

PAPER • OPEN ACCESS

## Machine Learning classification techniques applied to static air traffic conflict detection

To cite this article: Javier A. Pérez-Castán *et al* 2022 *IOP Conf. Ser.: Mater. Sci. Eng.* **1226** 012019

View the [article online](#) for updates and enhancements.

You may also like

- [The Researches on Planning of Airline Network under Flexible Airspace Usage](#)  
FangYue Guo, JinLiang Chen and Wenqian Zhang
- [Cross-border Free Route Airspace concept and its impact on flight efficiency improvement](#)  
Andrzej Majka and Aleksandra Pasich
- [Design of general aviation airspace planning and management system based on Google Earth](#)  
Jing Mou

# Machine Learning classification techniques applied to static air traffic conflict detection

Javier A. Pérez-Castán<sup>\*1</sup>, L. Pérez-Sanz<sup>1</sup>, J. Bowen-Varela<sup>1</sup> and L. Serrano-Mira<sup>1</sup>, Tomislav Radisic<sup>2</sup> and Thomas Feuerle<sup>3</sup>

<sup>1</sup>Escuela Técnica Superior de Ingeniería Aeronáutica y del Espacio (ETSIAE), Universidad Politécnica de Madrid (UPM), Plaza Cardenal Cisneros, 28040 Madrid, Spain

<sup>2</sup>University of Zagreb, Faculty of Transport and Traffic Sciences, Republic of Croatia

<sup>3</sup>Institute of Flight Guidance, Technical University of Braunschweig, Braunschweig, Germany

\* Correspondence: javier.perez.castan@upm.es; +34-910-67-59-36

**Abstract.** This article evaluates Machine Learning (ML) classification techniques applied to air-traffic conflict detection. The methodology develops a static approach in which the conflict prediction is performed when an aircraft pierces into the airspace. Conflict detection does not evaluate separation infringements but a Situation of Interest (SI). An aircraft pair constitutes a SI when it is expected to get with a horizontal separation between both aircraft closer than 10 Nautical Miles (NM) and a vertical separation closer than 1000 feet (ft). Therefore, the ML predictor classifies aircraft pairs between SI or No SI pairs. Air traffic information is extracted from The OpenSky Network that provides ADS-B trajectories. ADS-B trajectories do not offer enough SI samples to be evaluated. Hence, the authors performed simulations varying the entry time of the trajectories to the airspace within the same time period. The methodology was applied to a portion of Switzerland airspace, and simulations reached a 5% rate of SI samples. Cost-sensitive techniques were used to handle the strong imbalance of the database. Two experiments were performed: the Pure model considered the whole database, and the Hybrid model considered aircraft pairs that intersect horizontally closer than 20 NM and vertically lower than 1000 ft. The Hybrid model provided the best results achieving 72% recall, representing the success rate of Missed alerts and 82% accuracy, which means the whole predictions' success rate.

## 1. Introduction

The expected increase in air traffic demand for the following years is pushing the Air Traffic Management (ATM) to investigate in new technologies. This fact means two problems for the ATM system: air traffic demand and new technologies. The increase in air traffic demand is not a novel issue but is strongly limited by ATM capacity. Single European Sky in Air Traffic Management Research (SESAR) is the European effort to deal with the current and future ATM needs [1].

Safety is one of the ATM pillars due to the low acceptance of risk by the aviation community. One of the metrics to consider is the separation infringements in the airspace. A conflict is a situation where a loss of separation minima can occur [2]. Conflict detection (CD) or identification has been analysed



based on different timeframes. Strategic CD analyses the conflict risk based on air traffic flows and airspace design information [3]–[5]. Tactical CD analyses the separation infringements based on Four-Dimensional Trajectory (4DT) predictions [6]–[8]. This work takes advantage of the concept of Situation of Interest (SI). SI is similar to conflict but considering a pre-defined separation minimum larger than the current separation minima.

