“Time in Advance Go-Around Predictions for Decision Support in Air Traffic Management”

P. Hernández, L. Beller, C. Argerich and P. Koppitz

This is a copy of the author’s accepted version of a paper published in the proceedings of the 41st Digital Avionics Systems Conference (DASC).

The final published version is available at DOI: 10.1109/DASC55683.2022.9925848

This project has received funding from the SESAR Joint Undertaking (JU) under H2020-Call: H2020-SESAR-2019-2, Grant Agreement number: 892919. The JU receives support from the European Union’s Horizon 2020 research and innovation programme and the SESAR JU members other than the Union.

Time in Advance Go-Around Predictions for Decision Support in Air Traffic Management

1st Pablo Hernández
Innaxis Research Institute
Madrid, Spain
ph@innaxis.aero

2nd Lukas Beller
Technical University of Munich
Munich, Germany
Lukas.beller@tum.de

3rd Clara Argerich
Innaxis Research Institute
Madrid, Spain
ca@innaxis.aero

4th Phillip Koppitz
Technical University of Munich
Munich, Germany
phillip.koppitz@tum.de

***Abstract*—Maintaining the safety and cost-effectiveness of air transport operations, while increasing capacity, will push the next generation of ATM systems towards digitization. In the medium term, a digitized system in the human-managed ATM environment will be able to provide reliable predictive analytics based on automated information processing, thus providing decision support to human operators. This paper details the development of a machine learning based solution for go-around predictions, exemplified at two major European airports. Go-arounds are standard, high workload, procedures by which an aircraft in the final approach phase can safely discontinue the approach. The proposed machine learning solution is aimed at increasing safety levels in airport operations by enhancing air traffic controllers situational awareness and helping them better plan and adapt to go-around scenarios. This work leverages on more than two years worth of ADS-B and successfully uses oversampling technique to combat the high imbalanced in the data. In addition, we performed a benchmark study with a selection of the most common classification models. The final type of model selected was LightGBM for which a feasibility study for predictions at 2NM, 4NM, 6NM, and 8NM distance from the runway threshold was performed. The results for both airports showed that although the models' recall decreases with the distance from the threshold, we were able to maintain a high prediction precision between 90% at 2NM to 80% at 8NM. Finally, a study of the explainability of the probabilistic predictions was carried out by evaluating the most important features of the models.**

***Index Terms*—Go-around, safety, machine learning, lightGBM, SMOTE, explainability**

I. Introduction

The next generation of Air Traffic Management (ATM) systems are pushed more and more towards digitalization. New developed digitized systems will mainly rely on big data technologies and the fusion of data from multiple sources. This will be reinforced by the use of Artificial Intelligence (AI) algorithms that, through the use of this data, will be able to provide accurate prediction in a multitude of scenarios. The first step in this new era of digitalization is going to be the integration of newly developed systems with human operational management, introducing quantifiable performance predictions into the ATM decision making process. These new decision support tools, based on AI predictions, will enable safety applications to create a more proactive, data-driven approach to safety management, capable of predicting potential safety risks in real time. There are examples where data analysis and Machine Learning (ML) techniques have been successfully used on safety-related problems such as detection of unstable approaches [1], [2] or runway excursions [3].

Among the different phases of a flight, the approach and landing phase is often considered one of the most critical. Statistics from 1995 to 2015 place 23% of fatal accidents occur during in this flight phase [4]. This is due to the special nature of the final approach and landing phase. The occurrence rate of errors or problems arising is higher than in other phases since pilots and Air Traffic Control Officers (ATCOs) have to deal with more situational changes, more decision making in a short period of time and more operational activity [5]. In order to mitigate the safety risks in the approach and landing phase, there is a special procedure called go-around, or missed approach, that aims to interrupt a landing that is considered unsafe. According to Flight Safety Foundation statistics, on average one to three go-arounds occur per thousand approaches [6]. One of the main reasons for initiating a go-around is an unstable approach. This procedure seeks to discontinue the landing which, if continued could extend to a more serious incident such as a runway excursion. However, evidence showed, that only 5-10% of all unstable approaches (which typically occur at around 2-5% of all landings) ended in a go-around [7]. This is why in recent years, different policies have been deployed to encourage crews to perform a go-around more often.

Even though performing a go-around is encouraged and they are standardised procedures, they can evolve into other safety problems due to their complexity. It should be noted that go-arounds are usually executed in a high workload situations for pilots and ATCOs as well as low altitudes, low speeds and close to the ground. This is why the work presented in this paper focuses on the development of an ML-based prediction tool that can provide advance warning of those aircraft that will potentially perform a go-around.

This paper falls under the scope of the SafeOPS project which investigates how predictive AI tools can be used in ATM as decision support technologies to facilitate controllers making complex decisions. The overall objective of SafeOPS

This project has received funding from the SESAR Joint Undertaking (JU) under grant agreement No 892919. The JU receives support from the European Union's Horizon 2020 research and innovation programme and the SESAR JU members other than the Union.

is to determine whether a predictive AI tool for go-around forecasting could decrease the workload and stress of the ATCOs by increasing their situational awareness and at the same time adding safety and resilience through increased predictability to the overall system.

The paper is organized as follows. First, Section II includes a more in-depth go-around problem definition and assessment based on a user-centered development as well as review of the main previous works done. Then, Section III provides and overview of the data used. Section IV describes the methodology used for the development of the tool and a benchmark of different possible model solutions. Section V provides the results of the final model developed and assessment of the explainability of the results. Finally, conclusions are drawn in Section VI.

II. Problem assessment and definition

A. Go-around scenario

One of the main causes of approach and landing accidents is the lack of recognition of the need and execution of a go-around. The Flight Safety Foundation published a study which states, that in situations of unstable approach if flight crews had decided to perform a go-around, 83% of runway excursions and 54% of all accidents analyzed could have been avoided [6]. During the development of the ML go-around prediction tool, to ensure that the technical work was consistent with stakeholder needs and perceptions, a user-centered development process was established. This process based on recurring workshops of researchers and stakeholders, mainly air traffic controllers and pilots. In these workshops, after reaching a common understanding of the objectives, possible application scenarios and use cases were designed, from which an initial set of requirements were derived [8].

