aerospace

Article

Design of an ATC Tool for Conflict Detection Based on Machine Learning Techniques

Javier Alberto Pérez-Castán *, Luis Pérez-Sanz, Lidia Serrano-Mira, Francisco Javier Saéz-Hernando, Irene Rodríguez Gauxachs and Víctor Fernando Gómez-Comendador

ETSI Aeronáutica y del Espacio, Universidad Politécnica de Madrid, Plaza del Cardenal Cisneros, 28008 Madrid, Spain; l.perez@upm.es (L.P.-S.); lidia.serrano@upm.es (L.S.-M.); javier.saez.hernando@alumnos.upm.es (F.J.S.-H.); irene.rodriguez.gauxachs@alumnos.upm.es (I.R.G.); fernando.gcomendador@upm.es (V.F.G.-C.)
* Correspondence: javier.perez.castan@upm.es

Abstract: Given the ongoing interest in the application of Machine Learning (ML) techniques, the development of new Air Traffic Control (ATC) tools is paramount for the improvement of the management of the air transport system. This article develops an ATC tool based on ML techniques for conflict detection. The methodology develops a data-driven approach that predicts separation infringements between aircraft within airspace. The methodology exploits two different ML algorithms: classification and regression. Classification algorithms denote aircraft pairs as a Situation of Interest (SI), i.e., when two aircraft are predicted to cross with a separation lower than 10 Nautical Miles (NM) and 1000 feet. Regression algorithms predict the minimum separation expected between an aircraft pair. This data-driven approach extracts ADS-B trajectories from the OpenSky Network. In addition, the historical ADS-B trajectories work as 4D trajectory predictions to be used as inputs for the database. Conflict and SI are simulated by performing temporary modifications to ensure that the aircraft pierces into the airspace in the same time period. The methodology is applied to Switzerland's airspace. The results show that the ML algorithms could perform conflict prediction with high-accuracy metrics: 99% for SI classification and 1.5 NM for RMSE.

Keywords: air traffic; conflict detection; machine learning; ATC tool

Citation: Pérez-Castán, J.A.; Pérez-Sanz, L.; Serrano-Mira, L.; Saéz-Hernando, F.J.; Rodríguez Gauxachs, I.; Gómez-Comendador, V.F. Design of an ATC Tool for Conflict Detection Based on Machine Learning Techniques. *Aerospace* **2022**, *9*, 67. https://doi.org/10.3390/aerospace9020067

Academic Editors: Spiros Pantelakis, Andreas Strohmayer and Liberata Guadagno

Received: 26 November 2021
Accepted: 20 January 2022
Published: 26 January 2022

1. Introduction

In the same way that air traffic is constantly increasing, technological development is needed to deal with this air traffic demand. Single European Sky ATM Research (SESAR) is the European solution to tackle the evolution of the Air Traffic Management (ATM) system. One of the goals of SESAR is to research the feasibility of using new technologies in the ATM system [1] and, in particular, the Air Traffic Control (ATC) system. Automation is one of the pillars to ensure that traffic increases are managed safely [2]. This paper deals with this issue by developing a new data-driven approach for conflict detection between aircraft.

An accident is the worst event that can occur in ATM [3]. An accident means that the whole system and all safety barriers have failed. A conflict is a precursor of an accident. The ICAO defines a conflict as any situation involving aircraft and hazards in which the applicable separation minima may be compromised [4]. Then, it is crucial to avoid separation infringements that could lead, eventually, to a collision. Since the 1960s, several authors have developed different methods to study conflict and collision risk. Conflict and collision risk are metrics that analyse the level of safety for the ATM system. Reich [5,6] pioneered the development of collision risk modelling (CRM) for parallel routes in oceanic airspace. His work was paramount because it settled the pillars of collision risk based on random flight errors (positioning and velocity). Other authors have employed more or less complex techniques to evaluate collision and conflict risk [7–11]. The authors recommend the work of Netjasov and Janic [12] for a deeper review of CRM.

However, conflict detection differs from conflict risk because it is based on a tactical level (separation provision) instead of a strategic level (airspace design). Conflict detection tools aim to help Air Traffic Controllers (ATCos) to identify conflicts in the airspace. Besides this, conflict detection is typically studied at three different levels: long, mid-, and short term. This work analyses the mid- and short-term horizon (2–10 min) for ATC tools. ATC tools automatize conflict detection, helping ATCos in their labour to avoid separation infringements and reducing their workload. There are many different ATC tool models which were developed for conflict detection. Krozel et al. [13] stated that they can be divided into four areas depending on being static, being dynamic, uncertainty, and probability. Paielli et al. [14] developed a new approach based on tactical pairwise trajectory analysis. Other authors found their studies on complex probabilistic models to understand trajectory uncertainty [15–18].

Here, the applicability of data-driven approaches is analysed for conflict detection. Machine Learning (ML) is one of the branches that takes advantage of historical data. The European Aviation AI high-level group defines ML as "the ability of algorithms to learn from the input and output data that characterise them" [19]. The goal of ML algorithms is to learn patterns from historical situations (databases) that underline those situations. The output of the algorithm is to apply the learned rules to predict new situations. Typically, there are three types of ML algorithms depending on the input/output data required: supervised, non-supervised, and reinforcement learning. Supervised learning demands the output associated with the input data, unsupervised learning acquires patterns from the input data without knowing the output data, and reinforcement learning makes the model learn a sequence of decisions based on rewards and penalties. In addition, ML algorithms can be divided in turn into classification and regression problems. Classification problems separate samples into different classes (binary or multiclassification). Regression problems provide numerical predictions. The authors recommend the works [20,21] to understand these topics.

Aviation is a field with a huge amount of data that is increasingly available for research purposes. ML techniques take advantage of the data, and have recently been applied to different topics: trajectory prediction [22–24], airspace performance metrics [25,26] and atmospheric models [27,28]. One of the issues of employing a data-driven approach is conflict generation. Up until now, conflict simulation has used different methods. Top-down models generate customised conflicts in advance, based on predefined situations [29]. Other approaches perform simulations adding uncertainty to different operational variables (wind, weight, velocity, etc.) [29–32]. The goal is to obtain an extensive database with enough separation infringements, which is not trivial.

Therefore, this work aims to develop the pillars of a further ATC tool for conflict detection based on ML techniques. One of the strong points is the addition of Four-Dimension Trajectory (4DT) predictions based on historical ADS-B trajectories. This allows us to improve the ability to predict conflicts based on 4DT predictions. After the introduction, the framework for the ATC tool based on ML techniques is presented, and the data-driven approach is explained for the extraction of trajectories to constitute the database. Section 3 details the selected ML techniques and the process followed in order to apply them. Section 4 shows the results obtained for the classification and regression techniques. Conclusions are presented in Section 5.

2. Framework for ATC Tools Based on ML Techniques

A conflict detection ATC tool aims to provide information to the ATCo about aircraft pairs which are expected to infringe the separation minima. The ATC tool developed in this work is based on a data-driven approach using ML techniques. Here, it is essential to note that the ATC tool provides predictions focusing on conflict detection but not trajectory prediction. This means that the ML model does not perform the trajectory prediction, nor does it afterwards analyse if a conflict could occur; it performs the conflict prediction based on what happened in previous situations. This novel approach employs ML algorithms to learn the underlying factors that lead to separation infringement by including historical 4D trajectories as predictions. The primary characteristics and hypothesis are the following:

1. The ATC tool makes predictions based on the evolution of the aircraft throughout the airspace. The predictions depend on the time (t) and the operational features of both aircraft. The ATC tool updates the prediction every c minutes.
2. The ATC tool obtains the aircraft within the sector and the aircraft in proximity of the airspace that will penetrate the airspace in the following m minutes. c and m values should be defined by Air Navigation Service Providers (ANSPs) in advance.
3. The ATC tool receives a 4DT prediction for each aircraft based on historical data and the operational features (state vector) based on ADS-B data. The ML model uses both inputs.
4. The ATC tool was developed for the tactical ATCo, which continuously monitors the state of the aircraft throughout the airspace.
5. The ATC tool does not provide information on conflict resolution.

Figure 1 represents the operational concept of the ATC tool. Both aircraft i and j are flying in the airspace. The ATC tool performs conflict prediction between them, along with other aircraft flying in the airspace and the aircraft that will pierce in m minutes. The ATC tool demands two types of information: the state vector at the moment of the prediction and 4DT predictions. The state vector of each aircraft is accessible by different data sources based on ADS-B. However, the main problem is obtaining 4DT predictions of the aircraft. Typically, 4DT predictions are calculated by the ground server or provided by the aircraft throughout the flight. We do not have access to this type of prediction, and it should be one of the topics to research in further work. Hence, 4DT predictions are obtained by historical 4DT trajectories that fly in the same airspace. It is assumed that aircraft have flown similar trajectories in the airspace in a recent period of time.