Machine Learning (ML) is one of the most promising technologies based on novel technologies' data-driven approaches. EASA defines ML as "algorithms whose performance improve as they are exposed to data" [9]. The ML algorithms' goal is to learn from past situations the intrinsic features that characterise those situations to perform predictions to new equivalent samples. ML has been applied in different ambits for aviation. One of the most extended application may be for trajectory prediction [10]–[12], although it has been used for air traffic flow analysis or safety analysis [13], [14]. Conflict detection is an emerging topic that has been barely tackled based on ML techniques [15], [16].

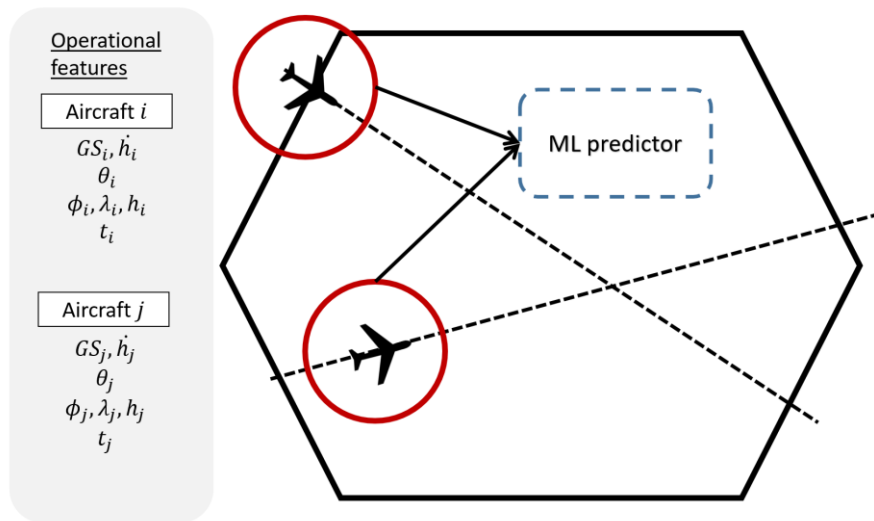
This work aims to analyse the feasibility of using ML techniques for conflict detection for en-route airspace. This work considers a static approach to classify aircraft pairs into separation infringements or not. Section 2 describes the static method to predict SI. Section 3 explains the method for building the database to get enough number of SI. Section 4 provides information about the process followed with the ML techniques, and the results are shown in Section 5. Finally, the primary conclusions are summarised.

## 2. Static conflict detection based on SI

This work focuses on the application of ML techniques for identifying separation infringements in en-route airspace. This work analyses the Situation of Interest (SI) in the airspace between aircraft pairs. One SI can be defined as an aircraft pair that will intersect, infringing horizontal and vertical pre-defined separations. This pre-defined separation may not match with the current separation minima. Herein, an aircraft pair constitutes a SI when it is expected to cross with a horizontal separation closer than 10 Nautical Miles (NM) and a vertical separation closer than 1000 feet (ft). This pre-defined separation is considered because ML predictions have unknown uncertainties that are difficult to quantify currently.

The approach is to perform a prediction when one aircraft pierces horizontally an airspace volume. This fix or static snapshot provides information with the rest of the aircraft located within that airspace volume. To classify the aircraft pairs into SI or No SI, the primary variable is the evolution of the separation  $s(t)$ . Based on computational processing, the separation evolves throughout the timestamps ( $t$ ) and the location of each aircraft (longitude, latitude and altitude). Once the aircraft pair reaches the minimum separation defined ( $\min s(t)$ ), this aircraft pair is classified as SI and no SI depending on the conjunction of horizontal and vertical distances. The occurrence or not of SI can be characterised based on specific features of an aircraft pair's operational and geometrical features.

Therefore, this approach aims to analyse ML techniques' feasibility to predict SI when one aircraft pierces into the airspace based on the operational features. This approach focuses on planner air traffic controller and can help him with conflict search and situational awareness. Figure 1 shows a scheme of this concept.