Each airport publishes specific missed approach procedures for each of its runways, which are briefed in the cockpit before landing attempts. In case of a go-around, the flight crew usually follows the briefed procedure, which nevertheless requires close coordination with ATCOs. Even though flight crews are trained for go-around situations, a go-around can be a demanding, peak workload situation. Due to the high amount of traffic before Covid-19 at most European airports, following the standard missed approach procedure is not necessarily the safest option to resolve a go-around. The operational situation may be such, that to ensure separation (visual, radar or wake) between aircrafts, the ATCO will vector the go-around aircraft, rather than simply clear it for standard procedure. Under these circumstances, the controller's and pilots's workload can increase, leading to a stressful situation. The flight crew can also be overwhelmed by the required deviations from the standard go-around, causing problems in complying with the ATCO requests [9]. Therefore, a decision support tool, predicting go-arounds time in advance, could provide additional insight into the operational situation, and positively influences ATCO situational awareness and avoids additional stress factors.

Through the work carried out in the workshops with stakeholders, different scenarios were defined in which a go-around could represent a situation of high workload and stress. With the selection of scenarios, we intend to describe situations, in which a decision support tool can have an impact on the go-around handling of ATCOs. Fig. 1 (not to scale), shows a scenario, in which the standard go-around procedure dictates an overflight of the runway continued by a left turn. In situations where the runway is used in mixed mode operation, meaning departures and arrivals are handled on the same runway, the aircraft performing the go-around could, due to a higher performance, catching up with the departing aircraft, which would cause separation problems. To ensure radar separation, the ATCO must decide whether the outgoing go-around aircraft or the departing one should be vectored. The solution shown in Fig. 1 is to vector the go-around immediately to the left, with the consequent increase in flight crew workload. The introduction of a prediction tool could increase ATCO situational awareness. Depending on how far in advance the prediction is given the ATCO could, among other options, holding back the departing aircraft from lining up and if finally there is a go-around, allow the approaching flight crew to follow standard procedure or increase the gap between the two conflicting aircrafts by issuing a speed reduction to the arriving aircraft.

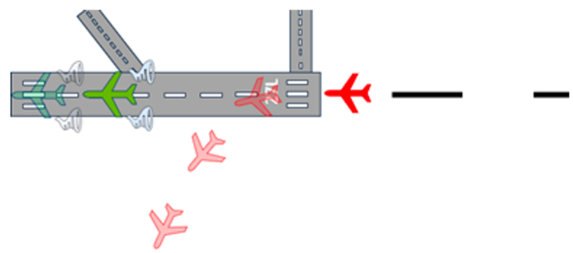

Fig. 1. Flight phase detection logic for ADS-B data.

B. Machine Learning prediction scenario

Based on the defined scenarios and user experience, gathered from the stakeholders, a set of requirements that englobe the go-around situations can be defined. These requirements are key to be able to translate the operational needs of the ATCOS into a more technical definition of the ML solution. The objective of the tool is to predict whether given an approach, the aircraft will land or perform a go-around. Therefore, we can define the ML problem as a binary classification problem. Additionally, to develop confidence and trust in the predictions, the solution presents a degree of explainability beyond just metrics to measure its performance, such as the confidence level in a prediction or feature importance. Moreover, we defined as a requirement that the level of false alarms should be as low as possible. A high level of false alarms can lead to loss of confidence in the tool. Therefore, the ML solution should to present high levels of precision even at the cost of detecting a lower number of go-arounds (recall).

The prediction horizon of the model is defined as the distance from the runway threshold of the runway at which the go-around prediction will be performed. To establish it, we take into account that the work focuses on the control zone and terminal control area, in which a a tower controller is responsible. The handover from the approach to the tower controller usually occurs approximately at 8-12 NM from the runway threshold, so this will be marked as the maximum prediction horizon of the model. ATCOs need to have as up-to-date a picture of the situation as possible to improve pre-planning and prepare as best as possible for changing circumstances. The earlier the information on a likely go-around is given, the more time the ATCO has to decide on its next actions. Therefore, four prediction horizon points spaced every 2 NM (At 2 NM, 4NM, 6NM and 8NM from runway threshold) are established. This distance spacing is decided for the initial prototypical development for the SafeOPS project, to evaluate the impacts of a decision support tool on the go-around handling. Depending on the field experts' feedback, the spacing might change in the later phases of the project. Here it is worth noting that, according to the feedback from experts collected during workshops, we can highlight the 4 NM prediction as the most relevant prediction horizon. This is the minimum distance at which the prediction can be still used to make operational changes (e.g., not authorizing a takeoff on the same runway or instruct a go-around to the approaching aircraft). From 4NM onwards all clearances and/or indications should be given and in this space nothing should be instructed (except in case of emergency) and thus, the prediction could only be used as a warning and to increase situational awareness.

Finally, the ML model should be able to make predictions with information that is available at the time of the prediction from the Air Traffic Control (ATC) side. This may include flight path information (e.g. ground speed or altitude), aircraft information (e.g. callsign), weather and airport traffic information.

C. State-of-the-art review

Most of the bibliography on go-arounds is focused on the study of their factors and causes [10], [11], or on the study of the high workload flight crew face in go-arounds procedures [9], [12]. Nonetheless, there are several works centered on go-around predictions with varying degrees of success.

The work done in [13] propose a prediction model based on Input-Output Hidden Markov Model (IO-HMM) predicting every 1 NM from the 10 NM of the runway threshold. The study period spanned approximately 6 months (July-December 2018) at JFK Airport with a total 100,032 arrivals and 371 go-arounds. The data used included: flight path data obtained from Integrated Flight Format (IFF) and Reduced Data (RD) summaries from the NASA Sherlock Data Warehouse, Airport Surface Detection Equipment Model X(ASDE-X)f for airport surface data and FAA aviation system ASPM) for airport configuration and weather related information. They applied a undersampling technique to mitigate the data imbalance problem. The best performance result obtained by the model was at 2 NM, detecting 41.7% of all go-arounds but with a precision of 15.5

The work done in [14] explores the use of different decision-tree based ML classification algorithms for the prediction of go-arounds. The study develops the models for runway 14 at Zurich International Airport. They use almost four years (1 January 2017 - 30 June 2020) of ADS-B (Automatic Dependent Surveillance–Broadcast) data which include almost 250,000 landings and 715 go-arounds. The authors methodology produced two types of models: macroscopic and microscopic. The macroscopic model aims to predict the occurrence of a go-around within the next hour, while the microscopic model predicts the go-around probability of each approach when the aircraft is at 10 Km (approximately 5.4 NM) from the runway threshold. As in the previous work [13], they also apply undersampling technique to mitigate the data imbalance problem. The best performance was obtained by using a Random Forest, which managed to detect 50% of the go-arounds but only 2% of the predicted go-arounds being actual go-arounds.