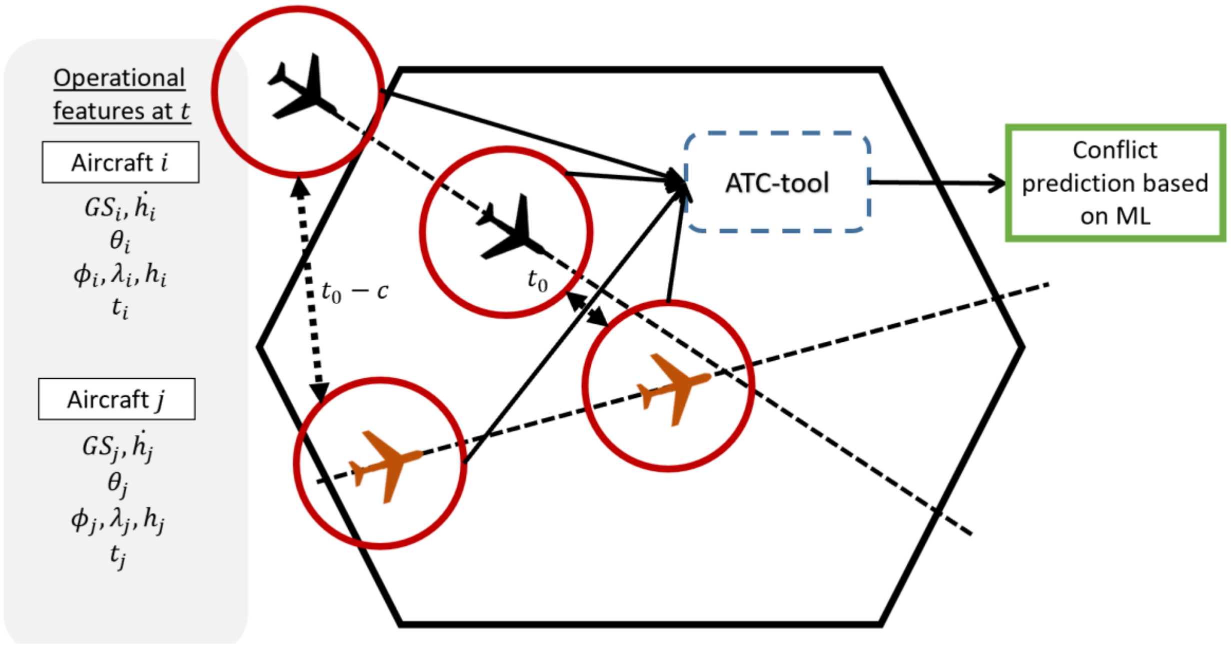

Figure 1. Operational concept of the ATC tool.

2.1. Conflict Detection Principles

Conflict detection aims to identify or detect aircraft pairs which are expected to infringe the separation minima in the short term (2–5 min). The current separation minima in the en-route airspace are 5 Nautical Miles (NM) horizontally (S_{min}) and 1000 feet (ft) vertically (H_{min}). The concept of the short term is ambiguous because it depends on the time frame considered. The strategical time frame (from several hours to one year in advance) is different from the tactical time frame (during the aircraft operation). This work focuses on a tactical time frame because it tries to develop a tool to help ATCos detect conflicts in the airspace.

One of the primary issues that affects ATC tools for conflict detection is the definition of the limits to inform us about whether there is a conflict or not. The separation monitoring of an aircraft pair throughout the airspace is not accurate because it presents several uncertainties based on navigation and surveillance. Two types of situations can arise due to the poor performance of conflict detection.

- Missed Alerts: The ATC tool does not inform us about the separation infringement between an aircraft pair because the system erroneously calculates that no separation infringement will occur, when in fact it will.
- False Alerts: The ATC tool informs us about the separation infringement between an aircraft pair because the system erroneously estimates a separation infringement when in fact it is not one.

Typically, ATC tools expand the limits of separation infringement to avoid critical missed alerts that identify aircraft pairs that cross with a separation larger than the current separation minima. To this end, this work uses the concept of a Situation of Interest (SI). One SI is a situation in which an aircraft pair is expected to intersect with a horizontal separation smaller than a predefined distance. Typically, this predefined separation is specified by the ANSP, and is larger than the current separation minima. In this work, an SI was defined as 10 NM.

Therefore, this work evaluates two metrics regarding the separation minima reached by a pair of aircraft.

1. Minimum Distance ($MinDis$): This is the minimum separation reached by an aircraft pair $\left(a_i,\ a_j\right)$. This variable considers the horizontal separation (s) and vertical separation (Δh), and provides information about the severity of the SI. The $MinDis$ considers two cases depending on the vertical separation—(1) the aircraft pairs cross with a vertical separation lower than the separation minima (H_{min}), and (2) the aircraft pairs intersect with a vertical separation higher than the vertical separation minima—and it is transformed considering a horizontal scale:

$$\begin{aligned} MinDis &= \min\left(s\left(a_i,\ a_j\right)\ if\ \Delta h < H_{min}\right. \\ otherwise\ MinDis & \\ &= s\left(s\left(a_i,\ a_j\right) = \min\left(s\left(a_i,\ a_j\right)\right)\right. \\ &\quad + \Delta h\left(s\left(a_i,\ a_j\right) = \min\left(s\left(a_i,\ a_j\right)\right)\right) S_{min}/H_{min} \end{aligned} \tag{1}$$

2. Aircraft pairs are denoted as being of interest (SI) when they cross with a vertical separation lower than H_{min} and a horizontal separation $s < s_{SI}$:

$$\begin{gathered} if\ \Delta h\left(a_i,\ a_j\right) < H_{min}\ and\ s < s_{SI}\ for\ the\ same\ t\ \rightarrow \\ SI = 1,\ otherwise\ SI = 0 \end{gathered} \tag{2}$$

2.2. Data-Driven Approach

One requirement for ML techniques is to acquire a database from which the ML algorithm could learn the underlying patterns in order to perform predictions. The database must have enough historical situations based on actual aircraft pair trajectories. The selected data source affects the constitution and performance because each provides different information. Here, the data source is ADS-B trajectories from the OpenSky Network [33].

The first step is to train an ML predictor to perform conflict predictions for each pair of aircraft. To this end, a database must be used to train the ML models. This database should have enough situations to represent the different situations that can arise between pairs of aircraft. As such, the number of cases considered in the database is paramount because the larger the number of them, the better the prediction ability will be. On the contrary, increasing the number of samples means a higher computational time.

Typically, the database is split into two sets: the training set and the testing set. The ML model uses the training set to learn the underlying relations and develop a mathematical model to make predictions. The testing set is used to evaluate the performance of the trained model for new instances. The features considered in this work come from the ADS-B information:

- Position: Longitude (λ), latitude (ϕ) and altitude (h).
- Velocity: Ground speed (GS) and vertical rate ($\dot{h}$).
- Heading (θ).
- Target variables based on 4DT predictions.

Although there are other features from ADS-B, they do not provide information about the state vector of the aircraft. The labels or targets (variables to predict) are:

- Minimum distance ($MinDis$): A numerical variable of the minimum distance expected to reach between an aircraft pair.
- Situation of interest (SI): Binary variable that classifies aircraft pairs as SI or $No\ SI$.

2.3. ADS-B Traffic from the OpenSky Network

This work uses ADS-B trajectories from the OpenSky Network [33]. Raw data downloaded from the OpenSky network cannot be fed immediately to the ML algorithm. ML algorithms require previous filtering and adjustment to be used. Data preparation performs the following actions:

- Cleaning trajectories that present errors in the ADS-B data. Errors in the ADS-B data are identified by the OpenSky functions [33] and represent faulty data such as callsigns or repeated positions. Trajectories with more than 10 ADS-B errors are removed. Duplicate trajectories are also removed.
- Trajectories are gathered for each day and time period. One operational day is split into nine time periods based on the traffic distribution throughout the operational day. The first period covers from 0 to 8 a.m., and the traffic is removed because the LSAZM567 is not opened (due to the low traffic); the rest of the time periods are sets of two hours, and assume similar weather conditions in the simulations.
- The generation of aircraft pairs based on the combinatorial problem.
- Relative variables that combine operational features for each aircraft pair are calculated.
- The generation of conflicts between aircraft pairs and the calculation of conflict metrics.

