions will later be input into the Alfriend-Akella algorithm to compute the PoC.

2. DATA PREPARATION

2.1. The data

The dataset used in this work is the same dataset of collected collision warning messages (conjunction data messages or short “CDM” [8]), which has been published for the Kelvins Collision Avoidance challenge [16], only it is not anonymised. The data consists of around 4 years of CDMs received by ESA’s Space Debris Office in support of real mission operations, containing around 221.767 single CDMs and 18.702 conjunction events, which make for an average of around 12 CDMs per event. The CDMs are grouped by a unique event ID and one event usually represents a time series of one week leading up to the predicted time of closest approach (TCA). To provide more information for Machine Learning algorithms to potentially benefit from, the data has been augmented with additional parameters such as some of the keplerian elements (semi-major axis a , eccentricity e , inclination i) or physical perturbations (e.g. the solar flux index F10), resulting in a total of 163 parameters. Of this number of parameters, actual mandatory CDM data is only information related to the object’s orbit determination and catalogue specifics, as well as the state vector and covariance matrices. As an additional augmentation step, the covariance matrices have been converted to the correlation matrices and standard deviations σ_R , σ_T and σ_N , which have been used as the actual prediction targets.

The dataset has been prepared and cleaned for use of Machine Learning algorithms. Events consisting of one single CDM have been removed. For parameters showing significant outliers, the data outside the 5%-95%-percentile regime has been removed, as well as physically meaningless values. Furthermore, intra-constellation conjunction events (SWARM and CLUSTER-II) have been removed. Since the influence of a manoeuvre on the time series evolution can not be taken into account, pre-manoevrue CDMs have been removed. Repeated conjunction events and CDMs containing only target updates have been kept, the latter especially to study their potential influence in making predictions.

Table 1. The CDM dataset of conjunction events.

Characteristics	Number
Events	18.702
CDMs (rows/samples)	221.767
Parameters (columns/features)	163
Average CDMs per event	12
Maximum CDMs of single event	24
Minimum CDMs of single event	1
% containing chaser updates	49%

2.2. Feature Engineering

As previously mentioned, the main challenge of risk assessment is that the recipient of a CDM issued by the 18th SPCS has limited insight into the process, particularly how the measurement uncertainties are introduced within the US Space Surveillance Network, how they are propagated, or even when the next observation update will be received at all. The irregular observability of space debris objects leads to very dynamic time series evolutions of conjunction events, which shows the evolution of the along-track position uncertainty of a chaser object over time to TCA from right to left. Since CDMs are automatically issued at regular intervals (i.e. every 8 hours), often times CDMs contain only new target data from an observed active satellite, leading to almost constant chaser values, introducing "plateaus" in the evolution, followed by sudden jumps or drops when a new chaser update is eventually being received. To account for this dynamic behavior and make it potentially interpretable by the Machine Learning algorithm, additional features have been introduced, such as a simple count of actual observation updates, the time since the last CDM has been received, or since the last CDM has been received actually containing a new observation update. Regarding continuous parameters contained in the dataset (such as the position uncertainties, but also other parameters such as e.g. the energy dissipation rate SEDR), statistical features have been introduced such as e.g.

the minimum, maximum, or mean value recorded until 2 days before TCA, and other statistical aggregations over the more recent time series evolution, such as e.g. an exponentially weighted moving average (EWMA). "Jumps" have been identified and accounted for by flagging deviations larger than the 2σ -area of the mean of the recent values.

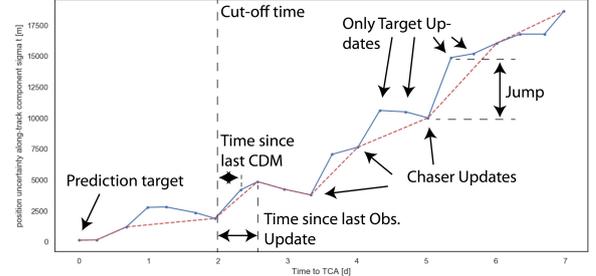


Figure 3. Along-track position uncertainty time series evolution of an example event with visualization of CDM/observation updates and features. Learning from the evolution until a dedicated cut-off time at 2 days before TCA, the final value of the time series is to be predicted.

