

## Should I stay or should I go? Machine Learning applied to Conjunction Analysis

**Cristina Pérez Hernández\***, **Daniel Lubián-Arenillas<sup>b</sup>**, **Carlos Paulete Periañez<sup>c</sup>**, **Jesús Tirado Velez<sup>d</sup>**  
**Alexandru Solomon<sup>e</sup>**

<sup>a</sup> *CDTI (Centre for the Development of Industrial Technology), Calle del Cid 4, 28001 Madrid, Spain, cristina.perez@cdti.es*

<sup>b</sup> *Elecnor Deimos, Ronda de Poniente 19, 28760 Tres Cantos, Spain, daniel.lubian@deimos-space.com*

<sup>c</sup> *GMV Aerospace & Defence SAU, Calle Isaac Newton 11, 28760 Tres Cantos, Spain, cpauleteperianez@gmv.com*

<sup>d</sup> *GMV Aerospace & Defence SAU, Calle Isaac Newton 11, 28760 Tres Cantos, Spain, jtirado@gmv.com*

<sup>e</sup> *GMV Aerospace & Defence SAU, Etaj 32, Calea Floreasca 246C, București 077190, Romania, alexandru.solomon@gmv.com*

### Abstract

The number of orbiting objects is increasing at an ever-growing pace. Inevitably, this is also driving up the number of conjunction events, especially in LEO. This is forcing the Space Situational Awareness (SSA) actors to expand the proportion of automated tasks within their conjunction detection and analysis pipelines. During SSA providers' nominal operations possible conjunctions are identified, tracked, assessed and, if necessary, a manoeuvre recommendation plan is created and iterated with the satellite operator along the event duration. The increasing workload of the conjunction analysis service and the growing number of satellite operators demanding this service, yields to the need to explore new techniques like Machine Learning (ML) to decrease the time needed during the tracking phase of a conjunction event, by anticipating its evolution. Thanks to the fact the S3TOC (Spanish SST Operations Centre) is one of the two operational centres in charge of the Collision Avoidance within the EU SST, working in a hot redundancy schema, a large number of detailed datasets has been generated over time with all the information along the conjunction events evolution being available. This has allowed to start research on ML algorithms and techniques considering the big quantity of data required. The work presented in this paper is focused into two main tasks: First, the determination of the probability of escalation or de-escalation of the risk level, which depends on the Probability of Collision (PoC). Therefore, during the event, attempts are made at anticipating whether it can be disregarded as a low-risk event or if it should be included on a watchlist in case its alert level is to be increased. Second, the prediction of the elements used to compute the PoC. The information of an event is updated along its duration through Conjunction Data Messages (CDMs). The CDMs contain all the information needed to compute the PoC such as the relative position between the objects and their covariances. In this second task, some of the elements in the CDM are predicted into the future, so that a future estimation of the PoC can be computed. This is just the beginning of the long journey of introducing Artificial Intelligence in the SSA field and S3TOC and EU SST are on-board.

### Acronyms/Abbreviations

CA: Conjunction Analysis  
CDM: Conjunction Data Message  
EU SST: European Union Space Surveillance and Tracking  
FN(R): False Negative (Rate)  
FP(R): False Positive (Rate)  
GEO: Geostationary Orbit  
IQR: Inter-Quartile Range  
LEO: Low Earth Orbit  
ML: Machine Learning  
PoC: Probability of Collision  
RL: Risk Level  
ROC curve: Receiver Operating Characteristic curve  
RSO: Resident Space Object  
RTN: Radial-Tangential-Normal  
S3TOC: Spanish Space Surveillance and Tracking Operational Centre

SSA: Space Situational Awareness  
SST: Space Surveillance and Tracking  
TN(R): True Negative (Rate)  
TP(R): True Positive (Rate)

### 1. Introduction

In the frame of the Spanish Space Surveillance and Tracking (SST) activities, and in particular, in regards to the operations an analysis of conjunction events between space objects, the S3TOC is undertaking research activities aimed at optimizing the use of its resources. One of those initiatives is centred on the improvement of efficiency and reduction of human time devoted to conjunction analyses, so it can maintain and improve its collision risk assessment service jointly to the fragmentation and re-entry analysis services as part of the European SST consortium (EUSST).

In an SSA operational environment, particularly with the rapid increase of RSOs in recent times due to the launch of satellite constellations in LEO, close approach events (conjunction events) happen more and more often. When such event is detected, several processes get kicked off to ensure the conjunction will not resolve into a collision. CDMs are published periodically containing information about the event itself that allows for the computation of the PoC. These CDMs contain physical and orbital information about each of the objects involved in the CA event as well as information on the uncertainty of these values.

Due to the workload increase resultant from the larger RSO population, there is an interest to resolve the conjunctions as early as possible without compromising reliability of the process. To that end, this work presents two applications of ML that tackle this problem. The first one provides an algorithm that predicts a future risk level change in the event. This way, it can be disregarded early or more resources can be devoted to it if the risk level is expected to decrease or increase respectively. The second application provides an algorithm that aims at improving the propagation made from the orbit determination performed with data available from SSA sensors to help provide a more reliable measure of the PoC.

ML is a data-driven modelling technique from the field of Artificial Intelligence (AI) that has been booming in the last decade due to the digitization of the society, the availability of a multitude of data in all industrial sectors and the increase of computing capabilities. This, together with the possibility of extracting and recognizing patterns easily thanks to programmatic techniques such as ML, makes it very interesting to import them to a sector such as space, and in particular in the area of Space Situational Awareness (SSA).

The high proliferation of space objects in orbits in recent years and the future prospect, make necessary the application of different mitigation strategies in cooperation with an exhaustive surveillance and tracking services of these objects, as provided by EUSST. Anything that can contribute to the improvement and automatization, and make these services provided more efficient should be definitely encourage. This includes, for example, the usage of ML techniques in some processes of the SST-chain, like sensor tasking and planning.

ML techniques are mainly based on statistics and optimization. Through this, a base algorithm automatically adjusts a multitude of parameters to a previously selected and prepared set of data. This collection and preparation is, perhaps, the most important and difficult step of the process. If the data used is not good enough, the model will not be able to find an underlying pattern and could not be used with new similar examples (or samples) of that data. In other words, it will not be able to ‘learn’ from it. Therefore, it is

important to have a model as good as the dataset, with at least similar performance.

In both cases, the algorithms are based on ML techniques and rely on data of previous events to learn and provide their respective products.

## 2. Prediction of risk level change for a CDM in a CA event

In this section, machine learning techniques are applied to real conjunction analysis events, specifically aiming to assess whether or not an event is significant on the day of analysis. In this case, the machine learning model would estimate whether a given event could change risk level the next day, based on historical data from similar events. This can have the very relevant consequence that, if the risk level is expected to increase, the analysis of the possible collision is started earlier, but it is also interesting if the level of risk is expected to decrease, since then the operators and SST network effort could be devoted to other tasks.

This problem can be modelled as a classification task in supervised learning, where the target labels are one of three options: risk level will increase, risk level will maintain, and risk level will decrease. The risk level is directly related to the probability of collision and is used operationally to classify CDMs in three levels, as shown in Table 2.1.

Table 2.1. Definition of the different risk levels for a conjunction event.

Info	Low risk event that falls inside screening volumes.
Warning	Medium risk event that could lead to further monitoring if close to alert thresholds.
Alert	High risk event that must be closely monitored and that may require a manoeuvre.

Thus, the idea is to compute the probability of a change in the risk level happening in the future of a conjunction event using current available CDMs for that event, and in which direction. Therefore, the model should take a CDM and return the probability that the event will increase, maintain or decrease its risk level.