In [15] the authors use a tree-based machine learning model for go-around detection and prediction at specific distances from the runway threshold. The work is focused on Philadelphia International Airport for a period of approximately 10 months (1 March 2019 - 31 December 2019) with a total of 132,118 arrivals and 662 go-arounds. The data used included: ADS-B for flight trajectories data, Meteorological Aerodrome Reports (METAR) for weather information and open-access resources for aircraft data and flight timing (e.g. arrival time, departure time or touchdown time). The authors also apply undersampling technique to mitigate the data imbalance problem. Using an ensemble method (XGBoost) they obtained the best performance results at 2 NM being to correctly predict 56% of all go-arounds with a 90% precision.

III. Data

For the development of the ML support tool, we decided to focus on two major European airports: Frankfurt airport (EDDF) and Munich airport (EDDM). We decided to develop separate ML models for each airport since the particularities that each airport may present (e.g. runway configuration, operation minimas, noise restrictions) could hinder the performance of single general ML solution. In addition, this gives us the opportunity to compare and assess the performance and usability of ML solutions in go-around prediction scenarios. The types of data collected can be classified into three groups: flight data, aircraft data and meteorological data.

For flight data we used Automatic Dependant Surveillance-Broadcast (ADS-B) data provided by the OpenSky Network [16]. ADS-B is surveillance data that relies on aircraft broadcasting their identity, position and other information derived from on board system. Aircraft data was also obtained through the OpenSky Network. They provide a historical aircraft database with information related to an aircraft's ICAO24 unique 24-bit identifier, model or their Wake Turbulence Category (WTC). Finally, meteorological data was obtained through historical Meteorological Aerodrome Reports (METAR) extracted from Iowa State University [17]. METAR are routine aviation weather reports of actual observed conditions at an airport or near one (e.g. wind, horizontal visibility, cloud coverage).

A. Data Processing

The ADS-B data used in the present study were extracted by defining a 2D polygon (approximately 50 NM by 30 NM) centered on the airports of interest and obtaining all trajectories contained therein from April 2018 to February 2020. The first step in the processing pipeline is to identify unique flight based on the timestamp, date and callsign and then filter out all those ADS-B trajectories of flight over flights. Then, based on the evolution of the flight parameters, especially the altitude, it is identified whether the trajectory corresponds to an approach or a take-off.

The processing of approaching flights is mainly characterized by the standardization of flight parameter units, transforming them into International system (SI) units, and the elimination of possible errors in the data. The major errors present in the data are noisy/outlier data points and erroneous data points in the touchdown phase of the approach. For noise and outlier filtering, a median filter is used to remove erroneous data points [18]. To not lose granularity in the data, an interpolation process is applied to the removed data. Erroneous data in the touchdown phase of the approach can cause identical data in several data rows and/or tracking departures under the previous callsign. For this purpose the flag "On ground" in ADS-B data is added as checking flag. This check utilizes an engineered height above airport level to verify the "On Ground" flag and if discrepancies are detected to change the indication of the flag according to the height above airport level.

Finally, prior to the feature engineering activities, the separate flight phases contained in the flight data are detected. The detection of the flight phases is key for the identification and labelling labeling of go-arounds and is described in more detail in the next subsection. The processing of the identified departures contains the same steps as the approaches with minor differences in the specific pipeline.

B. Go-around Labelling

The go-around labelling is performed by means of a flight phase detection algorithm. The flight phase detection algorithm was initially developed for H2020 SafeClouds.eu project [19], a predecessor project of SafeOPS. The algorithm uses a flight phase detection based on a state machine shown in Fig. 2. The state machine defines a set of possible flight phases (or states) which an aircraft can transition into from a current flight phase. For example, from the flight phase "Climb" it could be transitioned into "Cruise" or "Descent". For both options a logic is defined which describes the necessary behavior of the flight data to be categorized as either one. Three different categories of behavior are used. The first category are direct booleans from the flight data (e.g. On ground flag). The second category are trends where timeseries are smoothed with a moving average and divided in a positive, negative or no trend. The last category are simple thresholds for certain time series. The transitions between the different phases in Fig. 2 are abbreviated as follows:

- Booleans:
  - A: is airborne
  - ¬: not true
- Continuous time series used for trends and thresholds:
  - H: pressure altitude
  - VS: vertical speed
  - G: ground speed
- Trend definition:
  - =: trend remains constant / rate is zero
  - +: trend increases / positive rate
  - -: trend decreases / negative rate

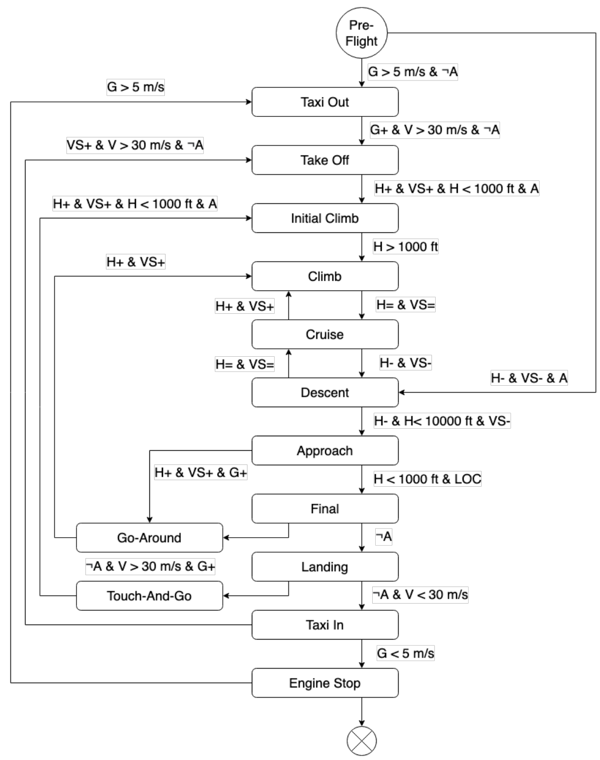

Fig. 2. Flight phase detection logic for ADS-B data.