2.3. Data Interpretation

Since dealing with time series data, in terms of data preparation and shaping different setups have been developed and deployed. For the "event aggregation setup" (developed for use of decision trees, Fig. 4), each event consisting of several CDMs (e.g. several rows in the data) has been condensed into one row containing characteristic information about the entire event until the prediction threshold at 2 days before TCA, such as the minimum, maximum, or last recorded value of a parameter, but also the total number of CDMs or actual observations. For use of recurrent neural networks, especially containing Long Short-Term Memory (LSTM) cells, the data has been partitioned into time steps of equal length (Fig. 5). For this "time step setup", the data has been aggregated just as the event aggregation setup, but at dedicated time steps at 6, 4 and 2 days before TCA, in order to provide more detailed information to the algorithm.

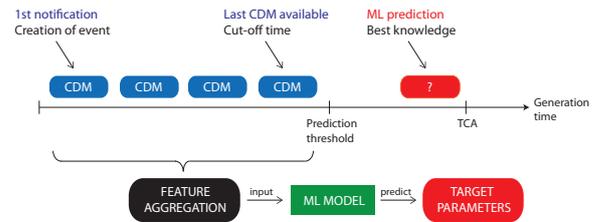


Figure 4. Event aggregation setup.

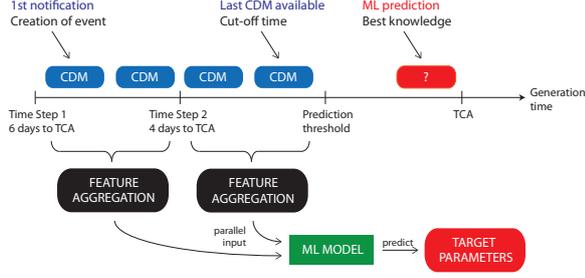


Figure 5. Time series setup.

3. RESULTS OF COVARIANCE PREDICTION USING MACHINE LEARNING

This section presents the results of predicting the positional standard deviations of the chaser objects in RTN-coordinates, using decision tree ensembles and recurrent neural networks. Various methods and setup variations have been applied in an iterative process. The results of predicting the along-track position uncertainty σ_t will be presented in the following as an example, detailed results can be found in [17].

3.1. Decision Tree results

Figure 6 shows the results applying scikit-learns *RandomForestRegressor* [11] on the event aggregation setup applying a 75/25 split between training and test data, and all features in the input feature space. The decision tree ensemble consists of 100 non-regularized decision trees with $min_samples_split = 2$ and $min_samples_leaf = 1$. Significant overpredictions occur in the regime of relatively low true final covariance values ($\sigma_{t,final} < 20.000m / \sigma_{t,pred} > 20.000m$).

3.2. Neural Network results

Figure 7 shows the results of a neural network containing one single LSTM layer, implemented in Keras (which builds on top of TensorFlow [12] with a ReLu (Rectified Linear Unit) activation function and the adam optimizer. The time step setup aggregating the data at 6, 4, and 2 days before TCA has been used and only the target parameter (in this case the along-track position uncertainty σ_t) and its statistics as input features. Although the results stick more closely to the dashed regression line, again significant overpredictions are visible in the regime of low true final covariance values.

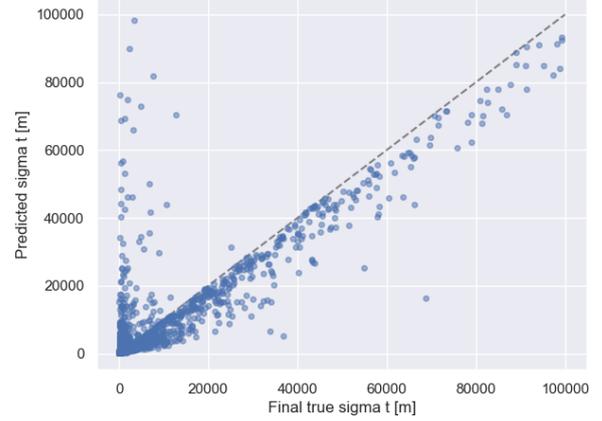


Figure 6. Prediction results using a Random Forest on 25% of the data. $RMSE = 6567,20m$. $R^2 = 0,804$.