### 2.1 Dataset

#### 2.1.1 Starting dataset

The dataset that is required to complete this task consists of a set of CDMs grouped by conjunction event. This dataset is available in the premises of S3TOC thanks to the continuous provision of conjunction analysis services to its users since 2016. These CDMs are also labelled using the definition shown before in Table 2.1 for the RL.

The dataset used includes about 2.5 million CDMs, which can be grouped in 160.000 conjunction events (with a median of 8 CDMs per event, see Fig. 2.3). All these CDMs contain at least the mandatory fields described in [1]: identifiers, state vector and covariance matrix for the primary object and the secondary object, miss distance and time of closest approach, among other metadata. Since this data was already processed by the S3TOC CA service, it also included the associated risk level for each CDMs, the overall risk level of the event, the originator of the CDM, orbital regime, collision probability method, manoeuvrability status of both objects, etc.

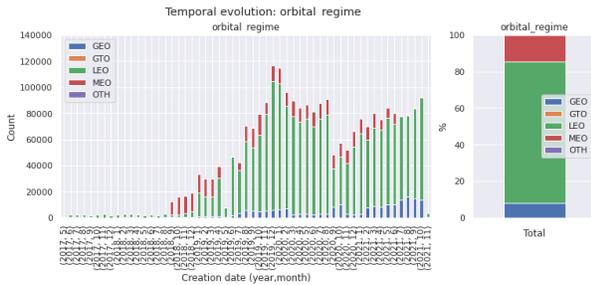


Figure 2.1. Distribution of CDMs per orbital regime over time.

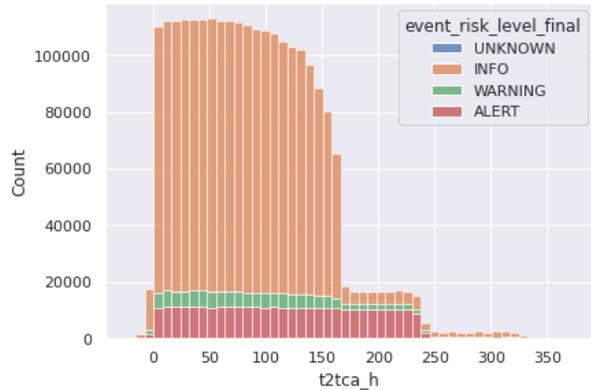


Figure 2.2. Distribution of the time to TCA in hours per event risk level.

Figures 2.1, 2.2 and 2.3 show some key statistics about the dataset. In the first one, one can see that the CDMs are distributed among five orbital regimes. Most of the CDMs are in LEO, but at the beginning of the S3TOC activities MEO dominated. This change is due to filtering CDMs for a pair of collocated MEO that generated too many CDMs when the risk of a conjunction was controlled.

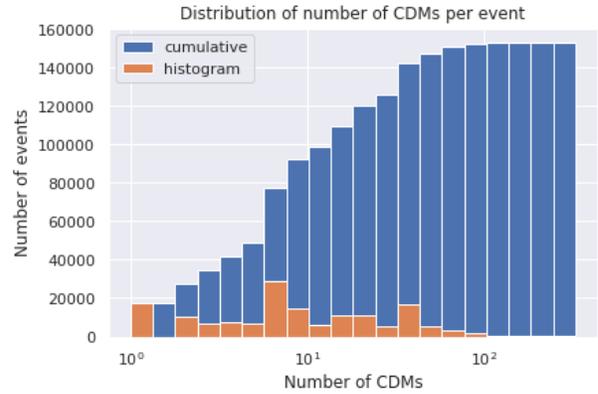


Figure 2.3. Number of CDMs per event.

### 2.1.2 Feature engineering

Feature engineering is the task where new information is created from the data, either combining features extracted from the dataset or using external data. In this case, only information already available in the provided dataset is going to be used to create new features. These features are, therefore, combinations of the provided data and serve the purpose of being more informative or more physically relevant. For example, the time to TCA is a very simple combination of the creation date and the TCA since it is just the difference between the two.

Another example is that the covariance matrices in RTN for the primary (21 features) and secondary (21 features) objects can be reduced to two numbers when they are combined, projected into the B-plane and diagonalised, as is usually done in CA for computing the PoC. These two numbers retain most of the information while reducing the dimensionality of the problem substantially. This projection was done following [2].

Similarly, in the field of orbital mechanics it is well known that the classical orbital elements are more informative than the position and velocity vectors, which are provided in the dataset since those are mandatory fields in a CDM. Semimajor axis, eccentricity and inclination for both primary and secondary are more relevant than the  $x$ ,  $y$  and  $z$  components of the position and the velocity as they define the geometry and orientation of the orbital ellipse.

In Figures 2.4 and 2.5 are represented the eccentricity and inclination versus the semimajor axis for the primary (blue) and secondary (orange) objects. LEO objects are distributed in all ranges of eccentricity and inclinations, whereas GEO are concentrated around zero degrees of inclination. The distribution of the cosine of the angle between the relative position vector in RTN and the relative velocity vector in RTN with origin in the primary object is shown in Fig. 2.6. Since most values are around the value 0, 1 and  $-1$ , it seems most of the CDMs correspond to objects going in or out of the direction of the secondary object.

Another interesting value that can be obtained from the position and velocity of the primary and secondary objects is the angle between both orbit planes (Fig. 2.7), computed as the angle between the two orbital angular momenta. It seems that GEO has small values of this angle, whereas in MEO and LEO the values are distributed between 0 and 180°. Note that the figure on the left is in log scale.

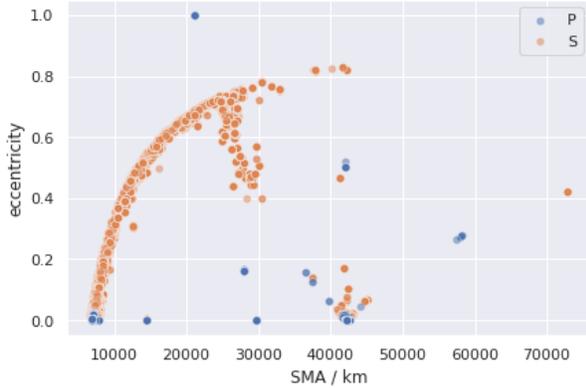


Figure 2.4. Distribution of primary (P) and secondary (S) objects in terms of semimajor axis and eccentricity.

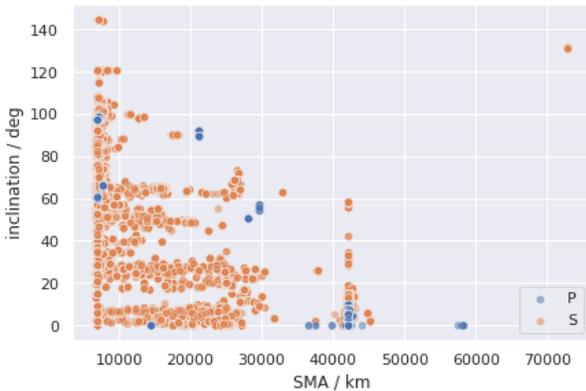


Figure 2.5. Distribution of primary (P) and secondary (S) objects in terms of semimajor axis and inclination.

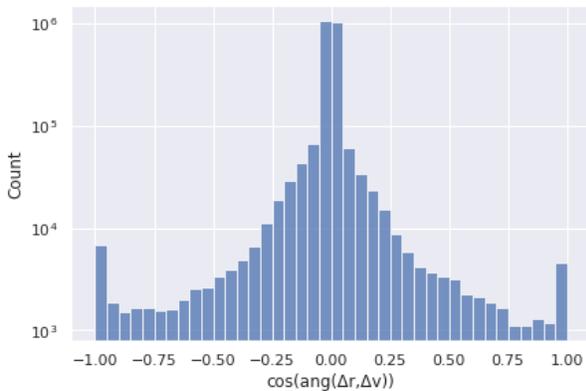


Figure 2.6. Distribution of the cosine of the angle between the relative position and velocity vectors.