**Figure 1.** Concept of static conflict detection based on SI.

### 3. Database generation

In this work, the data source considered is ADS-B trajectories. The OpenSky Network provides records of ADS-B aircraft trajectories that flew throughout European airspace [17]. ADS-B data give information on the status of the aircraft at each timestamp. Although ADS-B provides many information, just a small set of variables have been used in this work:

- Position: longitude ( $\lambda$ ), latitude ( $\phi$ ) and altitude ( $h$ ).
- Velocity: ground speed ( $GS$ ) and vertical rate ( $\dot{h}$ ).
- Heading ( $\theta$ ).

This information belongs to each aircraft at each timestamp ( $t$ ).

Mainly, this work focuses on LSAZM567 airspace in Switzerland with vertical boundaries from FL355 to FL660. The ADS-B database was constituted for 15 days and extracted from The OpenSky Network from the AIRAC cycle of June 2019. 7116 trajectories and 4612757 samples (each sample represent an ADS-B timestamp) built the database. Once the trajectories were extracted, it was performed the following cleaning process:

- Filtering of data duplicity: it has been removed duplicated trajectories.
- Erroneous paths: it has been removed those trajectories with more than 10 ADS-B erroneous data.
- NaN or missed data: missed data have been filled with the average value based on the prior and subsequent ADS-B data.
- Clustering per day and hours: it has been filtered the database to cluster the trajectories based on an operational day and nine time-periods. This clustering has been performed to reduce the number of trajectories and computational requirements.

The next step was to build the database based on aircraft pairs. Each set of trajectories clustered per day and hour has performed a process of aircraft-pair generation. The trajectories were modified temporarily to pierce into the airspace at the same time period. This trajectory customisation generated enough SI to be considered statistically. The separation pair's evolution was evaluated for each aircraft pair by considering new variables based on relative variables such as relative position, velocity, and heading. This information was calculated for each timestamp and each aircraft pair:

- Initial relative position: Longitudinal and vertical separation and course between positions.

- Initial relative velocity: variation of ground speed and vertical rate.
- Initial relative heading.

The initial values were considered in this work because they corresponded to the instant one aircraft pierces into the airspace while the other aircraft's ADS-B information belonged to the aircraft within the airspace. Moreover, one new feature is required to be added to the database: the SI feature. The SI feature labels each aircraft pair with a Boolean variable: 1 (SI) and 0 (No SI). This feature is the result of the aircraft pair trajectories and represents the variable that ML will predict. This feature means the label that is used for supervised ML algorithms. Therefore, 19 features constitute the ML database: 6 relative variables, 12 ADS-B variables (6 per aircraft) and 1 label. The duration of the simulation performance was around 1200 hours. The database volume is about 520 MB and contains more than 500000 samples. Results confirmed a strong imbalance in the database because just 5% of the samples were SI.

#### 4. ML approach

This work follows a typical ML approach from the data preparation until the ML optimisation process. The data pre-processing adequates the raw data into workable data for ML algorithms. Herein, the following pre-processing activities have been applied:

- Training and validation set: The dataset is split into the training and the validation dataset. The validation 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 validation set.
- Normalisation: Normalisation rescales numeric columns' values in the dataset following a normal distribution in this work. It does not distort differences in the ranges of values.
- Shuffling: It distributes the samples randomly into the training and validation datasets.
- Stratification: Apart from randomly distributing of the samples, the stratification process spreads the samples keeping the SI statistical distributions.
- Cross-validation (CV): It is a process to avoid overfitting on the ML models [18]. The training set is divided into k folds, one is considered the validation set, and the rest are regarded as the training set. This process is repeated every k fold, and the metrics are calculated based on the mean and the standard deviation.

Once the data is pre-processed, the ML process follows the following steps:

1. Analysis of ensemble ML algorithms. Ensemble ML algorithms provide the best results for unbalanced datasets [19]. Decision Tree, Random Forest, and Extra Trees are the models evaluated. The authors recommend seeing the work [20] for mathematical details. The goal is to identify the best algorithm based on classification metrics.
2. Feature selection. The feature selection analyses the influence of the different features on the ML model. The feature selection is performed based on graphical analysis, and the Recursive Feature Elimination (RFE) is based on feature permutation [18]. Features without influence on the ML model are removed from the database.
3. ML algorithm optimisation. This process aims to optimise the ML performance based on a grid search of hyperparameters. Hyperparameters are the algorithm's settings that can be adjusted to optimise the model performance. The hyperparameters are selected in advance to

training the ML algorithm. Moreover, the metric to optimise depends on the type of problem: classification or regression.