Figure 2 summarises the whole process.

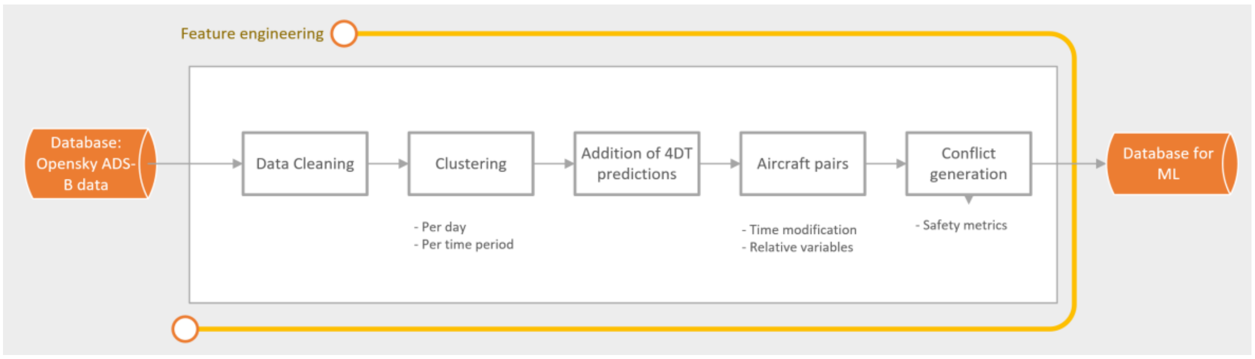

Figure 2. Flow of the feature engineering process.

Weather or operational variables of airspace status were not considered, and should be included in further work.

2.4. Historial ADS-B Database as 4DT Predictions

The operational concept considers 4DT predictions of the aircraft trajectories as inputs for the ML models. Each aircraft provides a trajectory prediction from the most basic information (the flight plan) until a 4DT prediction is calculated by the ground system. It was not possible to obtain this type of prediction because there was no access to the ground server or flight plans correlated with the ADS-B data. Therefore, the 4DT predictions were assumed to be historical trajectories stored from ADS-B data. It was assumed that the 4DT prediction can match with a similar trajectory stored in the ADS-B database. In case the system has another type of prediction, the process will be the same.

The main problem of using stored trajectories as 4DT predictions of other aircraft is ensuring the similarity between them. When one aircraft pierces into the airspace, the algorithm performs a search in the historical database in order to identify the most suitable trajectory that fulfils the following requirements:

1. The first filtering is about selecting one trajectory with the same callsign. Typically, aircraft repeat their trajectories.
2. The second filtering considers operational restrictions such as the ground speed, heading, and location of the entry point:
   - The GS difference must be lower than 10 knots.
   - The heading difference must be lower than 2°.
   - The location difference of the entry points must be lower than 5 NM.
3. In case there is more than one trajectory that fulfils the previous restrictions, it will select the trajectory with a lower heading difference.
4. If none of the trajectories with the same callsign satisfy the previous restrictions or there are no trajectories with the same callsign, the algorithm extends the search to other callsigns. The main issue is that the computational time to find a similar trajectory increases exponentially.

Figure 3 represents the selection of a 4DT prediction from the historical database that is the most similar to one random trajectory when the aircraft pierces into the airspace.

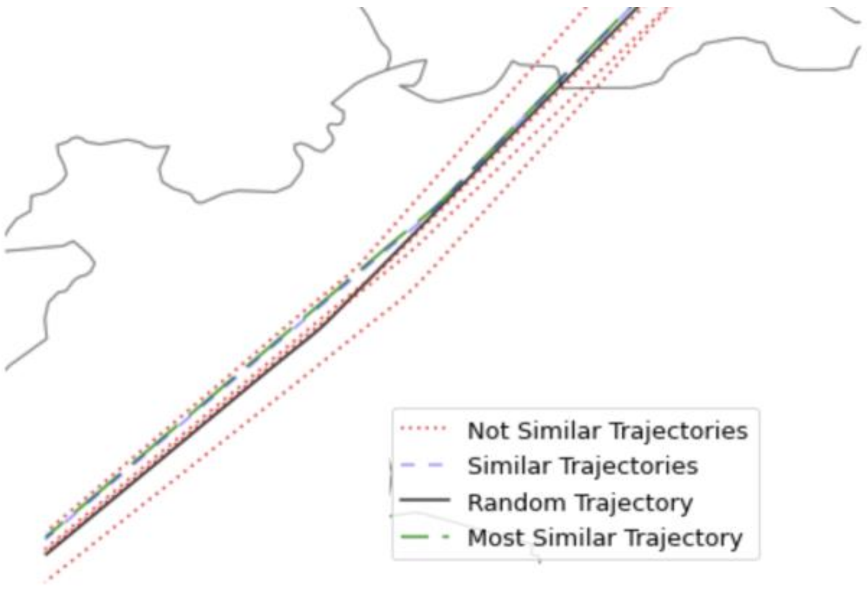

Figure 3. Representation of the selection of 4DT predictions.

Once the 4DT prediction is selected from the ADS-B database, two modifications are made to ease the interoperability. The first modification is about a resample of the 4DT prediction to reduce the number of samples that constitute the prediction. The goal of the resample is to reduce the computational workload and the size of the final database to consider due to the high number of samples. The resample is not an issue in the case where the aircraft fly straight lines, but if the aircraft is turning or performing a manoeuvre, it will not be reflected correctly.

The second modification aims to introduce uncertainty by adapting the 4DT prediction entry time to the actual trajectory. This uncertainty is a temporal deviation (τ) following a random distribution of ± 20 s. This is a value selected by the authors, and its suitability should be studied in further work. The higher this value, the larger the error between the actual trajectory and the 4DT prediction. The result is a database of 4DT trajectories composed of the actual trajectory and a similar trajectory considered as the 4DT prediction.

2.5. Generation of the Aircraft Pairs and Conflicts

The main limitation of the analysis of aircraft pairs extracted from ADS-B data is the lack of conflicts in real situations. Separation infringements safely occur in rare situations and under specific circumstances [34]. Conversely to these rare situations, ML models need a large and diverse database in order to learn the patterns that underlie separation infringement. Therefore, the first need is to generate a database with sufficient samples of conflict situations.

There are two ways to simulate conflicts. First, we can define the conflict parameters and then simulate backwards in order to obtain the conflicts which were previously defined. However, this solution does not encompass some operational parameters, such as airspace design or air traffic distribution. The second solution considers a real scenario and introduces modifications to the real trajectories in order to force separation infringements. These simulated conflicts are not real conflicts, but represent situations that could occur. These can be assumed to be pre-tactical conflicts based on real trajectories that consider many operational parameters, such as airspace design, speed uncertainty, and wind, etc. This work adopts the second approach and performs simulations that modify the entry time of the aircraft in a limited time period. In this way, an algorithm was developed to generate conflict situations by modifying their entry time. The process is as follows:

(1) First, each aircraft modifies its entry time randomly between the temporary boundaries (t^{init}_{entry}, t^{end}_{entry}). The temporary restriction is that every aircraft should be in the airspace between the limits t^{init}_{entry} and t^{end}_{entry}. The new entry times are calculated randomly for each aircraft.

(2) Secondly, the aircraft pair generation is constituted. Suppose that there is a set of n trajectories that constitutes $(n-1)n$ aircraft pairs. For each aircraft i is generated an $(n-1)$ aircraft pair to evaluate.

(3) For each aircraft pair at each time t, the database is constituted by storing the operational features of each aircraft pair.

(4) Relative variables and conflict metrics are calculated for each aircraft pair, both for the actual trajectory and the 4DT prediction. Conflicts when one aircraft pierces into the airspace are discarded because this would imply that the ATCo from the previous airspace would not have done his work correctly.

Figure 4 represents this temporary modification and the generation of pairs for aircraft i. Aircraft i pierces at time t_{a_i}, and the rest of aircraft receives aleatory new entry times between the time boundaries. The constitution of the aircraft pairs is repeated for every aircraft, avoiding duplicity.