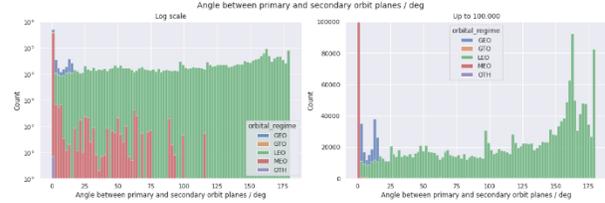


Figure 2.7. Distribution of the angle between the primary and secondary orbit planes per orbital regime.

### 2.1.3 Final dataset

The final dataset consists of a total of 1,820,258 samples with the following 19 features:

- *sma\_p/s*: semimajor axis of the primary/secondary object
- *ecc\_p/s*: eccentricity of the primary/secondary object
- *inc\_p/s*: inclination of the primary/secondary object
- *angle\_between\_p\_s\_planes*: angle between the primary and secondary orbital planes
- *t2tca\_h*: hours between the creation date and the time of closest approach
- *source*: originator of the CDM
- *manoeuvrable\_p/s*: whether the primary/secondary object is manoeuvrable or not
- *area\_pc\_p/s*: Actual area of the primary/secondary object
- *cos\_angle\_relative\_rv*: cosine of the angle between the relative position and velocity
- *relative\_speed*: relative speed
- *miss\_distance*: relative distance
- *cov\_bplane\_eig1*: first eigenvalue of the covariance matrix projected onto the B-plane at TCA
- *cov\_bplane\_eig2*: second eigenvalue of the covariance matrix projected onto the B-plane at TCA
- *autonomous*: whether the CDM is autonomous or not

The label *risk\_level\_future\_change* is distributed (Fig. 2.8) as 1524567 for the value MAINTAIN, 264127 for INCREASE and 31564 for DECREASE. The label was computed considering the difference between the risk level of the last CDM of the event and the evaluated CDM.

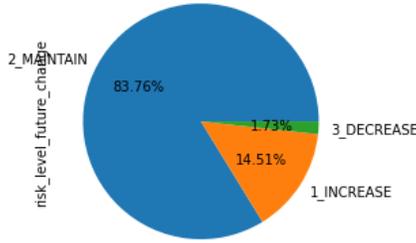


Figure 2.8. Distribution of samples per class.

## 2.2 Methodology

### 2.2.1 Dataset selection

The dataset was split in two parts, the training set and the test set. The training set is used during the fitting phase, whereas the test set is used to evaluate the trained model, since that data would not be used during the training and therefore the goodness of the model to work on unseen data is tested. This split is done considering a date. All data before 1<sup>st</sup> March 2021 is used for training. This gives about 76% of the data for training.

Since the different classes in the target variable are not approximately evenly distributed, a simple upsampling is done to augment the number of samples for the INCREASE and DECREASE categories to match the number of samples in the MAINTAIN class. For a first study and a demonstration of capabilities, this technique seems sufficient, although it could possibly be improved.

### 2.2.2 Dataset transformation

Features need to be scaled to improve the performance of certain algorithms, like the ones using gradient descent as optimizer (like the gradient-boosting family) or the ones based on distance. In this activity, the scaling is done by centring the values around the median and scaled them using the interquartile (IQR), that is, the range between the 1st quartile (25th percentile) and the 3rd quartile (75th percentile).

Other scalers available make use of the mean and the standard deviation, or the minimum and maximum values. These scalers are more sensitive to outliers, whereas the scaling based on quantiles is more robust.

### 2.2.3 Algorithm and hyperparameter selection

This problem as mentioned above, is a multi-label classification problem in the field of supervised learning. Tree-based ensemble learning algorithms have demonstrated to perform well in tabular dataset [3]. When adding their good computational performance, simplicity and ease of use, they appear to be good candidates to solve this problem.

Ensemble learning consists of a set of "weak" learners, which are trained together and their results are considered at the same time to create a "strong" learner. An example of this kind of weak learner is a Decision

Tree, which is a very simple model (a set of if-else rules whose thresholds are computed automatically). Depending on how a set of these Decision Trees is trained (rules of node splitting, voting, etc.), different ensemble learning algorithms arise.

The following algorithms [ 4] from this family are going to be considered:

- Random Forest: using a set of decision trees, each built from a sample drawn with replacement and at each node the best split is found using information of a subset of features. Then, the prediction of all trees is averaged for the result of the ensemble.
- Extremely Randomized Trees: similar to RF, but the splits are selected from the best of a set of random splits, increasing the randomness of the process.
- Gradient Boosting: gradient boosting on decision trees. Similar to random forest, but each decision tree of the ensemble is added and trained in conjunction to the previously trained decision trees, using the remanent errors of the previous fitted estimators.

Each algorithm has its own hyperparameters, that is, certain aspects of the algorithm or the model during the learning process.

They refer to the model selection part in the ML modelling process. For example, the selection of an algorithm or the number of estimators using in an ensemble learning algorithm are hyperparameters of the ML model. Another typical example is the number of layers and the number of weights per layer in a neural network.

Optuna is "an open source hyperparameter optimization framework to automate hyperparameter search", that is to say, a library that can be used to look for the best combination of hyperparameters to the problem for a selected metric. The hyperparameter space to use in the search and the score are defined by the ML engineer. The hyperparameters considered can be found in Table 2.2.

Table 2.2. Hyperparameters considered for tuning.

Hyperparameter	Possible values	Comment
classifier_obj	RandomForestClassifier, ExtraTreesClassifier, LGBMClassifier, XGBClassifier	
n_estimators	[20, 200], logarithmic	
max_depth	[2, 32], logarithmic	Controls the depth of the decision tree.
ccp_alpha	[0, 1], logarithmic	Only applicable to RF and ExtraTrees

learning_rate	[0.01, 1], logarithmic	algorithms. It controls the post-complexity pruning. Only applicable to gradient boosting algorithms.
---------------	------------------------	---

The score used is the macro average recall in the test set, which is also the average of the diagonal of the multiclass confusion matrix once it is normalized by the number of true samples of each class.

#### 2.2.4 Model validation

The confusion matrix, in binary classification, is a table where the predicted values are put against the actual values. This matrix is better the higher the values in the diagonal are:

		Predicted values	
		Negative	Positive
Actual values	Negative	True Negatives	False Positives
	Positive	False Negatives	True Positives

The accuracy is the ratio of correctly classified samples to the total number of samples. The precision measures the ability of the classifier not to label as positive a sample that is negative. It is also known as positive predictive value. The recall or sensitivity is, intuitively, the ability of the classifier to find all the positive samples. It can be viewed as the probability that a successful request is labelled as such. The F1-score is a combination of the previous two.

$$\text{Accuracy} = (TP + TN) / (TP + TN + FP + FN)$$

$$\text{Precision} = TP / (TP + FP)$$

$$\text{Recall} = TP / (TP + FN)$$

$$\text{F1-score} = 2 \cdot \text{precision} \cdot \text{recall} / (\text{precision} + \text{recall})$$

Since these metrics are ratios, they are better the closer they are to 1 because that means that the fraction of FP and/or FN is smaller.

\* <https://scikit-learn.org/>

† <https://lightgbm.readthedocs.io/>

These metrics can be extended to multiclass classification by considering a binary problem for each class, i.e., INCREASE vs NOT INCREASE (DECREASE + MAINTAIN). These combinations can then be averaged to obtain overall performance metrics.