As previously mentioned, the CD is a typical unbalanced problem in the tactical phase of the ATM, i.e., the number of SI samples compared with no SI samples are not represented equally. Unbalanced problems cannot be evaluated based on the accuracy metric [21]. An ML model may present a higher level of accuracy by predicting every sample as the majority class without considering the impact on the minority class. F1, recall, and precision focus on the minority class and are metrics more suitable for this problem [22]. However, there are several techniques to tackle unbalanced problems [19]. This work introduces cost-sensitive techniques. Cost-sensitive techniques potentially improve the metrics for unbalanced problems because they modify the impact of misclassifying the minority-class samples. Finally, this work has been performed in several computers Intel® Core™ i5-6600 CPU @ 3.30GHz, RAM 8,00 GB, 64 bits. ML models have been developed in Python®. Scikit-learn is the ML open-source library used for Python [18], [23]. PyCaret is another open-source library that eases ML algorithms' implementation in the Python environment [24].

## 5. Results

Two experiments were performed: the Pure model considered the whole database, and the Hybrid model considered aircraft pairs that intersect horizontally closer lower than 20 NM and vertically closer than 1000 ft. The goal of the Hybrid model is to handle the strong imbalance of the database. Table 1 shows the number of samples for both experiments; in parenthesis, the SI/no SI samples rate.

**Table 1.** Number of samples for each experiment.

	Training set	Validation set	Test set
<b>Pure model (5/95)</b>	792033	339444	125720
<b>Hybrid model (35/65)</b>	112463	48199	17852

The first step was to analyse the performance of the ensemble models. As commented before, the metrics to consider were F1, recall and precision ahead of accuracy because of the strong imbalance. The Decision Tree provided the best results for the Pure model and Random Forest for the Hybrid model. The models were evaluated based on a stratified CV to avoid overfitting.

The second step was about feature selection. The results were similar for both models. The most influential features were initial azimuth, initial separation, the variation of altitude and tracks. They encompassed almost 50% of the ML importance. RFE confirmed that vertical rate variables should be discarded due to their low influence. Hence, vertical rate variables were removed from the database, and the results were quite similar, 1% lower than the initial results.

The last step was the ML optimisation process by optimising the F1 metric. F1 metric was selected because it is the metric that leverages the behaviour of recall and precision at the same time. A CV grid search was implemented to seek the best hyperparameters regarding the number of estimators, complexity parameter, maximum depth, the maximum number of features, minimum samples of leaf and criterion. Besides, the hyperparameter class weight was considered cost-sensitive techniques that penalises misclassification errors from the minority class. Table 2 shows the results of the non-optimised and optimised models by applying a 5-fold CV.

**Table 2.** Metrics for the ML classification models.

Experiment	Model	Optimiser	Accuracy	AUC	Recall	Precision	F1
<b>Pure model</b>	Non-optimised Model	-	0.958	0.780	0.570	0.585	0.577
	Optimised Model	F1	0.961	0.814	0.651	0.587	0.617
<b>Hybrid model</b>	Non-optimised Model	-	0.813	0.886	0.606	0.800	0.690
	Optimised Model	F1	0.823	0.798	0.715	0.760	0.736

Results confirmed that optimised models provided better results than non-optimised models. However, these improvements were reduced up to 5%. The identification of the correct ensemble model was crucial. Besides, this 5% of improvements depended highly on cost-sensitive techniques. Cost-sensitive techniques meant about 3% improvements on metrics. The Hybrid model provided better results than the Pure model. The Pure model metrics presented a very high accuracy (over 95%) but low values for recall precision and F1 (around 60%). This performance was quite similar to the expected results for unbalanced problems. However, the Hybrid model improved those values with rates up to 75% by decreasing the accuracy by up to 80%. These results confirmed that the larger the imbalance, the worse the minority-class metrics. Moreover, these initial results are positive to continue researching this area. However, the use of these predictors based on ML techniques cannot be used in safety-critical systems as the conflict detection module in ATC.