Relative variables provide information about any situation that combines the operational features of a pair of aircraft. Relative variables are calculated at each timestamp when operational information is available. The mathematical descriptions of the relative variables are as follows:

- Horizontal separation ($s_{i,j}(t)$) is the horizontal separation between the locations of an aircraft pair:

$$s_{i,j}(t) = position_i(t) - position_j(t) \tag{3}$$

- Vertical separation ($\Delta h_{i,j}$) is the altitude variation (h_i, h_j) between an aircraft pair:

$$\Delta h_{i,j}(t) = h_i(t) - h_j(t) \tag{4}$$

- Course ($\gamma_{i,j}$) is the course that links the locations between an aircraft pair:

$$\gamma_{i,j}(t) = \gamma_i(t) - \gamma_j(t) \tag{5}$$

- Track variation ($\Delta\theta_{i,j}$) is the track variation ($\theta_i,\ \theta_j$) between an aircraft pair:

$$\Delta\theta_{i,j}(t) = \theta_i(t) - \theta_j(t) \tag{6}$$

- Ground Speed (GS) variation ($\Delta GS_{i,j}$): is the difference in the module of the GS ($GS_i,\ GS_j$) between an aircraft pair:

$$\Delta GS_{i,j}(t) = \left|GS_i(t) - GS_j(t)\right| = \sqrt{\left(GS_i^x - GS_j^x\right)^2 + \left(GS_i^y - GS_j^y\right)^2} \tag{7}$$

- Vertical rate variation ($\Delta\dot{h}_{i,j}$) is the variation of the vertical rate ($\dot{h}_i, \dot{h}_j$) between an aircraft pair.

$$\Delta\dot{h}_{i,j}(t) = \dot{h}_i(t) - \dot{h}_j(t) \tag{8}$$

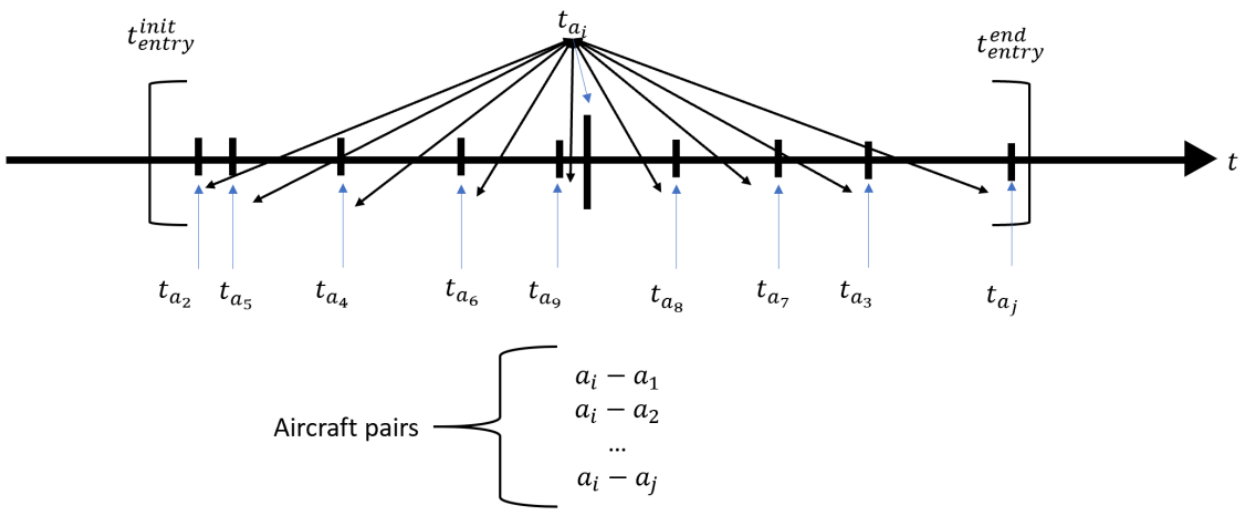

Figure 4. Generation of the aircraft pairs with temporary modifications.

The library *geographiclib* was used to calculate the relative variables based on aircraft positioning (longitude and latitude) because the library performs the calculations without transforming the aircraft positioning to Cartesian coordinates [35]. Therefore, 40 variables constitute the ML database: 12 ADS-B variables (six for each aircraft), six relative variables, two labels, and the same for the 4DT prediction.

2.6. Database for ML

The above process allows the performance of simulations to generate a database that could be used for ML models. The ML database presents situations of aircraft pairs that could constitute SI or not.

This work was developed by the AISA project, and focused on the LSAZM567 sector of the Switzerland airspace (from FL355 to FL660) [36]. LSAZM567 airspace is an en-route airspace where most of the airways are composed of straight lines. The 4DT were downloaded from the AIRAC cycle from 20 June 2019 to 04 July 2019 (15 days). Approximately 12,500 trajectories were downloaded, containing more than 7e6 samples (each sample represents an aircraft position). Table 1 shows the results for all of the simulations.

Table 1. Results of the simulations.

Variables	Number
Number of Aircraft	12,549
Number of pairs generated	1,257,234
Number of samples considered	675,3283
Samples with SI (%)	1,215,152 (18%)
Samples with conflict (%)	270,132 (4%)
Computational time (hours)	420

The number of aircraft varied from 80 to 120 for a one-hour period. The number of pairs generated was far from the whole combinatorial number of aircraft pairs because the simulations were performed for specific time periods. The total computational time required to perform the simulations was more than 420 h. This shows one of the issues that should be improved in future developments. Lastly, the simulations were performed in computer with Intel® Core™ i5-6600 CPU @ 3.30 GHz, RAM 8.00 GB, and 64 bits.

Figure 5 shows the histograms of the regression and classification targets.

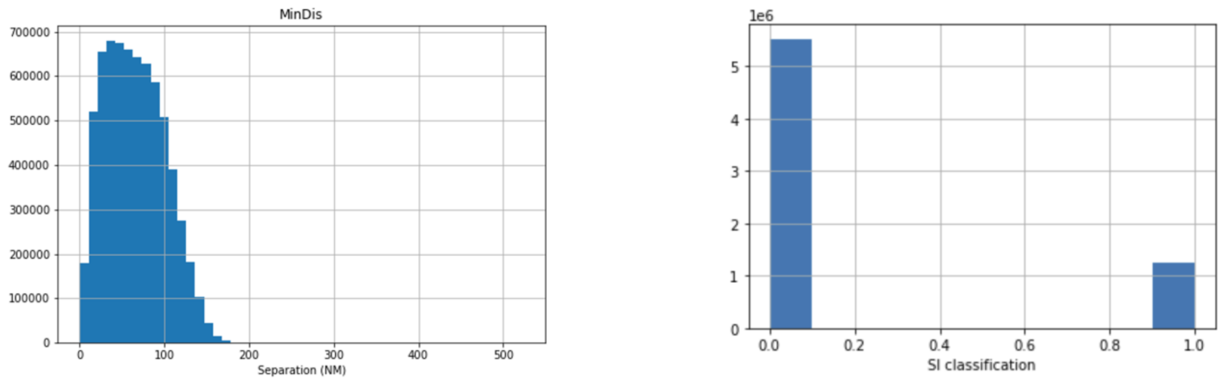

Figure 5. Histograms of the regression and classification targets.

As is shown in Figure 5, the $MinDis$ range of the distribution is from zero to more than 150 NM. The most significant density is between 10 and 20 NM. The $MinDis$ located at 0 NM refers to aircraft pairs that move away. Regarding the classification, there is a high imbalance between the SI and $No\ SI$ samples and conflict samples. These results show the problem of generating conflicts despite using specific simulations for this purpose.

3. Machine Learning Techniques

One of the current technological developments is the implementation of data-driven techniques, and particularly ML, which is one of the most promising technologies in data prediction. As explained above, this approach tackles two types of ML problems: classification and regression. The classification problem aims to classify aircraft pairs into SI or $No\ SI$ depending on the minimum separation that they are expected to reach. The regression problem seeks to predict the numerical value of the minimum separation they are expected to reach. It is important to note that both problems are solved independently. This allows one to analyse the validity of the combinations based on both predictions.

ML models are currently implemented in different programming languages. This work was developed in Python®. Scikit-learn is the library which was used to implement ML algorithms [37], and PyCaret is another library that eases this process [38].

3.1. ML Experiments

As the ML database shows, there is a high imbalance between the SI and $No\ SI$ samples. This is an issue in classification problems because the ML algorithm performs best with balanced databases. In order to tackle this problem, we considered a Hybrid model in which a filtering process based on operational restrictions was introduced to reduce the strong imbalance of the dataset. Then, there are two experiments:

1. Pure model: The ML model considers the whole database without any modification.
2. Hybrid model: The ML model considers a previously filtered database. The filtering is performed on the basis of operational restrictions. The database only contains aircraft pairs that cross with a horizontal separation of less than 20 NM and a vertical separation of less than 1000 ft.