#### 2.2.5 Open-source tools

Well-known and validated solutions in the machine learning industry were used for the development and training of the models. The Python libraries used implement the aforementioned algorithms and provide the necessary framework. The following libraries have been used for the development of this activity: *scikit-learn*<sup>\*</sup>, *optuna*<sup>†</sup>, *lightgbm*<sup>‡</sup> and *XGBoost*<sup>§</sup>.

### 2.3 Results

The Optuna framework found the combination of hyperparameters collected in Table 2.3 that provides the best score in all trials, with an average macro recall in the test set of approximately 0.74. Other hyperparameters of the algorithm are left with the default values included in the framework.

Table 2.3. Final results of the hyperparameter optimization with Optuna.

Hyperparameter	Value
classifier_obj	LGBMClassifier
learning_rate	0.0926520740299368
max_depth	5
n_estimators	55
ccp_alpha	N/A

The selected algorithm is taken from the Light Gradient Boosting Machine library. This algorithm is based on an ensemble of decision trees and implements a variation of the gradient boosting training technique.

The dataset used for the evaluation, as stated in section 2.1, corresponds to 25% of the input data. Only the CDMs after March 1st, 2021 are going to be used for evaluation and validation. The table below compiles a series of metrics for both the train and test set. The similarity in those metrics means that the model is able to learn from the training set and can generalise and extrapolate that behaviour to unseen data samples. In this case, CDMs of new events. The closer these metrics are to 1, the better the model is performing the classification task.

Table 2.4. Classification metrics per class applied to the training and test set.

TRAIN SET				
	precision	recall	F1	# samples
DECREASE	0.78	0.82	0.80	1518681

‡ <https://xgboost.readthedocs.io/>

§ <https://optuna.org/>

MAINTAIN	0.85	0.73	0.79	1520159
INCREASE	0.78	0.85	0.81	1519919
accuracy	0.80			4558759
macro avg.	0.80	0.80	0.80	
weighted avg.	0.80	0.80	0.80	
<b>TEST SET</b>				
	precision	recall	F1	# samples
DECREASE	0.07	0.86	0.13	7425
MAINTAIN	0.91	0.64	0.75	417942
INCREASE	0.57	0.72	0.64	148279
accuracy	0.66			573646
macro avg.	0.52	0.74	0.50	
weighted avg.	0.81	0.66	0.71	

Figures 2.10 and 2.11 present the confusion matrices for both the training set and test set, respectively. Again, the similarity between both confirms that there is no under- or overfitting in the model and is able to generalise to new CDMs. Even though the performance is not perfect and there is room for improvement, these results are very promising for a system of these characteristics. As a summary, in the test set:

- 86% of the predicted risk-level-decreasing CDMs are actually CDMs in events that will decrease their risk level in the future. This number being that high is good for the performance of the model, since the CA operator can trust when the model is saying that the risk level will decrease with confidence
- 64% of the predicted risk-level-maintaining CDMs are actually CDMs in events that will maintain their risk level in the future
- 72% of the predicted risk-level-increasing CDMs are actually CDMs in events that will increase their risk level in the future
- 17% of the CDMs categorised as risk-level-maintaining are actually CDMs that will increase their risk level in the future. These results should be improved, since that would mean that the operator might ignore 17% of the CDMs because of this.

Other ratios shown in the confidence matrix are not worrisome since they are conservative: the demand more attention from the CA operator than it might necessary.

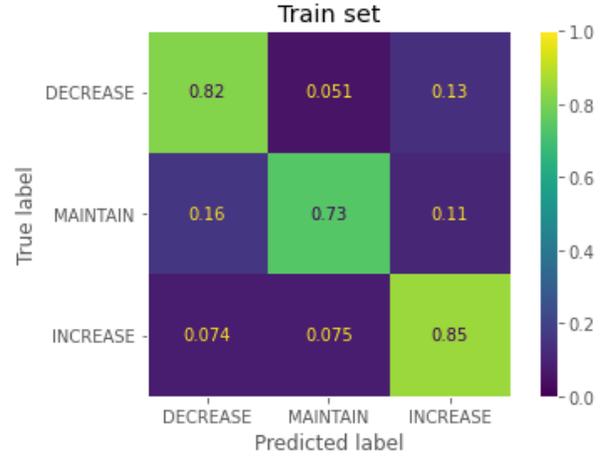


Figure 2.9. Confusion matrix of the final model applied to the training set.

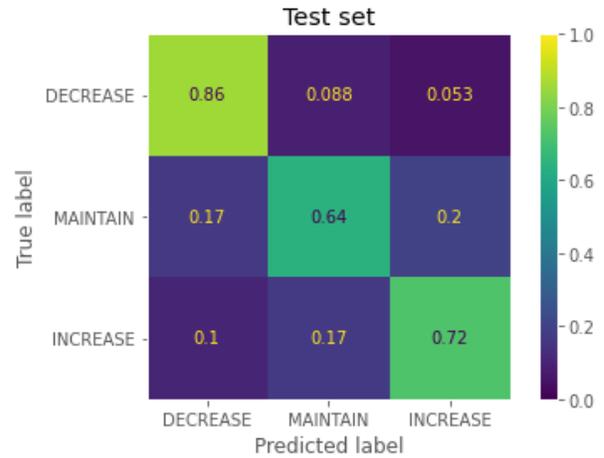


Figure 2.10. Confusion matrix for the final model applied to the test set.

The receiver operating characteristic (ROC) curve (Fig. 2.11) shows the trade-off between sensitivity (or TPR) and specificity ( $1 - \text{FPR}$ ). Classifiers that give curves closer to the top-left corner indicate a better performance. This curve is displayed below. Here is shown that the model can separate better the DECREASE class from the other classes than the other classes. This is in line with what was shown in the confusion matrix.

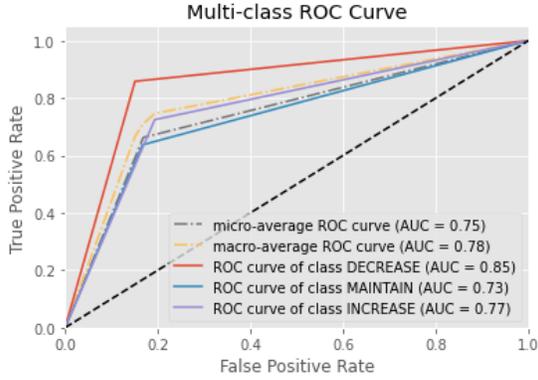


Figure 2.11. Multi-class ROC curve for the test set.

Finally, inspecting the model is an interesting task to do. Explainability in artificial intelligence is difficult: most models behave like black boxes and it has to be done by modifying the inputs to study the behaviour of the model. For example, taking from the test set only the samples of autonomous CDMs and non-autonomous CDMs and computing the confusion matrix for each yields Fig. 2.12. It seems that autonomous CDMs have fewer clear patterns for risk-level-maintaining CDMs. Further studies should be done to fully understand the model.

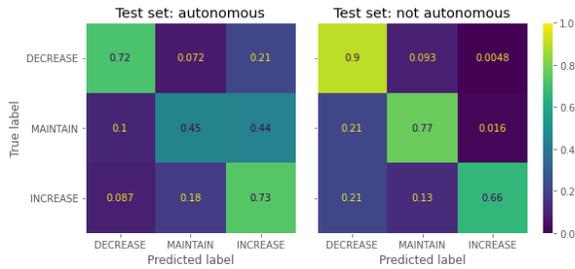


Figure 2.12. Confusion matrix on the test set, for autonomous and non-autonomous products.