Additionally, a shortcut was introduced from the Pre-Flight phase directly into the Descent phase, as the ADS-B data is already filtered in the processing pipeline to contain data from the terminal areas of the airports selected. The state machine also copes with flight data ending before the Cruise phase (i.e departing traffic). In the go-around labelling the state transition is defined with the three parameters pressure altitude, vertical speed and ground speed. The pressure altitude and the ground speed need to show a positive trend while the vertical speed needs to be positive. Fig. 3 shows, through the use of the flight phase detection algorithm, a go-around approach correctly identified and splitted into the go-around approach (blue) and the missed approach procedure and landing (red). However, not all go-arounds can be detected with such clear behaviour of the data, especially if there is a go-around in the early stages of the final approach. The thresholds that classify the trend can be fine-tuned to capture them correctly although, in most of these cases the maneuver starts even before 10NM prediction point so they are not of interest for this work.

Fig. 3. Visual example of go-around procedure.

C. Feature Engineering

Table I contains a summary of features that were used in the predictive models of this paper. The defined features have been grouped into four feature types according to their nature. In addition, we can distinguish between two features sources. ”Available in data” includes features that although may have undergone some transformation (e.g. change of units, remove outliers) are considered to be extracted directly from the data sources. ”Engineered feature” includes those features that have been constructed through the transformation and combination of primary features. This selection of features is intended to provide the most complete view of the operational scenario to model through the available data.

Table I also presents a brief description of each of the features extracted from the data. Even so, it is considered appropriate to give a more detailed explanation of some of the feature for their correct understanding. The ”Specific energy level” (SEL) is the sum of the specific potential energy and the specific kinetic energy:

$$SEL = h \cdot g + 0.5 \cdot V^2 \tag{1}$$

Where h is the aircraft’s altitude, V the ground speed and g the gravitational acceleration. SEL does not require having information related to the mass of the aircraft. However, the mass still has an influence as higher mass aircraft’s present higher ground speeds during the approach. ”Centerline deviation” refers to computed angular difference between the approached runways’s center-line and the true track between the aircraft position and the threshold position, as shown in Fig. 4. The ”Track/Runway Bearing deviation” feature refers to the computed angular difference between true track of the aircraft and true bearing of the runway, also shown in Fig. 4. Both these features combined with the distance to the runway threshold are used to identify the runway being approached by a flight, ”runway ID”. A metric defined as the geometric mean between these two features (Centerline deviation and Track/Runway Bearing Deviation) and the runway threshold is computed. The chosen runway will then be the one presenting a smaller value of this geometrical mean. Finally, the ”Approach type” is identified through the visibility and sky cover from the METAR reports. If visibility is lower than 5000 meters and the sky cover is below 1000 feet we establish IMC if not VMC.

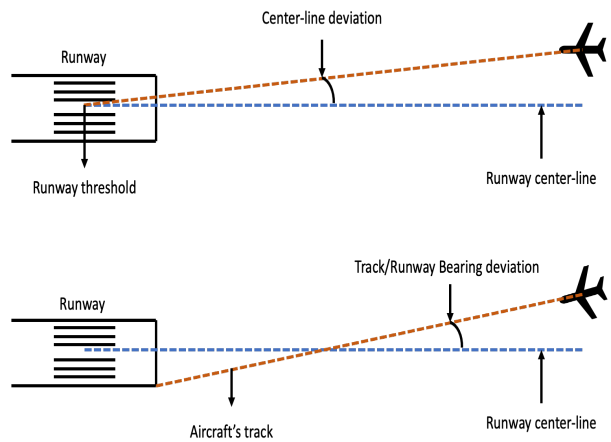

Fig. 4. Visual example of go-around procedure.

IV. Benchmark Study

Table II shows the total number of departures, approaches and go-arounds identified in the study period selected for this work. Due to the heterogeneous quality of ADS-B data source some arrivals and departures were discarded from the final data set, due to errors in the data (such as excessive number of outliers, on ground segment missing or to few data points during the approach phase). This may explain the discrepancy in Table II between the number of arrivals and departures. Although in the case of EDDF this discrepancy is less than 1%, in the case of EDDM it is higher due to significantly worse data coverage especially in the ground segment.

A preliminary analysis of the final data set showed that for EDDF and EDDM, approximately 98% of all go-arounds are initiated after 2NM prediction point and only 1% start before the 8NM prediction point. These results are similar to those obtained in other studies [20]. In addition, we can see how the ratio of go-arounds in the final data set is very small (approximately 3.5 per 1000 operations at both airports). This confirms that we are facing a highly imbalanced classification problem. In machine learning terms, a data set can be considered to be highly imbalanced when the ratio between the majority class and the minority class is greater than 1:100 [21]. In general, some classification models, by their nature, perform optimally on problems where the data

TABLE I
FEATURES DEFINED FOR PREDICTION MODEL

Feature type	Feature name	Sampling	Source	Description
Flight information	WTC	Static information	Engineered feature	Aircraft Wake Turbulence Category
	Approach attempt		Engineered feature	Flight approach attempt
	Hour		Available in data	Hour of the day
	Day		Available in data	Day of the week
	Week		Available in data	Week of the year
Weather data	Wind speed	Latest available METAR report at prediction	Available in data	-
	Wind direction		Available in data	-
	Temperature		Available in data	-
	Visibility		Available in data	-
	Approach type		Engineered feature	Instrument Meteorological Conditions (IMC) or Visual Meteorological Conditions (VMC)
	Dew point temperature		Available in data	Temperature below which the water will condense
	Ceiling height		Engineered feature	Ceiling height based on sky cover altitude and metar message
Approach performance	Runway ID	Distance from the threshold (every 0.5NM from 10 NM to 2 NM from threshold)	Engineered feature	Approached runway ID
	Specific energy level		Engineered feature	Aircraft specific energy level during the approach
	Ground speed		Available in data	Aircraft ground speed
	Vertical speed		Available in data	Descent vertical rate
	Vertical speed variance		Engineered feature	Descent vertical rate variance (window of 60s)
	Track		Available in data	Aircraft track
	Track variance		Engineered feature	Aircraft track variance (window of 60s)
	Altitude		Available in data	Aircraft altitude
	Track/Runway Bearing deviation		Engineered feature	Angular Deviation between aircraft track and runway bearing
	Centerline deviation		Engineered feature	Angular Deviation of aircraft position from runway centerline
Airport information	Total go-arounds	Time horizons (previous 10, 30 and 60 minutes)	Engineered feature	Total number of previous go-arounds at the airport
	Runway go-arounds		Engineered feature	Total number of previous go-arounds at the approaching runway
	Departures		Engineered feature	Total number of previous departures at the approaching runway
	Arrivals		Engineered feature	Total number of previous arrivals at the approaching runway
	Last departure time	Closest flight to approaching aircraft	Engineered feature	Time difference with previous departure at the approaching runway
	Last arrival time		Engineered feature	Time difference with previous approach at the approaching runway
	Last departure WTC		Engineered feature	WTC of the previous departure at the approaching runway
	Last arrival WTC		Engineered feature	WTC of the previous arrival at the approaching runway