The ML process performs the following steps to obtain the best ML model:

1. The analysis of different ML algorithms: The first step is to evaluate the performance of different ML algorithms in order to identify the ML model that provides better results. In total, 15 ML algorithms were assessed for classification and regression.
2. Feature selection: The feature selection identifies the influence of the different features that build the database, and analyses how they affect the model. The feature selection is performed based on graphical analysis and Recursive Feature Elimination (RFE) with CV [39]. RFE analysis evaluates the impact of the features on the model accuracy. Finally, features without influence on the ML model are removed from the database.
3. ML optimisation: The optimisation process aims to improve the algorithm's performance based on a grid search of the hyperparameters. The hyperparameters must be set up in advance of the training model, and can be defined as the settings of an algorithm. The metric to be optimised depends on the classification and regression problem.

The result of this process is the readiness of two trained and tested ML algorithms for conflict detection.

3.2. ML Database Preparation

The ML process requires the preparation of the database prior to feeding it to the algorithms. Data preparation modifies the raw data into valuable data for ML algorithms. Here, the following preprocessing activities are applied:

- The training and testing sets: The dataset is divided into the training and testing datasets. The testing set (also known as the hold-out set) works as a proxy for new data. It is not used in model training, and can be used to evaluate the model metrics' suitability. The database is split into 70% for training and 30% for the testing set.
- Normalisation: Normalisation rescales the values of the numeric columns following a normal distribution. It does not distort the differences in the range of values.
- Shuffling: This distributes the samples randomly into the training and testing datasets.

1. Stratification: Apart from randomly distributing the samples, the stratification process spreads the samples whilst maintaining the SI statistical distribution.
2. Cross-validation: This is a process which is used to avoid the overfitting of the ML model [39]. The training set is divided into k folds: one is considered the validation set, and the rest are considered the training set. This process is repeated for every k-fold, and the metrics are calculated based on the mean and standard deviation.

Table 2 shows the number of samples for both experiments; the rates of $SI/No\ SI$ samples appear in parentheses. Each sample represents the situation of one aircraft pair at one specific time.

Table 2. Number of samples for each experiment and set.

Model	Training Set	Testing Set
Pure model (18/82)	4,254,567	1,823,387
Hybrid model (68/32)	1,140,470	488,773

3.3. ML Algorithms and Metrics

Currently, there are several state-of-the-art ML algorithms, and we evaluate some of them. The evaluation of different algorithms allows one to identify the best among them. The goal of this section is not to explain the algorithms employed, but to denote which were trained: Extra Trees, Random Forest, Extreme Gradient Boosting, Light Gradient Boosting, CatBoost, Decision Tree, K Neighbors, Gradient Boosting, Ada Boost, Naïve Bayes, SVM—Linear Kernel, Logistic Regression, Ridge Classifier, Linear Discriminant Analysis, and Quadratic Discriminant Analysis. The authors encourage readers interested in the mathematical ML definitions to see [20]. Other ML techniques such as Deep Learning could improve the results, and should be evaluated in further work.

ML metrics are different for classification and regression problems:

- Classification problems include accuracy, precision, recall and F1. The most important is the F1 metric that leverages precision and recall. Accuracy is not a good option because the database is highly imbalanced.
- Regression problems include the Mean Absolute Error (MAE), Root Mean Square Error (RMSE), R2 and Root Mean Square Logarithmic Error (RMSLE). The most important metric is the RMSE, which increases the impact of the prediction distance from the true values.

4. Results

This section provides the results of the experiments, and are split into classification and regression. In addition, the performance of both predictors is analysed for different separation infringement stretches.

4.1. ML Techniques for Classification

ML classification techniques predict whether an aircraft pair is SI or not. The first step is to analyse the performance of different ML algorithms, aiming to identify the best one. The results of the different ML algorithms are shown in 'Appendix A ML results' in order to improve the reading of the paper. Both experiments agree that the three top algorithms vary between Extra Trees, Random Forest and Decision Tree, i.e., ensemble models. The random forest provided the best results for both experiments. The models were evaluated on the basis of stratified cross-validation techniques in order to avoid overfitting.

The feature selection for both experiments concluded that the most influential features were the prediction about the minimum distance and the SI prediction. The sum of both of them reached around 70% of the influence of the variable. The variation of separation is also important (up to 10%), and the rest of the features provide a similar impact between 1 and 5%. As can be seen in Figure 6, the feature selection is similar for both experiments (Pure and Hybrid models). This similarity confirms the significance of the prediction variables in the ML algorithm. The variables labelled '_n' refer to the aircraft trajectory, and '_re' refers to the 4DT prediction.

The next step is to optimise the Random Forest algorithm. The database is imbalanced in both experiments because the distribution of SI samples is not balanced. This issue is tackled by implementing cost-sensitive techniques during the optimisation process. We studied the optimisation of recall, precision and F1. However, the most balanced results were obtained by optimising the F1 metric. Table 3 shows the results of both experiments in the training set applying a fivefold cross-validation.

Table 3. Results of the classification experiments.

Experiment	Accuracy (std)	Recall (std)	Precision (std)	F1 (std)
Pure model	0.990 (0.01)	0.987 (0.01)	0.989 (0.02)	0.988 (0.01)
Hybrid model	0.991 (0.001)	0.990 (0.001)	0.991(0.001)	0.990 (0.002)

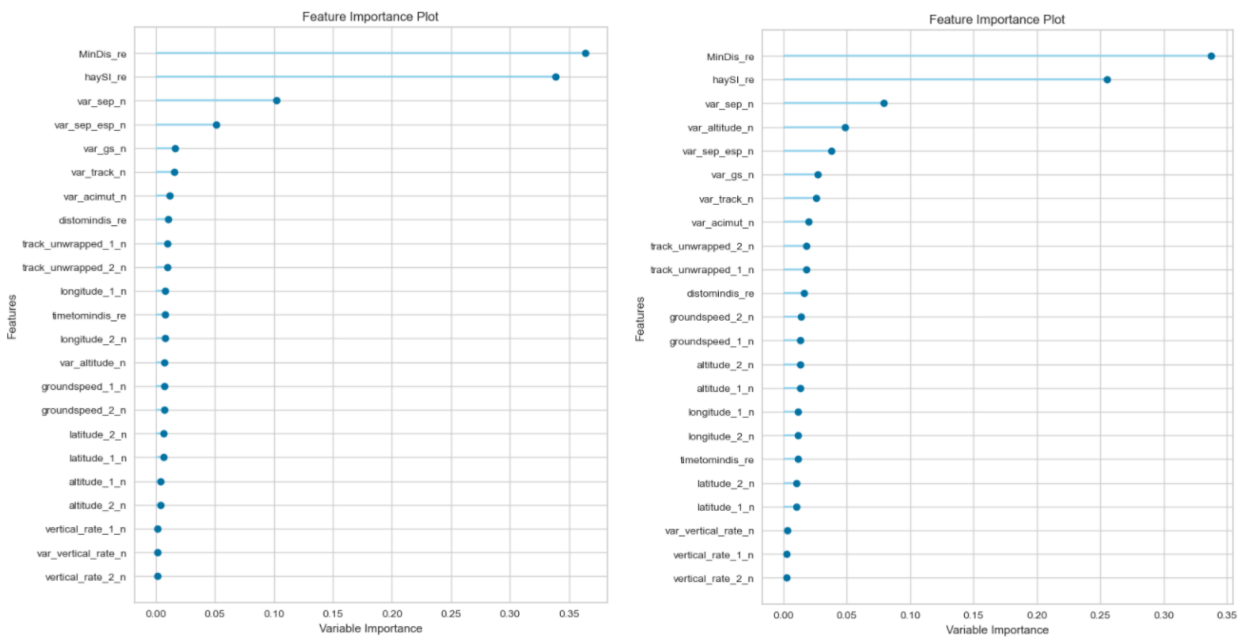

Figure 6. Feature importance of the Pure model (**left**) and Hybrid model (**right**) for classification.

The following conclusions were obtained:

- The optimisation process barely improved the initial metrics of the ensemble models. The initial values obtained for the ensemble models were very high, and the improvements obtained by optimising the hyperparameters were reduced by up to 1%. This implies that identifying the correct model is crucial because the subsequent optimisation process does not provide substantial gains.
- The metrics of both models were extremely high and similar; their rates were almost 99% for every metric. Although the Hybrid model provides better metrics than the Pure model, the difference was less than 2%. This implies that the usage of unbalanced datasets does not greatly affect classification purposes by implementing cost-sensitive techniques.