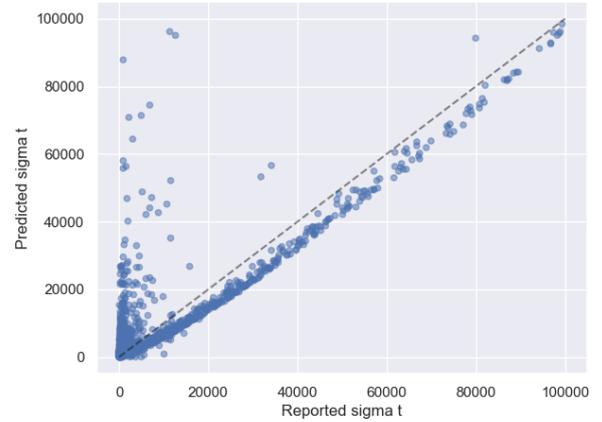


Figure 7. Prediction results using an LSTM on 25% of the data. $RMSE = 6560,07m$. $R^2 = 0,787$.

3.3. Error analysis

A short error analysis reveals the reason behind the overpredictions being made by both decision tree ensembles and neural networks. Figure 8 shows the along-track position uncertainty σ_t time series evolution of events that couldn't be captured by neither method. All misspredicted events share the same behavior, which is one initial chaser observation (starting off the conjunction event), followed by a longer period of CDMs containing only target updates, leading to plateaus in the chaser evolution. Eventually a new chaser observation update leading to a significant drop after the prediction threshold could then not be captured by the models. Since this type of event occurs too rarely in the dataset, it could not be accurately predicted. One solution to this issue is more data or a minimum number of observations has to be available to make predictions.

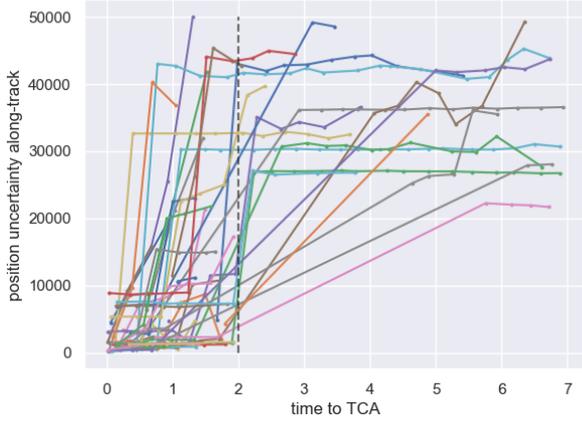


Figure 8. Along-track position uncertainty σ_t time series evolution of events overpredicted by both decision trees and neural networks.

3.4. SHAP Feature Importance analysis

A comparison of Fig. 6 and Fig. 7 and the regression metrics shows that the Random Forest (where all features have been used as input) and the LSTM (where only the target parameter has been used as input) show very similar performance. A feature importance analysis using the SHAP (SHapley Additive ExPlanations) library [13] [14] reveals a potential reason. SHAP values visualize the influence of single input feature data points on the corresponding prediction (e.g. the orbital altitude of a particular conjunction event and its effect on the predicted covariance), with a higher SHAP value on the x-axis resembling a high predicted target value for this particular event, and vice versa. A red marker resembles a high input feature value leading to the prediction, a blue marker visualizes a low input feature value. The features are ranked by global importance.

The plots in Fig. 9 are derived from the Random Forest predictions. On the top plot, we see that if the target parameter (in this case the along-track position uncertainty σ_t) has been available in the input feature space, it has almost entirely been used by the algorithm in making predictions, and no other parameter played a significant role. On the bottom plot, the feature importance values are shown for a run without the target parameter in the input feature space. Although generally decreasing prediction performance, the algorithm is seemingly making use of other features in a more balanced fashion, and interestingly the augmented features "time since last CDM" and the number of observations have been ranked as most important.

