EXPLORATORY RESEARCH

sesar JOINT UNDERTAKING

D4.2: Human interpretability framework for the selected user stories

Deliverable ID:	**4.2**
Dissemination Level:	**PU**
Project Acronym:	**SafeOPS**
Grant:	**892919**
Call:	**H2020-SESAR-2019-2**
Topic:	**SESAR-ER4-06-2019**
Consortium Coordinator:	**TUM**
Edition date:	**15 July 2022**
Edition:	**00.01.00**
Template Edition:	**02.00.03**

EUROPEAN PARTNERSHIP

Co-funded by the European Union

Authoring & Approval

Authors of the document

Name / Beneficiary	Position / Title	Date
Pablo Hernández / INX	WP4 Leader	11/07/2022
Clara Argerich / INX	Project Team Member	11/07/2022
Lukas Beller / TUM	Project Coordinator	11/07/2022

Reviewers internal to the project

Name / Beneficiary	Position / Title	Date
Pablo Hernández / INX	WP4 Leader	11/07/2022
Lukas Beller / TUM	Project Coordinator	11/07/2022

Approved for submission to the SJU By - Representatives of all beneficiaries involved in the project

Name / Beneficiary	Position / Title	Date
Pablo Hernández / INX	WP4 Leader	11/07/2022
Lukas Beller / TUM	Project Coordinator	11/07/2022

Document History

Edition	Date	Status	Name / Beneficiary	Justification
00.00.01	01/06/2022	Initial Structure	Pablo Hernandez/INX	
00.00.02	15/06/2022	Initial Content	Pablo Hernandez/INX	
00.00.03	01/07/2022	Final Draft	Pablo Hernandez/INX	
00.00.04	08/07/2022	Initial Word Export	Ines Gomez/INX	
00.01.00	15/07/2022	Initial Release	Pablo Hernandez/INX	Initial release

SafeOPS

FROM PREDICTION TO DECISION SUPPORT - STRENGTHENING SAFE AND SCALABLE ATM SERVICES THROUGH AUTOMATED RISK ANALYTICS BASED ON OPERATIONAL DATA FROM AVIATION STAKEHOLDERS

This deliverable is part of a project that has received funding from the SESAR Joint Undertaking under grant agreement No 892919 under European Union's Horizon 2020 research and innovation programme.

Abstract

The **SafeOPS** project aims at investigating the impact of possible **artificial-intelligence-based decision-support systems** on routine air-traffic operations. The context selected for this investigation are missed approaches, initiated by the flight crew of a landing flight and the subsequent go-around. The go-around scenario has a number of uncertainties and therefore makes it an ideal candidate for the integration of a predictive technology to support air traffic controllers (ATCO's) in managing aircraft in this situation.

To this end, three main pillars were defined in the project to develop the solution: **Operational Layer (OL)**, **Risk Framework (RF)** and **Predictive Layer (PL)**. The latter, which is developed within Work Package 4 of the SafeOPS project, addresses all big data and AI related tasks. It focuses on two main objectives. The first one covering all the related actions for the creation of the necessary automated data pre-processing and preparation pipelines. The second focuses on the AI/ML solutions for the predictive analytics that will be chosen and trained with a special focus on the human interpretability aspect of the solution. The trained AI/ML solutions will be developed, delivering the predictive analytics to the Risk Framework (RF).

This report addresses the second phase in the development process of the Predictive Layer. The report aims to provide the **predictive results** obtained through the use of the data infrastructure developed for the project (automated data processing pipelines responsible for the structuring, fusion, feature engineering and labelling of the data) for the two main ML case studies defined for go-around prediction scenarios. In addition, this report also includes an analysis of the **explainability** and **interpretability** of the results obtained from the models in order to make the models transparent and to **generate trust** between the model's performance and the possible human users.

EUROPEAN PARTNERSHIP

Co-funded by the European Union

Table of Contents

EUROPEAN PARTNERSHIP

Co-funded by the European Union

List of Tables

List of Figures

EUROPEAN PARTNERSHIP

Co-funded by the European Union

1 Introduction

1.1 Project overview

SafeOPS investigates the impact of possible **artificial-intelligence (AI) based decision-support systems** on routine air-traffic operations. Thereby SafeOPS focuses its research on "**from prediction to decision"**, a common decision-making paradigm in digitalization and predictive analytics. The envisioned decision support concept can be summarized by expanding the current ATM system with an information automation-based decision intelligence. Information Automation describes the automated acquisition and processing of operational performance data through big data technologies and AI algorithms, providing new information to the ATM systems.