TABLE II
TOTAL NUMBER OF DEPARTURES, APPROACHES AND GO-AROUNDS PER AIRPORT

Airport	Number of departures	Number of approaches	Number of go-arounds	Go-arounds per 1000 approaches
EDDM	264418	219488	773	3.52
EDDF	367861	370855	1318	3.55

set is balanced or slightly imbalanced. In these cases, it is sometimes necessary to apply resampling techniques. This resampling can be an undersampling (where the number of examples of the majority class is reduced) or an oversampling (increasing the number of examples of the minority class). In some cases resampling is not always a solution, and both undersampling and oversampling present their pros and cons.

In this work we have explored the use of the oversampling technique known as Synthetic Minority Oversampling Technique (SMOTE). When using SMOTE, the minority class is over-sampled by taking each sample from the minority class and introducing synthetic examples along the line segments linking any/all of the k nearest neighbors of the minority class [22]. Together with SMOTE, and as suggested by its authors [22], we explored combining it with a random undersampling technique by which examples of the majority class are randomly removed from the dataset. In addition, where applicable, the use of Cost-Sensitive Algorithms has been explored to combat the imbalance in the data by adding an additional cost for misclassification to some of the models developed.

To provide a more in-depth analysis of the possibilities of using ML models to predict go-arounds, we decided to produce an initial benchmark study where nine of the most common types of ML models were trained in order to compare their performance. In order to save computational effort, the benchmark was performed by training the different models for 4NM prediction. The models were trained with some minor modification of their hyperparameters, as the aim is to obtain an initial result of the most promising model and provide a baseline for comparison. Moreover, some of the resampling and cost-sensitive techniques were used to combat data imbalance. Three different types of models were proposed for study:

- Linear algorithms: Naive Bayes and Logistic Regression
- Non-linear algorithms: Decision Tree, K-Nearest Neighbours and Multi-layer Perceptron
- Ensemble Models: Random forest, Adaptive boosting and Gradient boosting

To validate the different models, we use a cross-validation technique called K-Fold cross-validation. Cross-validation is used to avoid typical problems in ML model development such as overfitting or selection bias. In K-Fold cross-validation, the data is divided into k subsets and the holdout method is repeated k times. Each time, one of the k subsets is used as the test data set and the other k-1 subsets are used as the training data set. The error estimate is averaged over all k trials to calculate the total model performance [23]. For each model of the benchmark we use three-fold cross validation repeated 2 times.

It is crucial during the model training and stage and cross-validation process to ensure that information from the validation data set is not used in training. When this happens it is known as "data leakage" and can cause overly optimistic results and therefore invalid predictive models to be created. It is also important to ensure that if an oversampling technique such as SMOTE is used, it is only applied to the training data and not to the validation data. Tables III and IV show the results of the performance metrics (precision, recall, F1-score and Receive Operating Characteristics-Area Under the Curve (ROC-AUC) for the benchmark study by airport.

TABLE III
EDDM MODEL BENCHMARK STUDY RESULTS

EDDM - 4 NM				
Model	**Precision**	**Recall**	**F1-score**	**ROC-AUC**
Naive Bayes (scikit-learn)	0.06	0.29	0.10	0.76
Logistic regression (scikit-learn)	0.67	0.15	0.25	0.80
K-Nearest Neighbours (scikit-learn)	0.84	0.12	0.21	0.64
Decision Tree (scikit-learn)	0.17	0.24	0.20	0.62
Multi-layer Perceptron (scikit-learn)	0.40	0.22	0.28	0.72
Random Forest (scikit-learn)	0.77	0.21	0.32	0.80
Adaptive Boosting (scikit-learn)	0.70	0.18	0.29	0.82
Gradient Boosting (XGBoost)	0.80	0.21	0.34	0.85
Gradient Boosting (LightGBM)	0.77	0.22	0.34	0.88

Results of the benchmark study show that for both airports the best overall performing models are the Ensemble models. These types of models work by combining the prediction of multiple so called simpler "weaker" models (e.g. decision trees) to improve their overall performance. Of the Ensemble models tested, the one with the most complete performance were the Gradient Boosting models. For the benchmark study we tested several implementations of this type of model from python libraries such as Scikit-learn, XGBoost and LightGBM. Among these implementations we finally decided to select LightGBM as the best suited for go-around prediction. Although its performance is equivalent to the others, LightGBM presents a more lightweight model which significantly shorter training times compared to other ensemble models others (up to 10 times less). LightGBM is a Gradient Boosting Decision Tree (GBDT) algorithm which combines the techniques of gradient-based one-side sampling and exclusive feature bundling. Unlike other GBDT algorithms, such as XGBoost, LightGBM grows trees vertically while other algorithms grow trees horizontally, which makes LightGBM, in certain occasions, more effective method for processing large datasets and features [24].