These results confirm the goal of this research on the introduction of ML techniques. Therefore, ML algorithms learn from the initial prediction and improve the quality of conflict prediction.

4.2. *ML Techniques for Regression*

ML regression techniques predict the numerical value of the minimum separation to be reached by an aircraft pair. The first step is to analyse the performance of different ML algorithms, aiming to identify the best one. The results of the different ML algorithms are shown in 'Appendix A ML results' in order to improve the reading of the paper. Both experiments agree that the three top algorithms vary between Extreme Gradient, Light Gradient and Gradient Boosting, i.e., ensemble models. Extreme Gradient provides the best results for both experiments. The models were evaluated based on stratified cross-validation techniques in order to avoid overfitting the results.

Both experiments concluded that the most influential feature is the prediction of the minimum distance. However, the influence varied from the Hybrid model (over 60%) to the Pure model (almost 100%). The rest of the variables provided influence values lower than 1% for the Pure model; as such, the Pure model was analysed without them. However, the Hybrid model requires more variables, such as the variation of the altitude and the horizontal separation among others. Figure 7 shows these results and does not confirm the similarity identified in the feature importance of the classification experiment. In addition, the analysis shows a performance decrease of more than 2%, which does not recommend removing the features.

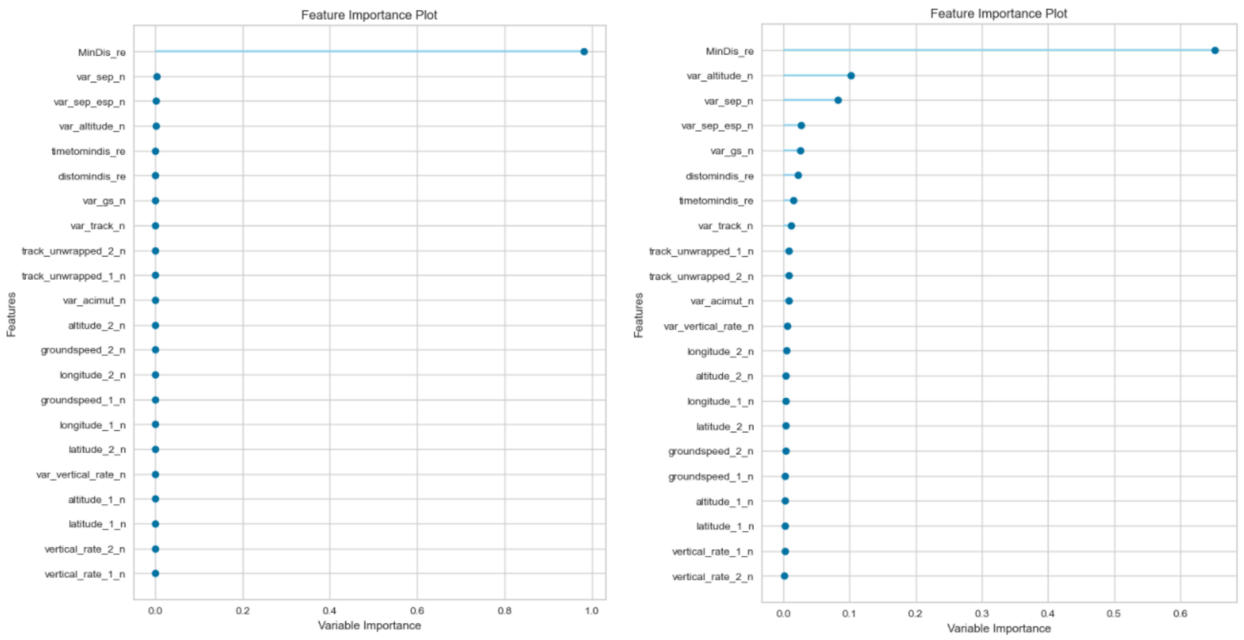

Figure 7. Feature importance of the Pure model (**left**) and the Hybrid model (**right**) for regression.

The next step is to optimise the Extreme Gradient algorithm. Only the optimisation of RMSE was studied. Table 4 shows the results of both experiments on the training set applying a fivefold cross-validation.

Table 4. Results (NM) of the regression experiments.

Experiment	RMSE	MAE	R2	RMSLE
Pure model	2.579	1.601	0.994	0.178
Hybrid model	1.473	0.992	0.934	0.235

The following conclusions were obtained:

- The optimisation process improves the initial metrics by up to 10% in some experiments. This implies that the behaviour of the Extreme Gradient model greatly depends on the optimisation process, conversely to the random forest during the classification optimisation.
- The results differed between the Pure and Hybrid models. The best results were obtained for the Hybrid model (1.5 NM). The Pure model worsens the RMSE over 1 NM, which implies an increase of over 40% compared to the Hybrid model.
- These results are different from the classification problem because cost-sensitive techniques cannot be applied to the Pure model. The higher imbalance of the Pure dataset means a clear decrease in the metrics of the regression predictors.

4.3. Analysis of ML Technique Prediction

This section analyses the performance of the predictions of the ML techniques between them in the testing set. The goal is to identify which predictions the ML models (classification and regression) tend to fail or guess right and match. Therefore, the predictions of both models will be compared considering only the Hybrid model that provided the best results. The predictions are split into four ranges in order to differentiate between SI and conflict.

Table 5 shows the allocation of the classification and regression predictions in the four ranges. The analysis concludes that the predictions fail at the 10 NM border. In case both ML predictors confirm a conflict, the conflict probability is 100%. In opposition, the probability is 100% when both predictors discard an SI. However, the samples with a predicted value of 10 to 15 NM only guess 82% right. The samples with a predicted value from 5 to 10 NM guess 97% right. It can be concluded that the model only fails in the classification when the predicted value of $MinDis$ is between 5 and 15 NM, especially if it is between 10 and 15 NM.

Table 5. Distribution of the classification and regression predictions.

Experiments	***MinDis_ML*** **= (0,5]**	***MinDis_ML*** **= (5,10]**	***MinDis_ML*** **= (10,15]**	***MinDis_ML*** **= (15,20]**
$SI\_ML = 1$	100%	97%	18%	0%
$SI\_ML = 0$	0%	3%	82%	100%

Table 6 makes the same comparison between the classification predictions and the real values of the samples.

Table 6. Distribution of the classification predictions and real values.

Experiments	***MinDis_Real*** **= (0,5]**	***MinDis_Real*** **= (5,10]**	***MinDis_Real*** **= (10,15]**	***MinDis_Real*** **= (15,20]**
$SI\_ML = 1$	100%	91%	6%	5%
$SI\_ML = 0$	0%	9%	94%	95%

Although the values are quite similar, there are some differences compared to Table 5:

- The percentages of correctly classified samples between 0 and 5 NM are the same (100%). These results confirm that training the model using SI instead of conflict provides a 100% guess of conflict.
- The model predictions fail at the 10 NM limit. The predictions from 5 to 15 NM present higher misclassification errors. However, when we use both predictions (classification and regression), the misclassification errors decrease from 5 to 10 NM by increasing the false rate from 10 to 15 NM.

The errors considered from 5 to 10 NM are samples considered from 10 to 15 NM; these errors do not impact the aircraft pairs classified as SI.

5. Conclusions

This article develops the pillars for the further development of an ATC tool based on ML techniques. This work is framed on a Technology Readiness Level 1–2 based on the European project to which it belongs. It was developed in order to analyse the viability of introducing ML techniques for conflict detection purposes by analysing their accuracy and expected errors. The introduction of ML techniques in safety-critical areas in air traffic management is very complex, and must be properly analysed in order to ensure the feasibility of these techniques. This paper develops a data-driven approach that considers the evolution of the aircraft within the airspace, and the separation evaluation along with it. The trajectories were extracted from the OpenSky Network based on ADS-B information. One of the difficulties was the generation of conflict. One limitation of using real ADS-B trajectories is that the trajectories that suffered a tactical modification by the ATC action could not be removed.

Real ADS-B trajectories were temporarily modified, considering specific aircraft sets in the same time period, to constitute the SI. In addition, historical ADS-B trajectories were used for 4DT predictions as inputs for the ML models. The introduction of 4DT predictions as an input is crucial in order to obtain such high metrics. One of the strong points of this approach is the use of real ADS-B trajectories to constitute the database. Using real trajectories means that this work could be developed for real scenarios. Moreover, the introduction of 4DT trajectories presents a twofold implication: (1) it shows the clear improvement provided by introducing this information, and (2) this methodology could be developed for other types of predictions (such as pretactical trajectories from the Network Manager or calculated by the ATC ground system).