## 6. Conclusions

This work presents a method to predict Situations of Interest (SI). The proposed model includes the application of ML techniques to perform the predictions. The methodology developed a static approach in which the CD makes a prediction when an aircraft pierces into an airspace volume. The model does not evaluate separation minima infringements but SI. The first step was to perform simulations to obtain enough data samples for SI in the database. These simulations were performed based on temporary modifications of ADS-B trajectories. The results were about 5% SI samples, which implied a strong imbalance in the database. This imbalance was tackled based on cost-sensitive techniques that allow penalising misclassification errors from the minority class. Six ensemble models were analysed, providing the Random Forest with the best results. Feature selection concluded that vertical rate variables did not influence the model. This is one characteristic due to the upper levels considered in LSAZM567 airspace. Two experiments were performed to tackle the unbalanced database. The Pure model considered the whole database, and the Hybrid model considered aircraft pairs that intersect horizontally closer than 20 NM and vertically closer than 1000 ft. The Hybrid model provided the best results. The Pure model metrics presented a very high accuracy (over 95%) but low values for recall precision and F1 (around 60%). The Hybrid model improved those values with rates up to 75% by decreasing the accuracy by up to 80%. These results confirmed that the larger the imbalance, the worse the minority-class metrics. Future work will focus on integrating trajectory predictions to improve CD and assessing the ML performance.

## 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.

## References

- [1] SESAR Joint Undertaking, "EUROPEAN ATM MASTER PLAN - Digitalising Europe's aviation Infrastructure," 2020. doi: 10.2829/10044.
- [2] ICAO, "Air Traffic Management - Doc 4444," 2012.
- [3] S. Alam, K. Shafi, H. A. Abbass, and M. Barlow, "An ensemble approach for conflict detection in Free Flight by data mining," *Transp. Res. Part C Emerg. Technol.*, vol. 17, no. 3, pp. 298–317, 2009, doi: 10.1016/j.trc.2008.12.002.
- [4] J. A. Pérez-Castán *et al.*, "Probabilistic Strategic Conflict - Management for 4D Trajectories in Free - Route Airspace," *Entropy*, vol. 22, pp. 1–17, 2020, doi: 10.3390/e22020159.
- [5] F. Netjasov, "Framework for airspace planning and design based on conflict risk assessment