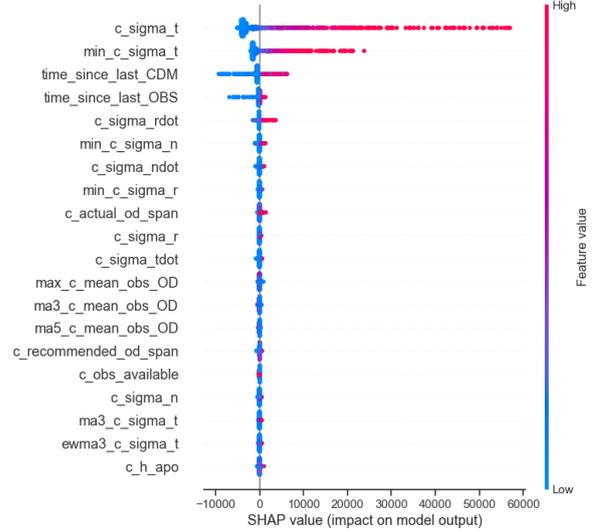


Figure 9. SHAP Local Feature Importance plot for Random Forest predictions of the final along-track position uncertainty σ_t , with the target parameter in the input feature space.

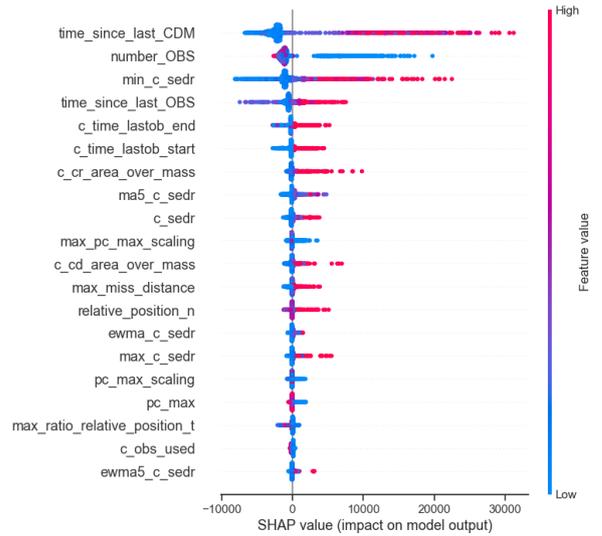


Figure 10. SHAP Local Feature Importance plot for Random Forest predictions of the final along-track position uncertainty σ_t , without the target parameter in the input feature space.

3.5. SHAP Feature Dependence

Using SHAP dependence scatter plots, the influence of single input feature values on a corresponding prediction can be visualized (in Fig. 9, the marker color only reflected a scale from "low" to "high"). Again the SHAP value itself represents the scale of the predicted target value on the y-axis, while the input feature can be found on the x-axis, and in this case the marker is colored by the overall most important

feature. Fig. 11 shows the dependance plot for the feature "number of observations", and Fig. 12 for the feature "time since last CDM".

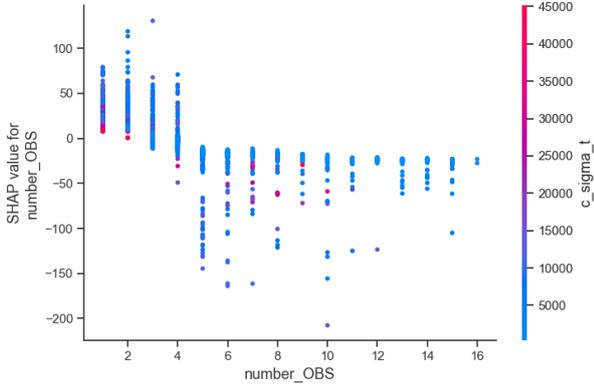


Figure 11. SHAP dependance plot for Random Forest predictions of the final along-track position uncertainty σ_t and the input feature "number of observations".