TABLE IV
EDDF MODEL BENCHMARK STUDY RESULTS

EDDF - 4 NM				
Model	**Precision**	**Recall**	**F1-score**	**ROC-AUC**
Naive Bayes (scikit-learn)	0.07	0.42	0.13	0.84
Logistic regression (scikit-learn)	0.74	0.22	0.34	0.82
K-Nearest Neighbours (scikit-learn)	0.89	0.34	0.49	0.73
Decision Tree (scikit-learn)	0.32	0.41	0.36	0.71
Multi-layer Perceptron (scikit-learn)	0.60	0.38	0.47	0.78
Random Forest (scikit-learn)	0.84	0.44	0.58	0.87
Adaptive Boosting (scikit-learn)	0.62	0.23	0.34	0.88
Gradient Boosting (XGBoost)	0.90	0.40	0.57	0.89
Gradient Boosting (LightGBM)	0.84	0.43	0.57	0.90

After selecting the LightGBM ensemble model as the model for the go-around prediction tool we proceed to a more exhaustive development. Applying the SMOTE resampling technique, we observe that the best results are obtained when the number of values in the minority class (go-arounds) is increased to about 5% of the number of values in the majority class. When combined with random undersampling and the use of cost-sensitive penalisations, we observed a worsening in performance. Its use caused an over-correction where the total number of detected go-arounds (recall) increased, but at the cost of drastically reducing the precision. For this reason, the use of these techniques in the final models was discarded.

To optimize the prediction capacity of the LightGBM model for both airports we proceed to optimize its hyperparameters. Therefore, we use Bayesian hyperparameter optimization via the Optuna library. Bayesian hyperparameter optimization works by building a probability model of the objective function and using it to select the most promising hyperparameters to evaluate in the true objective function [25]. In this case the objective function was to minimize the precision-recall (PR) performance metrics

Finally, for the hyperparameter tuning we split the total data set (per airport) into a ratio of 75%(training and testing)-25%(validation). This splitting is done randomly, although the distribution of the target variable (go-around/No go-around) in each partition is maintained. The 75% data partition is used to perform a Flat cross-validation optimisation of the hyperparameters. Once the best hyperparameters are found, they are used to retrain a final model using all the training and testing data. The validation data partition is used to validate the final model performance. Even though this procedure may not be the most optimal one, it sufficiently reduces bias and provides reliable information on model performance. Although the use of Nested cross-validation is considered a better option to reduce bias, there is evidence that its high computational cost in large data-sets does not usually generate a relevant improvement and the results are usually equivalent to using only a Flat cross-validation [26].

V. Results

A. LightGBM go-around prediction performance

Tables V and VI show the performance results of the models for EDDM and EDDF after performing the hyperparameter tuning. After the hyperparameter tuning, a slight improvement in the models performance can be seen. As expected, the model's performance improves as we approach the runway threshold. In the case of EDDM at the 2 NM prediction point we can see the model has been able to correctly identify approximately 54% of all go-arounds with only a 7% rate of false positives. For EDDF at the 2 NM prediction point the model correctly identifies 63% of all go-arounds with a 10% rate of false positives.

TABLE V
LightGBM performance metrics after hyperparameter tuning (EDDM)

EDDM				
Prediction point	**Precision**	**Recall**	**F1-score**	**PR-AUC**
2 NM	0.9364	0.5365	0.6821	0.6574
4 NM	0.9111	0.2135	0.3460	0.3521
6 NM	0.9090	0.1042	0.1869	0.1950
8 NM	0.8462	0.0573	0.1073	0.1832

TABLE VI
LightGBM performance metrics after hyperparameter tuning (EDDF)

EDDF				
Prediction point	**Precision**	**Recall**	**F1-score**	**PR-AUC**
2 NM	0.9071	0.6347	0.7468	0.7330
4 NM	0.8974	0.4281	0.5797	0.5285
6 NM	0.8957	0.3150	0.4661	0.4186
8 NM	0.8764	0.2378	0.3741	0.3616

Tables V and VI show that, although the performance increased with decreasing distance at both airports for all prediction points, the prediction accuracy for the positive class (Go-around) is very high (¿80%). This translates into less false positive alarms for the ATCO. However, it is also worth mentioning that the model is very selective in classifying the positive cases. For example, in the case of EDDF at the 4 NM prediction point the model manages to predict slightly less than half of the approaches that execute a go-around (43%). In the case of EDDM, the model only manages to predict slightly less than a quarter of the go-around approaches (21%). In comparison with the work done previously in [13], [14], the performance results obtained are better both in terms of precision and recall. Compared to the work done in [15], recall results for EDDM are similar to those in Philadelphia International Airport (KPHL), while EDDF has a higher recall for all prediction points. There is an improvement with respect to precision. While in [15] a high precision (90%) is obtained at 2 NM, there is a decrease as we move away from the threshold (e.g. 66% precision at 6 NM). In our case we obtain more stable precision values in both airports and for all prediction points. This means that with our ML solution an ATCO should expect a lower level of false alarms (false positives) making it a more efficient and effective tool for go-around prediction.

Fig. 5 and 6 present the Precision-Recall (PR) for both airports. The PR curve is usually a more appropriate metric than the ROC, because both precision and recall do not take into account True negatives. Furthermore, the PR curve also makes it easier to visualize the trade-off between precision and recall. The EDDF models present a higher PR-AUC (0.36-0.73) than the EDDM model (0.18-0.66). These results have to be seen from a data imbalanced perspective. Given that over 99% of the data present non go-arounds approaches, the threshold for a classifier without skill would be around 0.003. This means that for EDDF the classifier model is between 120 and 243 times better than a random classifier and in the case of EDDM between 60 and 220 times better.