The results confirmed that ML models can be applied to perform conflict prediction. The best algorithms for both problems are the ensemble methods. Notably, Random

Forest for classification and Extreme Gradient for regression provided the best results in the experiments. The main problem is the large dataset used to train the models, which demanded substantial computational time and computer resources. Classification techniques reach a success rate of more than 99% for SI prediction and regression techniques up to 1.5 NM in RMSE. In addition, both ML models were identified to have ensured 100% success in conflict detection (a separation infringement lower than 5 NM) combining both predictions. These results confirm the appropriateness of using 10 NM as the boundary to train the model instead of 5 NM.

Finally, this work brings to light the possibility of introducing ML techniques for conflict detection purposes, although it deals with some limitations in different areas due to its novelty. Further work should focus on: (1) the introduction of Deep Learning based on neural networks, (2) the introduction of operational and environmental variables not considered herein, and (3) the introduction of other sources for 4DT predictions as an input.

Author Contributions: Conceptualization, J.A.P.-C. and L.P.-S.; methodology, J.A.P.-C., L.P.-S. and L.S.-M.; software, F.J.S.-H. and I.R.G.; validation, F.J.S.-H. and I.R.G.; investigation, J.A.P.-C. and L.S.-M.; data curation, F.J.S.-H. and I.R.G.; writing—original draft: J.A.P.-C.; writing and review—L.P.-S., L.S.-M. and V.F.G.-C.; funding acquisition, J.A.P.-C. and V.F.G.-C. All authors have read and agreed to the published version of the manuscript.

Funding: This paper is part of a project that has received funding from the SESAR Joint Undertaking under grant agreement No 892618 under the European Union's Horizon 2020 research and innovation programme.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data is available upon request.

Acknowledgments: Notably, the authors would like to acknowledge Tomislav Radisic from FTTS and Lars Schmidt from TUBS.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. ML Results

Table A1 shows the results for the different ML classification algorithms obtained for the Pure model.

Table A1. Classification metrics for the Pure model.

Model	Accuracy	AUC	Recall	Prec.	F1	Kappa
Random Forest Classifier	0.987	0.999	0.954	0.971	0.962	0.952
Extra Trees Classifier	0.986	0.999	0.955	0.972	0.961	0.951
CatBoost Classifier	0.984	0.998	0.953	0.962	0.957	0.948
Decision Tree Classifier	0.983	0.973	0.955	0.954	0.955	0.945
Extreme Gradient Boosting	0.981	0.998	0.943	0.953	0.948	0.936
Light Gradient Boosting Machine	0.978	0.997	0.934	0.943	0.939	0.925
Gradient Boosting Classifier	0.975	0.995	0.927	0.935	0.931	0.916
Ada Boost Classifier	0.973	0.993	0.921	0.928	0.925	0.908
K Neighbors Classifier	0.970	0.987	0.910	0.923	0.916	0.898
SVM—Linear Kernel	0.966	0.000	0.902	0.909	0.906	0.884
Ridge Classifier	0.966	0.000	0.902	0.909	0.906	0.884
Linear Discriminant Analysis	0.966	0.984	0.902	0.909	0.906	0.884
Logistic Regression	0.965	0.987	0.902	0.908	0.905	0.884
Naive Bayes	0.965	0.979	0.904	0.902	0.903	0.882
Quadratic Discriminant Analysis	0.960	0.980	0.906	0.877	0.891	0.866

Table A2 shows the results for the different ML classification algorithms obtained for the Hybrid model.

Table A2. Classification metrics for the Hybrid model.

Model	Accuracy	AUC	Recall	Prec.	F1	Kappa
Random Forest Classifier	0.984	0.998	0.984	0.989	0.985	0.952
Extra Trees Classifier	0.982	0.989	0.982	0.988	0.981	0.949
Decision Tree Classifier	0.972	0.967	0.979	0.979	0.979	0.935
CatBoost Classifier	0.964	0.994	0.973	0.974	0.974	0.918
Extreme Gradient Boosting	0.954	0.991	0.966	0.967	0.966	0.894
K Neighbors Classifier	0.951	0.987	0.965	0.963	0.964	0.886
Light Gradient Boosting Machine	0.943	0.987	0.958	0.959	0.959	0.870
Gradient Boosting Classifier	0.935	0.982	0.954	0.952	0.953	0.851
Ada Boost Classifier	0.926	0.976	0.941	0.950	0.946	0.831
Logistic Regression	0.894	0.949	0.914	0.930	0.922	0.759
Naive Bayes	0.894	0.943	0.905	0.937	0.921	0.759
SVM—Linear Kernel	0.894	0.000	0.902	0.941	0.921	0.762
Ridge Classifier	0.894	0.000	0.902	0.941	0.921	0.762
Linear Discriminant Analysis	0.894	0.942	0.902	0.941	0.921	0.762
Quadratic Discriminant Analysis	0.893	0.929	0.905	0.936	0.920	0.757

Table A3 shows the results for the different ML regression algorithms obtained for the Pure model.

Table A3. Regression metrics for the Pure model.

Model	MAE	RMSE	R2	RMSLE
Extreme Gradient Boosting	1.832	2.954	0.991	0.199
Light Gradient Boosting Machine	1.991	3.173	0.990	0.209
Gradient Boosting Regressor	2.179	3.464	0.988	0.224
Linear Regression	2.313	3.763	0.986	0.254
Ridge Regression	2.313	3.763	0.986	0.254
Bayesian Ridge	2.313	3.763	0.986	0.254
Least Angle Regression	2.315	3.765	0.986	0.255
Orthogonal Matching Pursuit	2.293	3.777	0.986	0.256
Huber Regressor	2.263	3.781	0.986	0.251
Decision Tree Regressor	2.024	3.825	0.986	0.236
Lasso Regression	2.506	3.921	0.985	0.269
K Neighbors Regressor	3.975	5.720	0.968	0.368
Passive Aggressive Regressor	4.145	5.549	0.969	0.382
AdaBoost Regressor	7.510	9.111	0.919	0.637
Elastic Net	8.130	10.242	0.897	0.605
Lasso Least Angle Regression	26.181	31.984	-0.000	1.082

Table A4 shows the results for the different ML regression algorithms obtained for the Hybrid model.

Table A4. Regression metrics for the Hybrid model.

Model	MAE	RMSE	R2	RMSLE
Extreme Gradient Boosting	1.281	1.881	0.892	0.294
Light Gradient Boosting Machine	1.371	2.014	0.876	0.312
Gradient Boosting Regressor	1.548	2.216	0.850	0.336
Decision Tree Regressor	1.174	2.231	0.848	0.336
K Neighbors Regressor	1.883	2.560	0.801	0.395
AdaBoost Regressor	2.299	2.835	0.755	0.419
Linear Regression	2.489	3.393	0.650	0.467
Least Angle Regression	2.490	3.393	0.650	0.467
Ridge Regression	2.490	3.393	0.650	0.467
Bayesian Ridge	2.490	3.393	0.650	0.467
Orthogonal Matching Pursuit	2.493	3.410	0.646	0.466
Huber Regressor	2.265	3.665	0.591	0.425
Lasso Regression	2.892	3.668	0.590	0.525
Elastic Net	3.218	3.904	0.536	0.576
Passive Aggressive Regressor	3.618	4.782	0.302	0.629
Lasso Least Angle Regression	4.799	5.732	-0.000	0.770