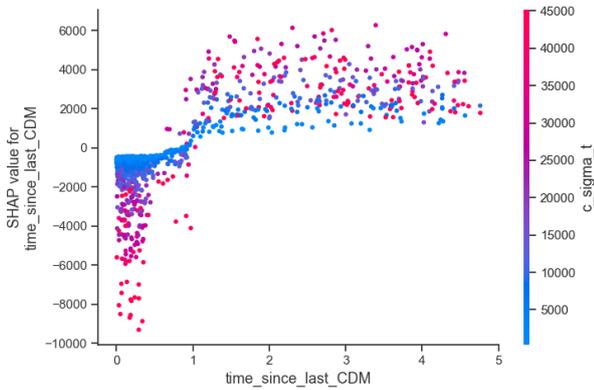


Figure 12. SHAP dependance plot for Random Forest predictions of the final along-track position uncertainty σ_t and the input feature "time since last observation".

4. RESULTS OF COLLISION PROBABILITY COMPUTATION

4.1. PoC computation using Random Forest covariance predictions

Since it is used in ESA operations and to compare with the results of the Kelvins Collision Avoidance challenge, the probability of collision (PoC) has been computed via the Alfriend-Akella algorithm, using the covariance Machine Learning predictions of the Random Forest. The Alfriend-Akella algorithm takes as input the state vector of both objects as well as the positional components of the covariance matrix,

the relative distance (i.e. the "miss distance"), and the object size ("Hard Body Radius" (HBR)) [5] [7].

The setup was implemented as follows: assume the target to be known (e.g. through O/O ephemeris) and thus use the final values available in the dataset. The chaser however is assumed to be entirely not known at the time of making predictions, hence the last recorded value available at prediction time (i.e. a "naive prediction") has been used for the chaser object's position and correlation factors (to complete the covariance matrix), and finally the standard deviation predictions being made using Random Forest predictions. Table 2 gives an overview of the obtained performance metrics.

Table 2. Risk computation evaluation

Risk regression metrics (ALL HR)	
RMSE	6.038 3.169
RMAE	1.588 1.081

High risk confusion matrix		
pred. →	0	1
↓ true		
0	2421 (TN)	35 (FP)
1	23 (FN)	72 (TP)

High risk classification metrics	
precision	0.673
recall	0.758
F1	0.713
F2	0.739

Fig. 13 shows the true final values of the PoC (on the y-axis) over the values computed from the predictions (on the x-axis), Fig. 13 shows a cut-out of the true final high risk population (see y-axis in bottom plot of figure 13 and the corresponding values computed from the predictions).

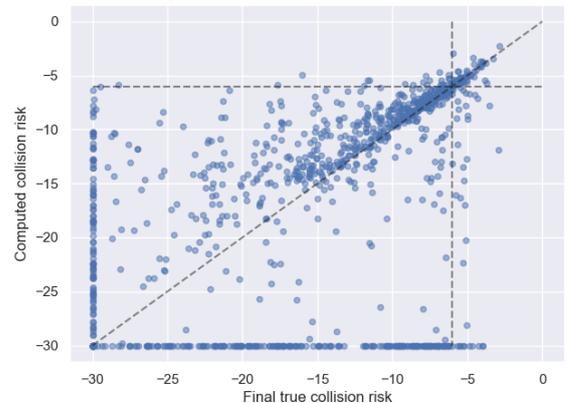


Figure 13. All test set events.

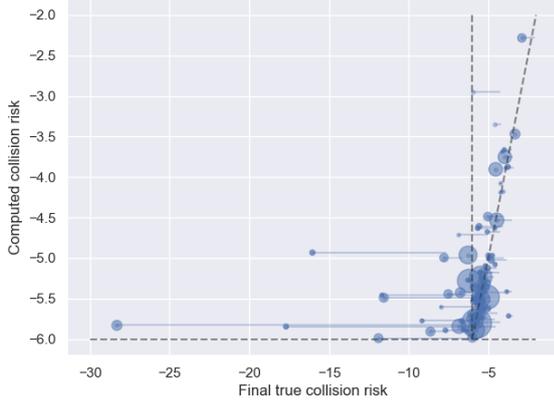


Figure 14. True high risk events. Markers scaled with absolute σ_T -prediction error.

4.2. PoC intervals via covariance prediction distribution sampling

Since a prediction made by a Random Forest algorithm is essentially the average of the training target value distribution of an end leaf, it is also possible to derive other statistical quantities of this distribution, such as e.g. the 5th and 95th percentile of the prediction distribution. By sampling from covariance prediction distributions, corresponding PoC distributions could be derived in order to provide a range of potential event outcomes. Fig. 15 shows a sample distribution of the along-track position uncertainty σ_t (top) and the corresponding PoC distribution (bottom) for an example HR-event (to compute the PoC, sample distributions of σ_r and σ_n were also created).