### 3. State vector propagation enhancement

This section proposes a methodology to enhance orbital propagation applied to conjunction events using machine learning techniques. After a sensor has obtained observations of an RSO, its position in space and the orbit on which it is located are estimated. Using this initial estimate and using a physical model-based orbit propagation method, the object's state vector is propagated to a future epoch (the TCA). Both the initial

estimate and the orbit propagator introduce errors into the estimate of the final RSO state vector. The ML-based model is thus introduced to directly alter the propagation of the final satellite state so that the result obtained is closer to reality. There are examples in the literature of this very problem [5] [6]. However, these authors used support vector machines. In this work a more innovative approach is proposed using neural networks.

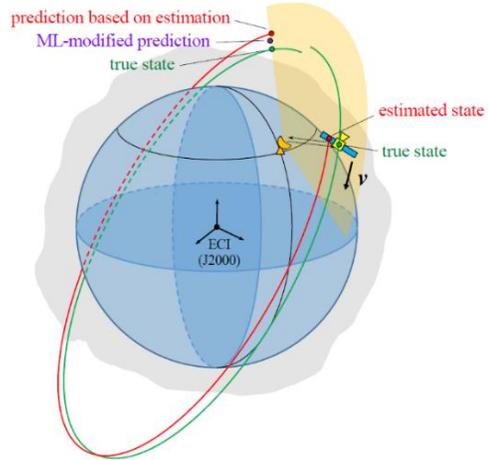


Figure 3.1 State vector propagation enhancement diagram [5]

It should be noted that the presented method is not influenced by the initial position of the RSO, but only uses the propagation based on the initial estimate (already included in a CDM) and other information about the satellite and its environment to reduce the error of the final estimate, without distinguishing between the causes of the error.

The overall objective of the method is to yield a correction to the propagated state vector provided in a particular CDM. This CDM has additionally a TCA computed with the best-known data at the time of publication ( $TCA_1$  in Figure 3.2). To provide a known truth of the state vector, the last CDM in the event chain is used. However, this last CDM contains a different TCA as the orbit of the object is, in principle, different ( $TCA_2$  in Figure 3.2). To evaluate the correction to the state vector computed by the model, the state vectors of both CDMs must be propagated to the same epoch. To that end, the state vector of the last CDM is propagated from  $TCA_2$  to  $TCA_1$ .

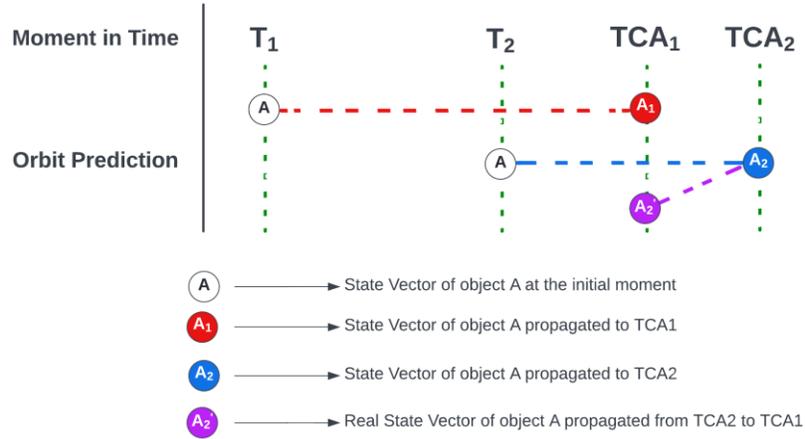


Figure 3.2 Propagation of ground truth state vector

### 3.1 Dataset

The dataset used in the current project consists of approximately 152.000 CDM files, coming from operator/owner users from EUSST and obtained through SpaceTrack. This dataset is separate from that presented in section 2.1. The period covered by the measurements in the data files is from 2016 to 2020, while the information and data in the files cover a wide variety of types of space objects following a diverse set of orbits.

The CDM files were received in XML format and they present information about a close approach event in which two objects, a target and a chaser, are involved. The focus of this project was put on the secondary object in each conjunction, as the number of primary objects is lower and less diverse. However, in the case of secondary objects, they add up to a total of 2223 individual objects, from which can be distinguished 1631 debris, 87 payloads, 64 rocket bodies and 441 objects whose type is classified as unknown. From all these objects, only 101 of them are known to be manoeuvrable, although, in the observations documented in the CDM files received, none of the space objects used their propulsion system to perform any sort of manoeuvre.

Regarding the models used to perform the state vector propagations, it is noted that the time period for propagation is no longer than 9 days. The reference frame in which the measurements were taken was ITRF and there have been used a total of 4 gravity models, which are comprised of series of geopotential models. The aforementioned models are: EGM-96: 36D 36O, EGM-96: 24D 24O, 36Z, 36T: 36D 36O and an additional custom one. By far the most used is the first one, being considered in more than 151.000 of the CDM files. In addition, the atmospheric density model used for the state vector propagation is JBH09, which also includes a solar

storm prediction model. The N-body gravitational perturbations used are the Moon and the Sun in almost all cases, as well as information about solar radiation pressure and earth tides. In terms of data measurements, the orbit determination process was based on average on 93 measurements. In addition, some other information about the space objects is known, such as their cross-section area, which ranges from 0.0005 to 24 square meters, its specific energy dissipation rate, the coefficient of the perturbation of the object due to atmospheric drag or due to solar radiation pressure.

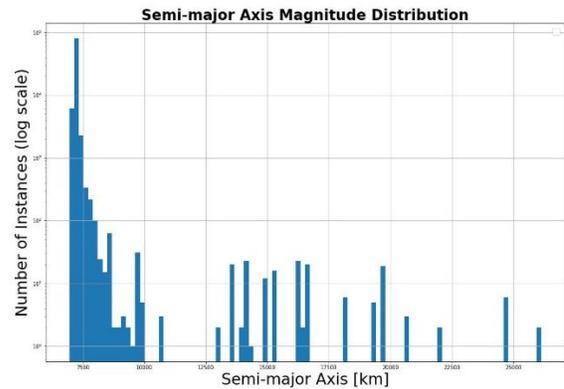


Figure 3.3 Distribution of semi-major axis of the secondaries

In terms of orbit shapes, the primary objects in the CDM files have polar orbits, and since the probability of conjunction is highest in the along-track direction, with objects of similar orbits, it is highly probable that the secondary objects also present polar orbits. Indeed, this is the case, as the number of orbits for the secondary objects with inclinations lower than 60 degrees is lower than 1000, meaning the rest of 151.000 CDM files presenting objects with inclinations between 60 and 90

degrees, indicating polar orbits. The altitude of the orbits is also not very diverse, with only about 1000 orbits being above 2000 kilometres above the Earth’s surface, meaning beyond the Low Earth Orbit threshold. In addition, in terms of eccentricity, the orbits are close to being circular, most of them having eccentricity levels below 0.05.

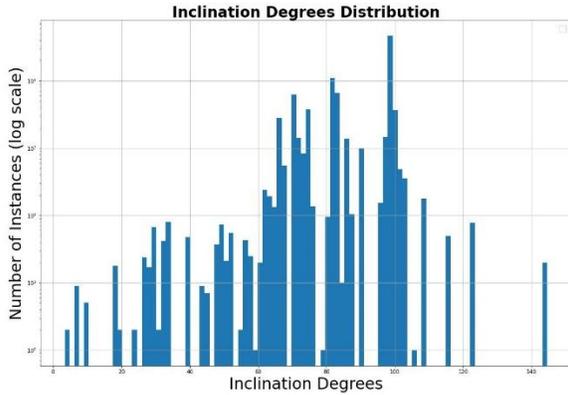


Figure 3.4 Distribution of orbital inclination of the secondaries

In terms of the components of the propagated state vectors, meaning the cartesian components for the position and velocity, the data is normally distributed in the cases of the X and Y components, with most values being very close to 0 and ranging from approximately -7000 km to +7000 km. In the case of the Z component, since most conjunctions seem to happen around the poles, the values on this axis are usually around +/- 7000 km.