References

1. SESAR Joint Undertaking SESAR 2020 Concept of Operations. 2017. Available online: https://ec.europa.eu/research/participants/documents/downloadPublic?documentIds=080166e5b6d2b912&appId=PPGMS (accessed on 25 November 2021).
2. SESAR Joint Undertaking. *European Atm Master Plan—Digitalising Europe's Aviation Infraestructure*; SESAR Joint Undertaking: Brussels, Belgium, 2019.
3. ICAO. *Air Traffic Management—Doc 4444*; ICAO: Montreal, QC, Canada, 2012.
4. ICAO. *Doc 9689-AN/953—Manual on Airspace Planning Methodology for the Determination of Separation Minima*; ICAO: Montreal, QC, Canada, 1998; pp. 1–205.
5. Reich, P.G. Analysis of long-Range air traffic systems—Separation Standards I. *J. Inst. Navig.* **1966**, *50*, 436–477. [CrossRef]
6. Reich, P.G. Analysis of long-range air Separation Standards—II. *J. Navig.* **1966**, *19*, 169–186. [CrossRef]
7. Kim, S.H. Conflict Risk Assessment of Structured and Unstructured Traffic of Small Unmanned Aircraft Systems. In Proceedings of the 2018 Aviation Technology, Integration, and Operations Conference, Atlanta, GA, USA, 25–29 June 2018; Volume 15, pp. 25–29.
8. Pérez-Castán, J.A.; Gómez Comendador, F.; Rodríguez-Sanz, Á.; Arnaldo Valdés, R.M. Conflict-risk assessment model for continuous climb operations. *Aerosp. Sci. Technol.* **2019**, *84*, 812–820. [CrossRef]
9. Pérez-Castán, J.A.; Gómez Comendador, F.; Rodríguez-Sanz, A.; Armas Cabrera, I.; Torrecilla, J. RPAS conflict-risk assessment in non-segregated airspace. *Saf. Sci.* **2019**, *111*, 7–16. [CrossRef]
10. Siddiqee, W. A mathematical model for predicting the number of potential conflict situations at intersecting air routes. *Transp. Sci.* **1973**, *7*, 571–577. [CrossRef]
11. Sunil, E.; Ellerbroek, J.; Hoekstra, J.M.; Maas, J. Three-dimensional conflict count models for unstructured and layered airspace designs. *Transp. Res. Part C Emerg. Technol.* **2018**, *95*, 295–319. [CrossRef]
12. Netjasov, F.; Janic, M. A review of research on risk and safety modelling in civil aviation. *J. Air Transp. Manag.* **2008**, *14*, 213–220. [CrossRef]
13. Krozel, J.; Peters, M.E.; Hunter, G. *Conflict Detection and Resolution for Future Air Transportation Management—TR97138-01*; NASA Ames Research Center: Moffett Field, CA, USA, 1997.
14. Paielli, R.A.; Erzberger, H. Trajectory Specification for Terminal Air Traffic: Pairwise Conflict Detection and Resolution. In Proceedings of the 17th AIAA Aviation Technology, Integration, and Operations Conference, Denver, CO, USA, 5–9 June 2017.
15. Kuchar, J.K.; Yang, L.C. Conflict detection and resolution, air traffic control, alerting systems, warning systems. *IEEE Trans. Intell. Transp. Syst.* **2000**, *1*, 179–189. [CrossRef]
16. Erzberger, H.; Lauderdale, T.A.; Chu, Y.-C. Automated conflict resolution, arrival management, and weather avoidance for air traffic management. *Proc. Inst. Mech. Eng. Part G J. Aerosp. Eng.* **2012**, *226*, 930–949. [CrossRef]
17. Pérez-Castán, J.A.; Comendador, F.G.; Rodríguez-Sanz, Á.; Barragán, R.; Arnaldo-Valdés, R.M. Design of a conflict-detection air traffic control tool for the implementation of continuous climb operations: A case study at Palma TMA. *Proc. Inst. Mech. Eng. Part G J. Aerosp. Eng.* **2019**, *233*, 4839–4852. [CrossRef]
18. Tang, H.; Robinson, J.E.; Denery, D.G. Tactical Conflict Detection in Terminal Airspace. *J. Guid. Control Dyn.* **2011**, *34*, 403–413. [CrossRef]
19. EAAI-HLG. *The FLY AI Report Demystifying and Accelerating AI in Aviation/ATM*; EAAI-HLG: Bournemouth, UK, 2020.
20. Geron, A. *Hands-On Machine Learning with Scikit-Learn and TensorFlow*; O'Reilly Media, Inc.: Newton, MA, USA, 2017; ISBN 9781491962299.
21. Mitchell, T.M. *Machine Learning*; McGraw-Hill Science/Engineering/Math: New York, NY, USA, 1997; ISBN 0-07-042807-07.
22. Alligier, R.; Gianazza, D.; Durand, N. Machine Learning and Mass Estimation Methods for Ground-Based Aircraft Climb Prediction. *IEEE Trans. Intell. Transp. Syst.* **2015**, *16*, 3138–3149. [CrossRef]
23. Alligier, R.; Gianazza, D. Learning aircraft operational factors to improve aircraft climb prediction: A large scale multi-airport study. *Transp. Res. Part C Emerg. Technol.* **2018**, *96*, 72–95. [CrossRef]
24. Fernández, E.C.; Cordero, J.M.; Vouros, G.; Pelekis, N.; Kravaris, T.; Georgiou, H.; Fuchs, G.; Andrienko, N.; Andrienko, G.; Casado, E.; et al. DART: A machine-learning approach to trajectory prediction and demand-capacity balancing. *SESAR Innov. Days* **2017**. Available online: https://www.sesarju.eu/sites/default/files/documents/sid/2017/SIDs_2017_paper_65.pdf (accessed on 25 November 2021).
25. Sridhar, B. Applications of machine learning techniques to aviation operations: Promises and challenges. In Proceedings of the 2020 International Conference on Artificial Intelligence and Data Analytics for Air Transportation (AIDA-AT), Singapore, 3–4 February 2020; pp. 1–12.
26. Galar, M.; Fernandez, A.; Barrenechea, E.; Bustince, H.; Herrera, F. A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **2012**, *42*, 463–484. [CrossRef]
27. Cho, D.; Yoo, C.; Im, J.; Cha, D.H. Comparative Assessment of Various Machine Learning-Based Bias Correction Methods for Numerical Weather Prediction Model Forecasts of Extreme Air Temperatures in Urban Areas. *Earth Sci.* **2020**, *7*, 1–18. [CrossRef]
28. Haupt, S.E.; Cowie, J.; Linden, S.; McCandless, T.; Kosovic, B.; Alessandrini, S. Machine learning for applied weather prediction. In Proceedings of the IEEE 14th International Conference on e-Science (e-Science), Amsterdam, The Netherlands, 29 October–1 November 2018; pp. 276–277.

29. Chaimatanan, S.; Delahaye, D.; Mongeau, M. Aircraft 4D trajectories planning under uncertainties. In Proceedings of the IEEE Symposium Series on Computational Intelligence, Cape Town, South Africa, 8–10 December 2015; pp. 51–58.
30. Persiani, C.A.; Bagassi, S. Route planner for unmanned aerial system insertion in civil non-segregated airspace. *Proc. Inst. Mech. Eng. Part G J. Aerosp. Eng.* **2013**, *227*, 687–702. [CrossRef]
31. Pérez-Castán, J.A.; Rodríguez-Sanz, Á.; Pérez Sanz, L.; Arnaldo Valdés, R.M.; Gomez Comendador, V.F.; Greatti, C.; Serrano-Mira, L. Probabilistic Strategic Conflict Management for 4D Trajectories in Free Route Airspace. *Entropy* **2020**, *22*, 159. [CrossRef] [PubMed]
32. Alam, S.; Shafi, K.; Abbass, H.A.; Barlow, M. An ensemble approach for conflict detection in Free Flight by data mining. *Transp. Res. Part C Emerg. Technol.* **2009**, *17*, 298–317. [CrossRef]
33. The OpenSky Network. The OpenSky Network API. Available online: https://opensky-network.org/apidoc/index.html (accessed on 7 October 2020).
34. Boeing Commercial Airplanes Statistical Summary of Commercial Jet Airplane Accidents World Wide Operations 1959–2018. 2019. Available online: http://www.boeing.com/commercial/safety/investigate.html (accessed on 25 November 2021).
35. Karney, C.F.F. Geographiclib. Available online: https://geographiclib.sourceforge.io/html/python (accessed on 10 September 2020).
36. AISA Concept of Operations for AI Situational Awareness. 2021. Available online: https://aisa-project.eu/downloads/AISA_D2.1_CONOPS.pdf (accessed on 25 November 2021).
37. Buitinck, L.; Louppe, G.; Blondel, M.; Pedregosa, F.; Mueller, A.; Grisel, O.; Niculae, V.; Prettenhofer, P.; Gramfort, A.; Grobler, J.; et al. API design for machine learning software: Experiences from the scikit-learn project. *arXiv* **2013**, arXiv:1309.0238.
38. Ali, M. PyCaret. Available online: https://pycaret.org (accessed on 30 November 2020).
39. Varoquaux, G.; Buitinck, L.; Louppe, G.; Grisel, O.; Pedregosa, F.; Mueller, A. Scikit-learn. *GetMobile Mob. Comput. Commun.* **2015**, *19*, 29–33. [CrossRef]