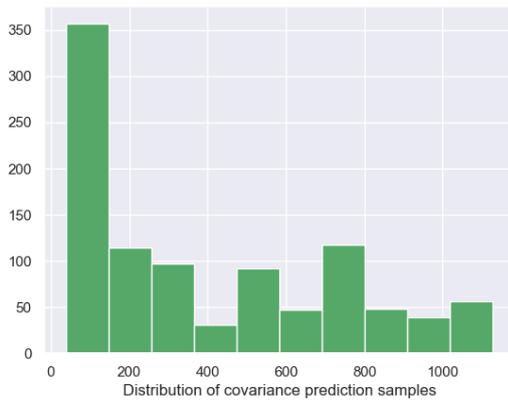


Figure 15. Sample distribution of along-track position uncertainty σ_t for an example HR-event.

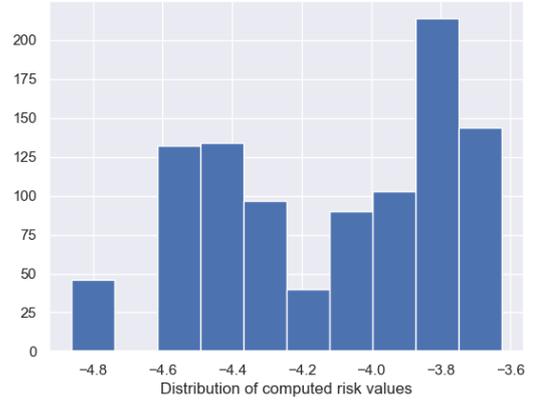


Figure 16. Corresponding PoC distribution of the example HR-event.

Figure 17 shows the true high risk event population of the test dataset (e.g. approx. 100 samples, see x-axis) and the corresponding values computed from the predictions. The pink shaded area visualizes the 5th to 95th percentile regime of the corresponding PoC distribution, and the violet shaded area the 25th to 75th percentile regime. True target values are shown with red markers, values computed from the predictions (using the "standard approach", e.g. the mean value of RF prediction distributions as before) are visualized with blue markers. It can be seen that for approx. 25% of test events the computed PoC range can be determined, however for the majority of events the lower limit (i.e. the 5th percentile) stretches out to the lowest PoC saturation at -30. However, the proposed method shows potential to deliver collision risk ranges from covariance decision tree ensemble predictions.

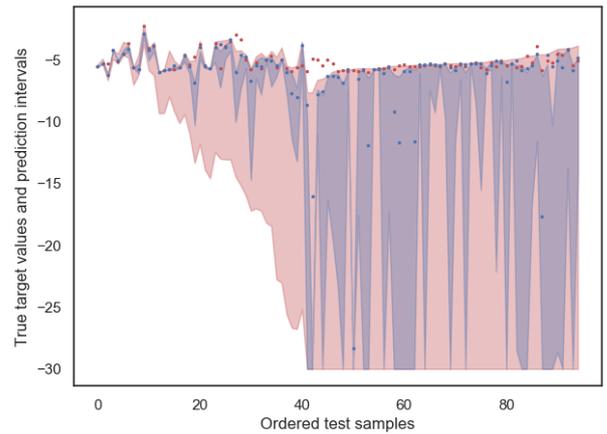


Figure 17. True high risk event population of the test dataset and corresponding risk prediction intervals (Pink: 5-95%, violet: 25%-75%). Red markers: true target values. Blue markers: PoC values computed from standard RF prediction approach.

5. CONCLUSION

This work presented a first approach in analyzing a real-world dataset of operational CDMs, and a Machine Learning approach to predicting orbital position uncertainties of space objects and derive collision probability estimations. Several data augmentation and interpretation strategies have been developed and implemented, leading to a workflow from raw CDM data to collision risk prediction intervals. The preliminary comparison has shown that recurrent neural networks using only the recent time series evolution of the target value already outperform decision tree ensembles using additional data. Events with chaser updates after the prediction threshold and a long period of only target updates could not be captured by any method used in this work, which leads to the assumption that a minimum amount of chaser observations is needed to make reliable predictions. The application of Machine Learning to collision risk estimation is a promising approach that can be further intensified with larger amounts of data to build reliable models on.