### 3.2 Methodology

This section covers the methods and techniques used to train the ML models used for state vector propagation correction.

#### 3.2.1 Model Architecture

Tackling the problem at hand is well suited to a feed forward deep neural network. In this work the overall architecture used can be seen in Figure .

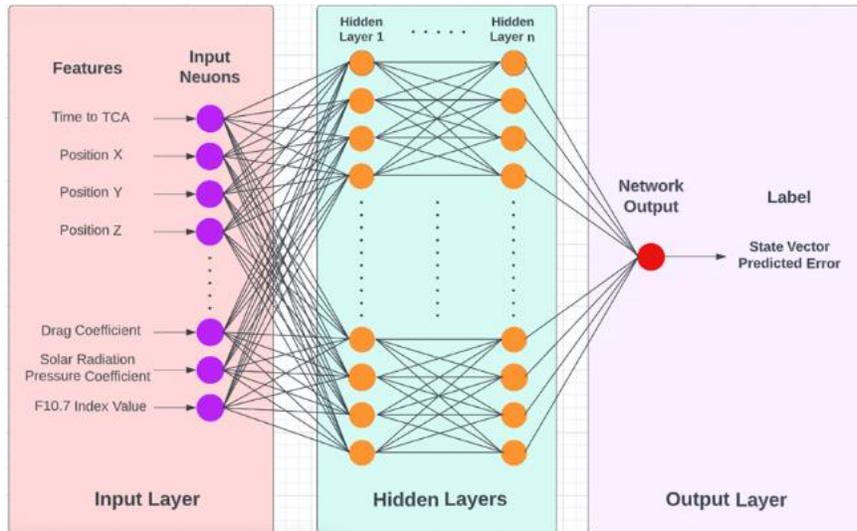


Figure 3.5 Model architecture

In this architecture, a set of inputs from a CDM is transformed into the prediction of the correction to the state vector as output. While the diagram shows the overall architecture of the model, several hyper-parameters were used to define the specific properties of each model trained. In this case the hyper-parameters available are:

- Number of intermediate layers.
- Number of neurons per intermediate layer.
- The activation function of each layer.
- The use or not of a dropout.
- Loss function.
- Optimization algorithm.

- Initial learning rate of the optimizer.

A hyper-parameter search was performed by creating multiple architectures with different such parameters and comparing their performance to determine the best configuration. The selection was made performing a random search and the performance of each configuration is determined using the validation dataset to choose the model best able to generalize the relationship between input and target parameters.

#### 3.2.2 Dataset Pre-processing

As with any ML algorithm, at least two datasets (ideally three) are required: the training dataset, the

dataset used both for validating training performance and for comparing hyperparameter performance and the test dataset, used to determine the ability of the algorithm to generalize on a yet unseen dataset - the latter dataset is used in the determination of model performance.

In this case, the way in which the datasets were chosen from the complete dataset is as follows:

- Training data set: 39.2%
- Data set for validation: 30.8%
- Test data set: 30%

The CDMs were randomly divided into the three datasets keeping events together, with the percentages chosen from experience.

In order to facilitate training, the data must be scaled before being fed into the model. A variety of scalers were used as appropriate. For relatively uniform a linear scaling to a range of 0-1 was enough. For variables with more complex distributions, quantile transformer or logarithmic based transformers were employed.

Regarding the features used to generate the propagation correction, all of them are available within a CDM except for the F10.7 index. No features that regard the primary object specifically are used. All features and their corresponding scalers are presented in Table 3.1.

Table 3.1 Features and scaler used

Feature description	Scaler used
Time to TCA	Quantile transformer
Propagated RSO position at TCA	Quantile transformer
Propagated RSO velocity at TCA	Scaling to range 0-1
Semimajor axis	Quantile transformer
Eccentricity	Log transformer
Inclination	Quantile transformer
RAAN, argument of perigee and true anomaly at TCA	Scaling to range 0-1
Number of observations used	Yeo-Johnson
Residuals accepted for orbit determination	Yeo-Johnson
RMS of residuals	Yeo-Johnson
Effective RSO area	Yeo-Johnson
Drag coefficient to mass ratio	Yeo-Johnson
Solar radiation disturbance coefficient to mass ratio	Yeo-Johnson
Specific energy dissipation rate	Yeo-Johnson
Covariance matrix elements at TCA	Quantile transformer
F10.7 index	Quantile transformer

### 3.2.3 Training methodology

In this section, the process followed to train the models is discussed. The main objective is to train models that are free from each individual dataset bias and that are able to generalize outside the dataset they are trained in.

To that end, the model is trained using early stopping. Early stopping is a technique by which the model is evaluated on a small section of the training set not used for actual training. If the loss on that subset did not decrease for a set number of epochs the model is considered trained.

Trained models are then evaluated on the validation set. This allows for the comparison of different sets of hyper-parameters. The model with the best performance on the validation set is chosen as the best set of hyper-parameters.

Finally, the actual performance of the model is evaluated in the test set which had not been used until this point. This overall process minimises the risk of biases towards the training set, by choosing models based on the validation set performance, and towards the validation set, by obtaining the final performance of the model with the test set.

### 3.3 Results

The following subsections show the absolute and relative error distributions of the ML model and the classical propagator, in the RTN coordinate system. The origin of the coordinate system is taken to be the centre of mass of the RSO at time TCA of the input CDM, i.e., the origin is at the point that determines the actual position of the satellite at the time for which the orbit propagation is done. The direction of the R axis is the same as the direction of the satellite position vector, the direction of the T axis is in the direction of its velocity, and the direction of the N axis is given by the vector product of the R and T axis vectors.

In order to provide a measure of the performance of the model, a baseline model is provided. This baseline is a null model that always yields a correction of zero. That is in effect, the best state vector available to an operator without the use of the ML algorithm presented here.

This coordinate system is used in error analysis as it illustrates in a more intuitive way the behaviour of ML model predictions than the geocentric coordinate system in which the model makes its predictions. As is also shown in the following subsections, the errors on the T-axis direction in the case of the classical predictor are the largest, being two orders of magnitude larger than the errors on the other two axes. This is most likely due to uncertainties in the modelling of the drag force, since it acts in the direction of the velocity (i.e., the T-axis), in the opposite direction. This tendency for model error cannot be observed using the geocentric reference system as a benchmark, which is why the analysis presented in this section was carried out.

### 3.3.1 Radial position component performance

This subsection shows the performance of the model against that of the baseline in the radial direction for the position components of the state vector. Figure shows the comparison of the distribution of errors between the model and the baseline. The model yields smaller errors than the baseline. Not only is the 95<sup>th</sup> percentile (upper whisker in the plot) lower, thus providing a reduction of the largest error expected. The median value is also smaller.

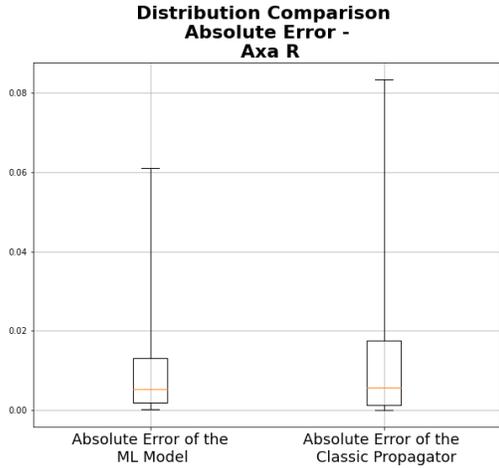


Figure 3.6 Distribution of baseline and model errors in radial position in kilometres.

One can also pay attention to the performance as a function of the time to TCA of the prediction as shown in Figure .