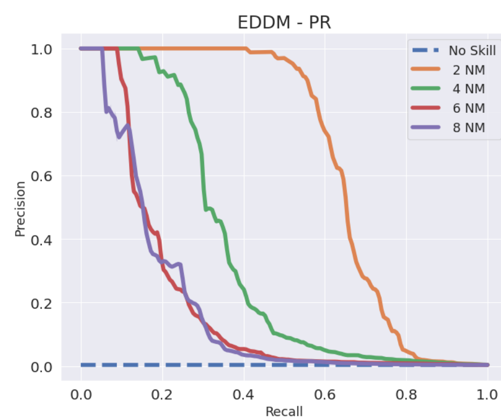

Fig. 5. EDDM - Precision-Recall (PR) Curve

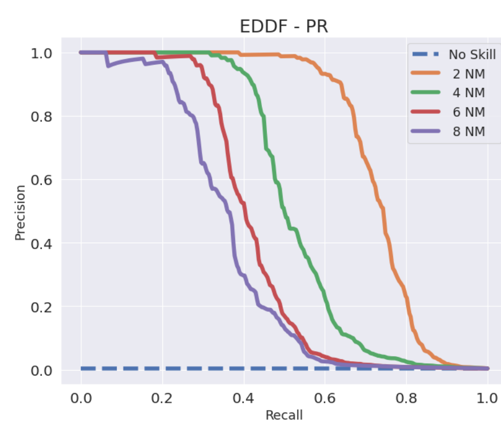

Fig. 6. EDDF - Precision-Recall (PR) Curve

B. LightGBM explainability

A general analysis of the explainability of the developed models is carried out in order to better understand the the inner workings of the model, validate its performance and generate trust in the model by the end users. For this purpose, a feature importance ranking is performed and the results are analyzed for the different prediction points. The feature importance is generated using the Python library SHAP (SHapley Additive exPlanations). SHAP is a game theoretic approach to explain the output of any machine learning model [27]. Fig. 7 and 8 shows the SHAP "Local explanation" plot for the 4 NM prediction point models developed for EDDM and for EDDF. Each point on these figures is a SHAP value for a feature and an instance in the data. The y-axis contain the 20 most important features for each model ordered from top to bottom from most to least important. The x-axis shows the SHAP value where positive values indicate that they favour a positive prediction (go-around) and negative values favour an approach without a go-around. Finally, the colour scale represents t the value of the feature from low to high. Grey color features are categorical features (e.g. WTC) which do not have an intrinsic order in them.

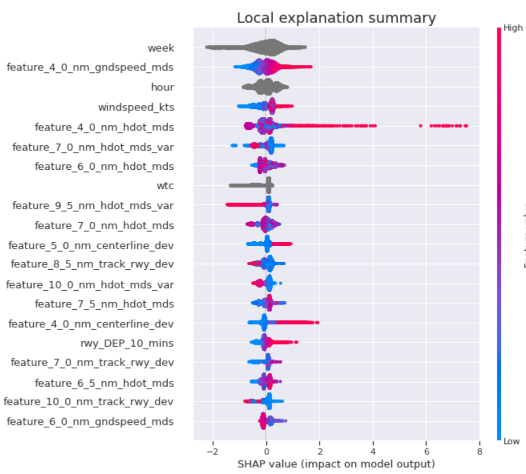

Fig. 7. EDDM - Top 20 SHAP Feature importance at 4 NM

Fig. 7 and 8 show how the main features differ slightly between the two models. The most important feature in the EDDM model is the week of the year followed by the ground speed (gndspeed_mds) at 4NM. In the case of the EDDF model, the week of the year is not in the top 20 and the ground speed at 4NM is the 4th most important feature. We can see in the case of EDDM how approaches with high values of ground speed and vertical speed (hdot_mds)are more likely to be classified as go-arounds. For EDDF the most important features are centerline deviation (centerline_dev) at 4NM and energy level at 4NM. In both cases it can be seen how high values of this feature favor a prediction of a go-around. Even with these differences, in both cases there is a preponderance of "Approach performance" type features.

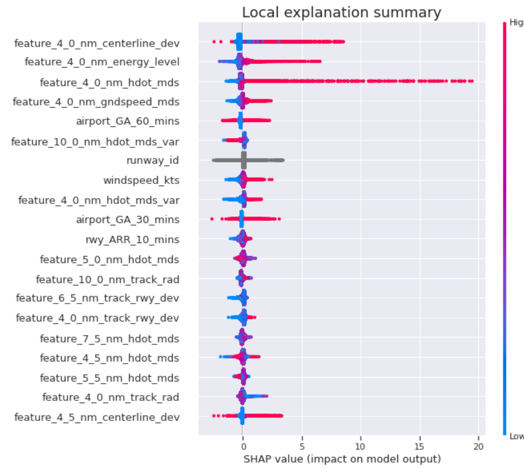

Fig. 8. EDDF - Top 20 SHAP Feature importance at 4 NM

Especially noteworthy is the relevance of the vertical speed and vertical speed variance (hdto_mds_var) features where in both models appears repeatedly at different sampling points. This gives an indication that the vertical speed profile that an aircraft has during an approach is a major determining factor in the occurrence of a go-around. We can also see the high importance in both airports of the wind speed features as well as "Airport information" type features such as the total number of go-arounds in the previous 60 minutes (airport_GA_60_mins) or runway departures in the previous 10 minutes (rwy_DEP_10_mins).

For the prediction point at 2NM at both airports, the main features are almost exclusively related to the "Approach performance". This may be due to the fact that these are generally the ones that were used in the labeling process, see Fig. 2, and therefore are the most likely ones to influence the model, being more relevant at distances closer to the runway threshold. On the other hand, at 8NM for both EDDM and EDDF "Approach performance" features such as vertical speed or centerline deviation are still the most important. However, we also find a greater number of "Weather data" and "Airport information" features. In particular, we can highlight ceiling height, visibility, runway departures and arrivals in the previous 10 minutes.

VI. Discussion

The work presented in this paper is a continuation of efforts to achieve accurate predictions of go-arounds. A unique data labelling system based on flight phase detection has been used and a benchmark study with different ML classification models has been performed. In addition, we have also explored the explainability of the final ML solution developed for transparency and to generate trust between users and tool.

An analysis of the most suitable models allowed us to conclude that an ensemble model (LightGBM) combined with an oversampling technique (SMOTE) gave the best results. We obtained promising results and compared them to the current state of art. Contrary to undersampling, oversampling does not entail a loss of information, which given the rare nature of go-arounds and their similarity with normal approaches we believe is the reason for the performance of the models. The results show an improvement of the current state of art.

Even so, the complexity of predicting these events has been proven due especially to the large imbalance in the data (3 go-arounds per 1000 approaches). Go-arounds are high workload procedures in which the pilot's decisions have a great influence and whose information is difficult to reflect in the data used. Also hindering performance was the limited information in ADS-B (e.g. high level of outliers/noise, temporal resolution or lack of information on surface movements), as well as the limited time resolution and quality (limited to on-ground measurements) of METAR meteorological data.

Even with all these challenges, we have been able to obtain high precision and decent recall values for all prediction points in both airports. Validating these results with stakeholders shows some viability to a tool with this performance. The level of false alarms would be very low and although not all go-arounds are captured as they are standardized procedures, false negatives do not pose a safety risk. However, work will be still needed to develop the adequate risk framework models to correctly understand how such a tool would impact the ATCO's decision making process.