- Part 1 : Conflict risk assessment model for airspace strategic planning,” *Transp. Res. Part C Emerg. Technol.*, vol. 24, pp. 190–212, 2012, doi: 10.1016/j.trc.2012.03.002.
- [6] R. A. Paielli and H. Erzberger, “Trajectory Specification for Terminal Air Traffic : Pairwise Conflict Detection and Resolution,” in *17th AIAA Aviation Technology, Integration, and Operations Conference*, 2017, no. 1–19, doi: 10.2514/6.2017-4256.
- [7] J. A. Pérez-Castán, F. G. Comendador, Á. Rodríguez-Sanz, R. Barragán, and R. M. Arnaldo-Valdés, “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.*, vol. 233, no. 13, 2019, doi: 10.1177/0954410019831198.
- [8] H. Tang, J. E. Robinson, and D. G. Denery, “Tactical Conflict Detection in Terminal Airspace,” *J. Guid. Control. Dyn.*, vol. 34, no. 2, pp. 403–413, 2011, doi: 10.2514/1.51898.
- [9] EASA, “Artificial Intelligence Roadmap A human-centric approach to AI in aviation,” 2020. [Online]. Available: <https://www.easa.europa.eu/sites/default/files/dfu/EASA-AI-Roadmap-v1.0.pdf>.
- [10] R. Alligier, D. Gianazza, and N. Durand, “Machine Learning and Mass Estimation Methods for Ground-Based Aircraft Climb Prediction,” *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 6, pp. 3138–3149, 2015, doi: 10.1109/TITS.2015.2437452.
- [11] R. Alligier and D. Gianazza, “Learning aircraft operational factors to improve aircraft climb prediction: A large scale multi-airport study,” *Transp. Res. Part C Emerg. Technol.*, vol. 96, no. July, pp. 72–95, 2018, doi: 10.1016/j.trc.2018.08.012.
- [12] E. C. Fernández *et al.*, “DART: A machine-learning approach to trajectory prediction and demand-capacity balancing,” *SESAR Innov. Days*, no. iii, 2017, [Online]. Available: [https://www.sesarju.eu/sites/default/files/documents/sid/2017/SIDs\\_2017\\_paper\\_65.pdf](https://www.sesarju.eu/sites/default/files/documents/sid/2017/SIDs_2017_paper_65.pdf).
- [13] B. Sridhar, “Applications of machine learning techniques to aviation operations: Promises and challenges,” *2020 Int. Conf. Artif. Intell. Data Anal. Air Transp. AIDA-AT 2020*, pp. 1–12, 2020, doi: 10.1109/AIDA-AT48540.2020.9049205.
- [14] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera, “A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches,” in *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, 2012, vol. 42, no. 4, pp. 463–484, doi: 10.1109/TSMCC.2011.2161285.
- [15] Z. Wang, M. Liang, D. Delahaye, and W. Wu, “Learning real trajectory data to enhance conflict detection accuracy in closest point of approach problem,” *13th USA/Europe Air Traffic Manag. Res. Dev. Semin. 2019*, 2019, [Online]. Available: <https://www.semanticscholar.org/paper/Learning-Real-Trajectory-Data-to-Enhance-Conflict-Wang-Liang/797957c9992f5176bc23f870f2156b2985ab0408>.
- [16] Z. Wang, M. Liang, and D. Delahaye, “Data-driven conflict detection enhancement in 3d airspace with machine learning,” *2020 Int. Conf. Artif. Intell. Data Anal. Air Transp. AIDA-AT 2020*, no. February, 2020, doi: 10.1109/AIDA-AT48540.2020.9049180.
- [17] T. O. Network, “The OpenSky Network API,” 2017. <https://opensky-network.org/apidoc/index.html> (accessed Oct. 07, 2020).
- [18] G. Varoquaux, L. Buitinck, G. Louppe, O. Grisel, F. Pedregosa, and A. Mueller, “Scikit-learn,” *GetMobile Mob. Comput. Commun.*, vol. 19, no. 1, pp. 29–33, 2015, doi: 10.1145/2786984.2786995.
- [19] G. Weiss, K. McCarthy, and B. Zabar, “Cost-sensitive learning vs. sampling: Which is best for handling unbalanced classes with unequal error costs?,” in *DMIND*, 2007, pp. 1–7, [Online]. Available: <http://storm.cis.fordham.edu/~gweiss/papers/dmin07-weiss.pdf>.
- [20] A. Geron, *Hands-On Machine Learning with Scikit-Learn and TensorFlow*. O’Reilly Media, Inc., 2017.
- [21] M. F. Uddin, “Addressing Accuracy Paradox Using Enhanced Weighted Performance Metric in Machine Learning,” in *ITT 2019 - Information Technology Trends: Emerging Technologies Blockchain and IoT*, 2019, pp. 319–324, doi: 10.1109/ITT48889.2019.9075071.

- [22] C. Goutte and E. Gaussier, “A Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation,” *Lect. Notes Comput. Sci.*, vol. 3408, pp. 345–359, 2005, doi: 10.1007/978-3-540-31865-1\_25.
- [23] L. Buitinck *et al.*, “API design for machine learning software: experiences from the scikit-learn project,” pp. 1–15, 2013, [Online]. Available: <http://arxiv.org/abs/1309.0238>.
- [24] M. Ali, “PyCaret,” 2020. <https://pycaret.org/> (accessed Nov. 30, 2020).