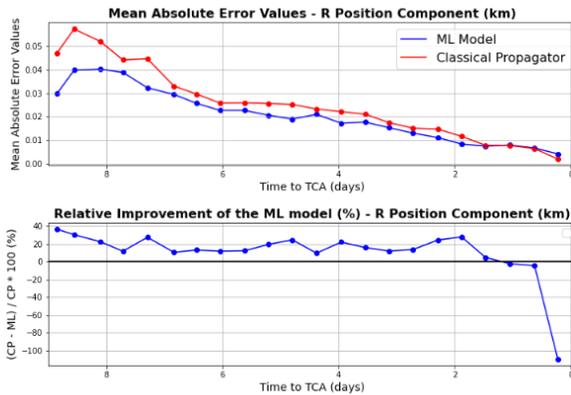


Figure 3.7 Model performance in radial position as a function of time to TCA

Again, the model gives an improvement of  $\approx 20\%$  until 2 days to TCA at which point performance is equal to the baseline until the last point in the plot where the baseline is better than the model. However, at this point

the correction is not so valuable as the propagation is already very close to the true value.

### 3.3.2 Tangential position component performance

This subsection compares the performance of the model against the baseline in the tangential direction only considering the position.

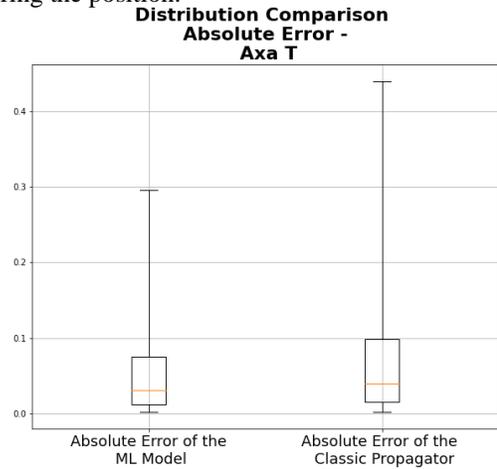


Figure 3.8 Distribution of baseline and model errors in tangential position in kilometres.

From Figure , the model outperforms the baseline significantly, both in the upper bound (95<sup>th</sup> percentile) and the median value. This result is particularly interesting as the majority of the uncertainty of the error resides in the tangential direction.

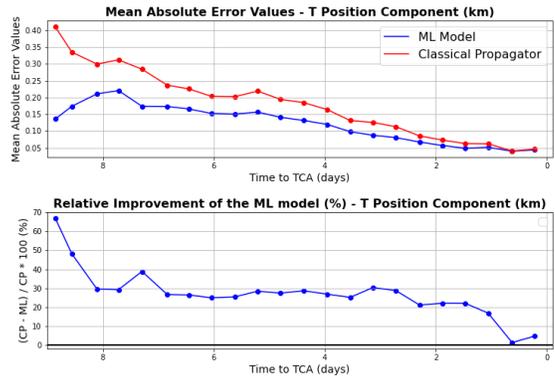


Figure 3.9 Model performance in tangential position as a function of time to TCA

From Figure , the model has a distinct advantage over the baseline throughout the whole event chain. Maintaining a 30% error decrease during most of the time span shown.

### 3.3.3 Cross-track position component performance

In this section, the cross-track component of the position error is analysed. First, the distribution of errors

can be seen in Figure . The model outperforms the baseline in all significant points of the plot. Particularly in the 95<sup>th</sup> percentile upper whisker. This can also be seen in Figure where the model outperforms the baseline by a small margin except for at the very closest time to TCA where errors are low in general.

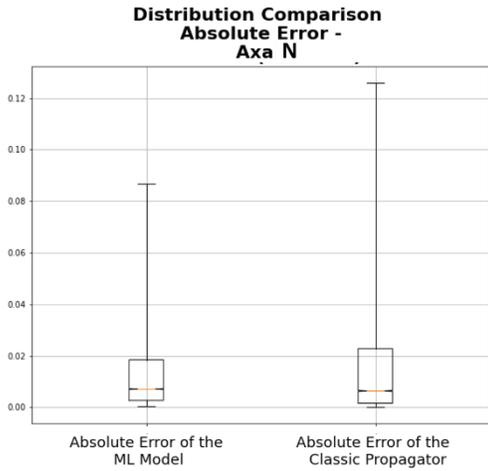


Figure 3.10 Distribution of baseline and model errors in cross-track position in kilometres.

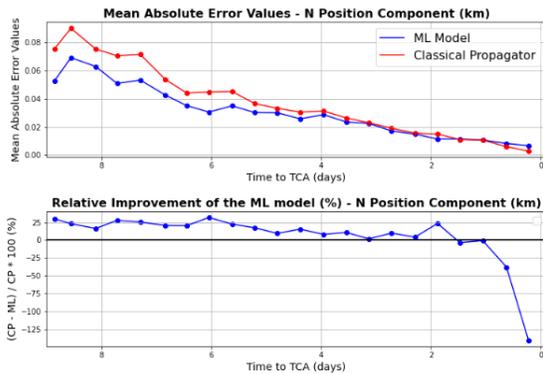


Figure 3.11 Model performance in cross-track position as a function of time to TCA

### 3.3.4 Radial velocity component performance

This subsection covers the analysis of the results of the model against the baseline for the velocity component in the radial direction.

The best model found offers significant improvements over the baseline throughout the considered range of time to TCA. As seen in figures 3.12 and 3.13.

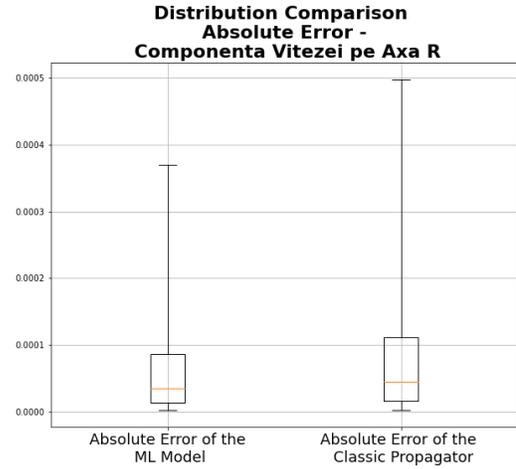


Figure 3.12 Distribution of baseline and model errors in radial velocity in kilometres per second.

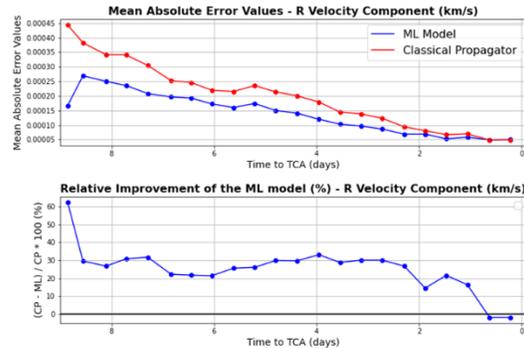


Figure 3.13 Model performance in radial velocity as a function of time to TCA

### 3.3.5 Tangential velocity component performance

This subsection shows the performance comparison between the model and the baseline in the tangential direction for the velocity of the RSO.

In this case, the best model found actually performs worse than the baseline in both plots shown Figure 4 and Figure . Therefore, it is not advisable to employ the model for this particular case.