ACKNOWLEDGEMENTS

This work has been enabled by the Space Debris Office of ESA/ESOC, especially by the help of Dr. Francesca Letizia, and the Institute for Flight Systems and Automatic Control (FSR) of Technische Universität Darmstadt under supervision of Prof. Dr.-Ing. Reinhold Bertrand. The authors would like to thank in particular Dr. Jan Siminski, Dr. Silvia Sanvido and José Antonio Martínez Heras for sharing their technical expertise and dedicating their time and knowledge to a research activity of great relevance to future Space Traffic Management and Space Situational Awareness.

REFERENCES

1. The European Space Agency (2021). Space Debris by the numbers. Online at https://www.esa.int/Safety_Security/Space_Debris/Space_debris_by_the_numbers (as of 9 April 2021).
2. ESA Space Debris Office (2020). Space Environmental History in Numbers. In *ESA's Annual Space Environment Report*.
3. Merz, K., Bastida Virgili, B., Braun, V., Flohrer, T., Funke, Q., Krag, H. Lemmens, S., and Siminski, J. (2017). Current Collision Avoidance service by ESA's Space Debris Office. In *Proceedings of the 7th European Conference on Space Debris*.
4. ESA's Advanced Concepts Team (ACT) in partnership with ESA's Space Debris Office (2019). Kelvins Collision Avoidance Challenge. Online at <https://kelvins.esa.int/collision-avoidance-challenge/> (as of 9 April 2021).
5. Akella, M. R., Alfriend, K.T. (2000). Probability of Collision Between Space Objects. *Journal of Guidance, Control, and Dynamics*.
6. The European Space Agency (2019). AI challenged to stave off collisions in space. Online at https://www.esa.int/Enabling_Support/Space_Engineering_Technology/AI_challenged_to_stave_off_collisions_in_space (as of 9 April 2021).
7. Klinkrad, H. (2006). Operational Collision Avoidance with Regard to Catalog Objects. In *Space Debris - Models and Risk Analysis*.
8. The Consultative Committee for Space Data Systems (2013). Recommendation for Space Data System Standards. *Conjunction Data Message Recommended Standard CCSDS 508.0-B-1 Blue Book*.
9. Hochreiter, S., Schmidhuber, J. (1997). Long Short-Term Memory. In *Neural Computation 9(8): 1735-1780, 1997*.
10. Breiman, L. (2001). Random Forests. *Machine Learning 45, 5-32*. <https://doi.org/10.1023/A:1010933404324>
11. Pedregosa, F., et al (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research 12, pp. 2825-2830*. <https://scikit-learn.org/stable/index.html>
12. Chollet, F., and others (2015). *Keras*. <https://keras.io>
13. Lundberg, S. M., Lee, S.-I. (2017). A Unified Approach to Interpreting Model Predictions. *Advances in Neural Information Processing Systems 30*. Ed. by I. Guyon et al. Curran Associates, Inc, pp. 4765-4774.
14. Lundberg, S. M. et al (2020). From local explanations to global understanding with explainable AI for trees. *Nature Machine Intelligence 2.1, pp. 2522-5839*.
15. Uriot, T., Izzo, D., Simões, L. F., Abay, R., Einecke, N., Rebhan, S., Martínez-Heras, J., Letizia, F., Siminski, J., Merz, K. (2020). Spacecraft Collision Avoidance Challenge: design and results of a machine learning competition. <https://arxiv.org/abs/2008.03069>
16. Uriot, T., Izzo, D., Martínez-Heras, J., Letizia, F., Siminski, J., Merz, K. (2021). Collision Avoidance Challenge dataset [Data set]. Zenodo. <http://doi.org/10.5281/zenodo.4463683>
17. Metz, S. (2020). Master Thesis: Implementation and comparison of data-based methods for collision avoidance in satellite operations. <https://doi.org/10.13140/RG.2.2.35021.26089>.