Future work will focus in trying to improve the performance of the developed models. To this end, we will continue to detect and correct any possible errors in the data, explore the development of new features as well as a performing a feature selection during the training stages to try to eliminate less important features that may introduce noise to the data. Moreover, we will explore the use of more powerful deep learning models. These models can allow us to better capture the nature of go-arounds increasing performance at the cost of losing some level of explainability.

ACKNOWLEDGMENT

The authors would like to thank the "SafeOPS" consortium for their support in the work carried out as well as to all the stakeholders involved in the various workshops for providing their operational expertise in helping us understand the go-around scenario.

REFERENCES

[1] Singh, Narendra Pratap, Sim Kuan Goh, and Sameer Alam. "Real-time unstable approach detection using sparse variational gaussian process." In 2020 International Conference on Artificial Intelligence and Data Analytics for Air Transportation (AIDA-AT), pp. 1-10. IEEE, 2020.

[2] Martınez, Darıo, Antonio Fernández, Pablo Hernández, Samuel Cristóbal, Florian Schwaiger, José Marıa Nunez, and José Manuel Ruiz. "Forecasting unstable approaches with boosting frameworks and lstm networks." 9th SESAR Innovation Days (2019).

[3] Odisho, Edwin V., Dothang Truong, and Robert E. Joslin. "Applying machine learning to enhance runway safety through runway excursion risk mitigation." Journal of Aerospace Information Systems 19, no. 2 (2022): 98-112.

[4] Boeing, Statistical summary of commercial jet airplane accidents, worldwide operations, 1959–2015. Boeing Commercial Airplanes, Seattle, WA (2016).

[5] Wang, Lei, Yong Ren, Hui Sun, and Chuanting Dong. "A landing operation performance evaluation system based on flight data." In International Conference on Engineering Psychology and Cognitive Ergonomics, pp. 297-305. Springer, Cham, 2017.

[6] Blajev, Tzvetomir, and W. Curtis. "Go-Around Decision Making and Execution Project: Final Report to Flight Safety Foundation." Flight Safety Foundation (2017).

[7] IATA, "Unstable Approaches: Risk Mitigation Policies, Procedures and Best Practices, 3rd Edition," 2013. [Online].

[8] SafeOPS Consortium, "D2.1 - Development of Use Cases , User Stories and Requirements," 2021.

[9] Dehais, Frédéric, Julia Behrend, Vsevolod Peysakhovich, Mickaël Causse, and Christopher D. Wickens. "Pilot flying and pilot monitoring's aircraft state awareness during go-around execution in aviation: A behavioral and eye tracking study." The International Journal of Aerospace Psychology 27, no. 1-2 (2017): 15-28.

[10] Kumar, Satvik G., Samantha J. Corrado, Tejas G. Puranik, and Dimitri N. Mavris. "Classification and analysis of go-arounds in commercial aviation using ADS-B data." Aerospace 8, no. 10 (2021): 291.

[11] Gariel, Maxime, Kevin Spieser, and Emilio Frazzoli. "On the statistics and predictability of go-arounds." arXiv preprint arXiv:1102.1502 (2011).

[12] Schmidt, Tim André, Houda Kerkoub Kourdali, and Jim Nixon. "Evaluating process-based and crew-centred approaches to procedure design in aviation: Workload and performance changes in go-around manoeuvres." Applied Ergonomics 90 (2021): 103244.

[13] Dai, Lu, Yulin Liu, and Mark Hansen. "Predicting Go-around Occurrence with Input-Output Hidden Markov Model." In International Conference on Research in Air Transportation. 2020

[14] Figuet, Benoit, Raphael Monstein, Manuel Waltert, and Steven Barry. "Predicting airplane go-arounds using machine learning and open-source data." Multidisciplinary Digital Publishing Institute Proceedings 59, no. 1 (2020): 6.

[15] Dhief, Imen, Sameer Alam, Chan Chea Mean, and Nimrod Lilith. "A Tree-based Machine Learning Model for Go-around Detection and Prediction."

[16] Bringing up OpenSky: A large-scale ADS-B sensor network for research Matthias Schäfer, Martin Strohmeier, Vincent Lenders, Ivan Martinovic, Matthias Wilhelm ACM/IEEE International Conference on Information Processing in Sensor Networks, April 2014

[17] I. E. Mesonet, "Asos-awos-metar data download."

[18] Stone, David C. "Application of median filtering to noisy data." Canadian Journal of chemistry 73, no. 10 (1995): 1573-1581.

[19] SafeClouds.eu Consortium, "D4.3 Predictive analytics and identification of unknown hazards"

[20] Proud, Simon Richard. "Go-around detection using crowd-sourced ADS-B position data." Aerospace 7, no. 2 (2020): 16.

[21] Krawczyk, Bartosz. "Learning from imbalanced data: open challenges and future directions." Progress in Artificial Intelligence 5, no. 4 (2016): 221-232.

[22] Chawla, Nitesh V., Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. "SMOTE: synthetic minority over-sampling technique." Journal of artificial intelligence research 16 (2002): 321-357.

[23] Rodriguez, Juan D., Aritz Perez, and Jose A. Lozano. "Sensitivity analysis of k-fold cross validation in prediction error estimation." IEEE transactions on pattern analysis and machine intelligence 32, no. 3 (2009): 569-575.

[24] Ke, Guolin, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. "Lightgbm: A highly efficient gradient boosting decision tree." Advances in neural information processing systems 30 (2017).

[25] Wu, Jia, Xiu-Yun Chen, Hao Zhang, Li-Dong Xiong, Hang Lei, and Si-Hao Deng. "Hyperparameter optimization for machine learning models based on Bayesian optimization." Journal of Electronic Science and Technology 17, no. 1 (2019): 26-40.

[26] Wainer, Jacques, and Gavin Cawley. "Nested cross-validation when selecting classifiers is overzealous for most practical applications." Expert Systems with Applications 182 (2021): 115222.

[27] Lundberg, Scott M., Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan M. Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. "From local explanations to global understanding with explainable AI for trees." Nature machine intelligence 2, no. 1 (2020): 56-67.