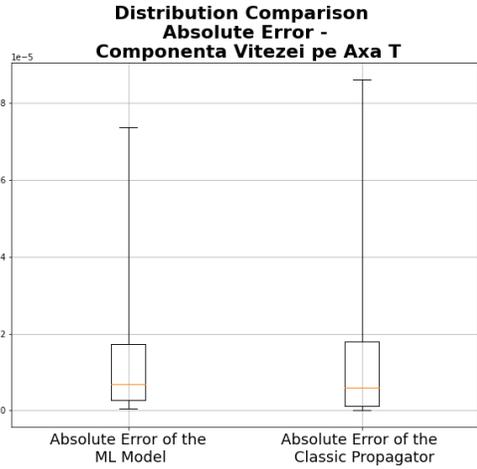


Figure 3.14 Distribution of baseline and model errors in tangential velocity in kilometres per second.

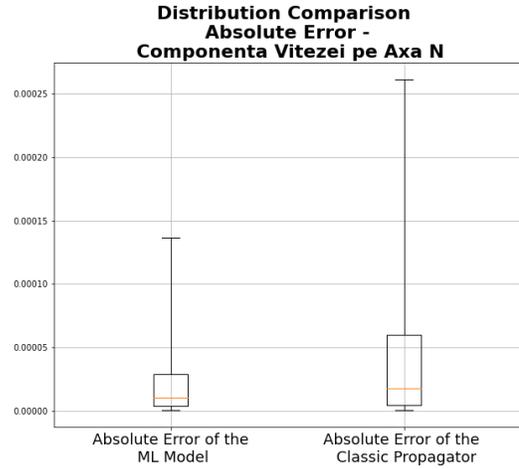


Figure 3.16 Distribution of baseline and model errors in cross-track velocity in kilometres per second.

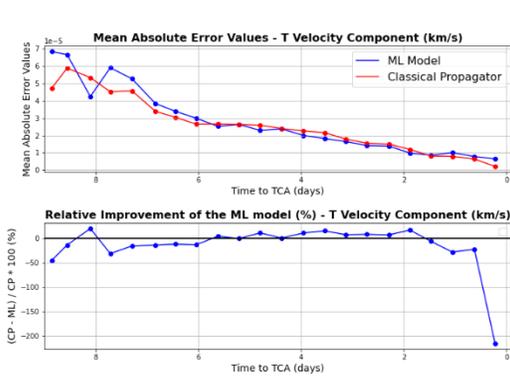


Figure 3.15 Model performance in tangential velocity as a function of time to TCA

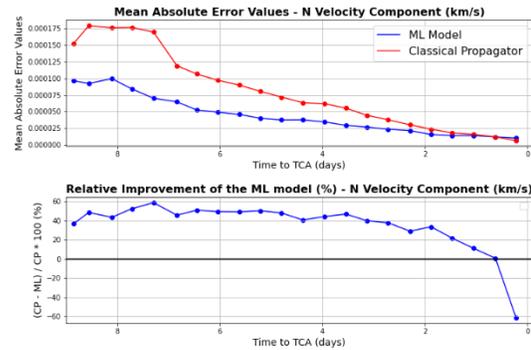


Figure 3.17 Model performance in tangential velocity as a function of time to TCA

### 3.3.6 Cross-track velocity component performance

This subsection analyses the performance given by the model in the cross-track velocity direction.

This is the only direction for the velocity in which the model offers an advantage over the baseline as seen in Figure . In fact, from Figure , this increased performance is achievable at all times to TCA available in the test dataset.

## 4. Conclusions

This study has demonstrated the applicability of ML algorithms and modelling techniques to real conjunction analysis scenarios obtained from operational context of SSA in S3TOC for two tasks, prediction of risk level change and state vector propagation enhancement.

For the first task, its applicability has been demonstrated successfully in the prediction of risk change in a CDM. This functionality can be useful if it is integrated into an operational conjunction analysis tool that can help operators in their daily work. It was demonstrated that the system is capable of predicting whether an event, using information from a single CDM, would change its risk level, especially if it would decrease (86%) or increase (72%).

In the course of this study, several problems have been encountered which have been solved along the way at data ingestion and pre-processing level because too many CDMs were incomplete, missing non-mandatory data like the probability of collision, which could have been useful and had to be discarded as a feature to the model. Either way, only fully healthy data has been used

being the main problem to discard bad data. In machine learning projects, usually the preparation of the dataset takes the most time of the whole process. In any case, since it is a study to demonstrate the capabilities of ML to reduce the number of manual operations in CA, there are other possible improvements in the process. For example, the upsampling technique used to balance the training set, or the generation of other interesting features related to time series trending. During the training, the inclusion of a validation set extracted from the training set as part of a cross-validation set might improve the quality of the training processes.

The second problem tackled has shown great promise. A reduction of the propagation error can be achieved in the vast majority of cases. Often with significant decreases of  $\approx 30\%$ . This has been achieved with a limited dataset and using singular CDMs as input. Therefore, this work could be generalized outside SSA and be applied generally to reduce the errors committed by propagators. By using an initial propagation as input to the model, the model was allowed to focus on the trends found the propagation errors. In this way, the known physics that have an effect on the trajectory of the RSO need not be learnt by the model which reduced computational complexity greatly.

Finally, the deployment of such models could be considered, but always taking into account a feedback loop to complement it and verify its results. ML algorithms require constant supervision, update and validation to ensure that the results are still consistent with the reality. The mere inclusion of such models can change the nature of the data since the focus of the operators might change. Further developments and analysis would be required to ensure that worrisome biases are not introduced into an operational environment.

## 5. Future work

In this work, it has been demonstrated that ML can be applied to conjunction analysis. However, several improvements could be made to either of the two applications to possibly increase the performance and reliability further.

For the first task, there are two possible places where to do further developments and research: the dataset and the hyperparameters of the model. The dataset could be improved removing corner cases that might not be useful to the model and might create a bias towards conjunctions that is not usually treated as such, like flight formation or micro-constellations. In the future, active debris removal and in-orbit servicing will need to be taken into account for the same reasons. In the side of the dataset, the resampling method used is very basic and could be substituted by a more advanced one, so the dataset does not use exactly the same samples. Secondly,

the training and validation of the model can be improved by adding new features considering the history of the full event (or even switching to a model prepared for time series), as well as using cross-validation during the hyperparameter tuning, so a better set of hyperparameters is found, one that improves the quality of the validation metrics.

After the second task, several possibilities for future development are possible. First, a lack of performance for velocity components was observed as compared with positional components. To that end, further hyperparameter search could prove useful in improving performance. Simultaneously, different ML architectures could be used. For instance, sequential models could be used to consider the whole sequence of CDMs instead of just the latest one. Providing additional information should, theoretically, improve overall performance.

Finally, it would be of interest to test the model outside the dataset used for this work. This dataset was dominated by LEO polar orbits. Evaluating the model for other regimes and/or orbit types would increase the validity of the results presented.

## Acknowledgements

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 760459.

Disclaimer: The content of this paper reflects only the view of the SST Cooperation and the European Commission and the Research Executive Agency are not responsible for any use that may be made of the information it contains.

## References

- [1] Conjunction Data Message. Recommended Standard. Blue Book. June 2013. CCSDS 508.0-B-1
- [2] H. Klinkrad. Space Debris: Models and Risk Analysis. (2006) 10.1007/3-540-37674-7
- [3] R. Shwartz-Ziv, A. Armon. Tabular data: Deep learning is not all you need. *Information Fusion* 81 (2022) 84-90.
- [4] T. Hastie, R. Tibshirani, J.H. Friedman. The elements of statistical learning: data mining, inference, and prediction, 2<sup>nd</sup> ed, Springer, New York, 2009.
- [5] H. Peng, and X. Bai, Improving Orbit Prediction Accuracy Through Supervised Machine Learning, 2018, pp. 1-30, <http://arxiv.org/abs/1801.04856>.
- [6] H. Peng, and X. Bai, Exploring Capability of Support Vector Machine for Improving Satellite Orbit Prediction Accuracy, *Journal of Aerospace Information Systems*, 2017. doi:10.2514/1.1010616.