The **scenario** selected in SafeOPS for this investigation is the **go-around** performed by landing aircraft and the subsequent missed approach procedure. The go-around scenario, which scope is defined in Deliverable 2.1 [1], for this project, has a number of uncertainties and safety critical factors associated with it, which are discussed in more detail in Deliverable 3.2. It is therefore an ideal candidate for studying the integration of a predictive technology, with the aim of providing greater support to Air Traffic Controllers (ATCOs). For the selected go-around scenario, the project develops and provides an integrated model of risk, incorporating all potential uncertainties. The model allows discussing safety scenarios in a coherent, probabilistic approach. It will include historical aircraft, weather and traffic data, and the outcome of AI/ML models. The computed risk is added information, which flows into the planning and operational management of the overall ATM system. Using this approach, potential risks could be actively managed.

The question the SafeOPS project looks to answer is, how the nature of these information will change the way the system, in this case the tower controller's approach and go-around handling, is operated. Beyond "information overflow", human operators using AI/ML systems will have to adapt not just to more information, but especially to the **probabilistic nature of this information**. While very powerful, many AI/ML solutions are far from being deterministic as they use of **randomness during learning**. Although this may seem negative, it allows algorithms to avoid getting stuck and to achieve results that deterministic algorithms cannot achieve. On the contrary, users will have to understand and interpret (to a greater or lesser extent) correctly the probabilistic nature behind these systems. Clever HMI refinements will certainly help to mitigate the potential overflow of information. However, also research on the impact of information automation on the ATM system needs to be conducted. It must show that an increase of capacity and cost-efficiency can be achieved and also the resilience of the system is maintained or further improved. SafeOPS aims to foster a collaborative paradigm that involves both the world of ATM and the world of airline operations to identify possibly hidden safety risks.

The work presented in this deliverable focuses on the more technical side of the project. It builds on previous work done in the project, mainly:

- Investigate concepts for **the integration of AI/ML based decision support tools in ATM**, and evaluate the effects on capacity **safety and resilience** of the ATM operation. Several potential Use Cases were identified for a data-driven decision support tool in the go-around scenarios. This work can be found in **Deliverable 2.1** of SafeOPS [1].

EUROPEAN PARTNERSHIP

Co-funded by the European Union

- **Human Factors assessment of risk information provision**, following a user centric approach to the design on how to adequately provide real time information of different ML solutions to the end users. This work can be found in **Deliverable 3.3** of the project [2].
- The development of a complete **data pipeline** and initial **AI/ML solution assessment** for the prediction of go-arounds scenarios using available operational (ADS-B) and weather (METAR) data. This work can be found in **Deliverable 4.1** of the project [3].

1.2 Deliverable objective

This is the second deliverable of **Work Package 4 (WP4)** of the SafeOPS project. The overall objective of this work package is to perform all the technical tasks related to the development of an AI/ML solution (**Predictive Layer**). The overall objective of this deliverable is to provide the final predictive results of the developed ML models as well as an analysis on the **interpretability** of the models themselves in order to provide some transparency on how they work. The results will presented of an update of the model developed in the deliverable 4.1 [3] after correcting some errors detected in the data processing and the creation of new features as well as an extension where the prediction is now performed for different threshold distances (2NM, 4NM, 6NM and 8NM). In addition, results are also presented for the second ML case study (**ML_CS_02**) where the prediction is performed continuously at fixed time intervals during the final approach phase. For both cases, an analysis of the **interpretability** and **explainability** is carried out, identifying the main factors identified by the models for the prediction of go-arounds. Finally, an analysis of all the work done in WP4 is carried out and, in particular, assess how our developed ML solutions meet the requirements set out in the early stages of the project in **WP2** [1].

1.3 Deliverable structure

The present deliverable includes the following sections:

- Section 2 contains an introduction to the idea of **Interpretability** and **Explicability** with regards to development and deployment **Machine learning solutions**;
- Section 3 contains a review of the final **performance** results of the machine learning solutions developed for the **proposed case studies** in the project;
- Section 4 provides a **in-depth analysis** of the interpretability of the developed solutions through the use of different **interpretability techniques**;
- Section 5 contains the main **conclusions** extracted from the work done in WP4 as well as review of the **fulfilment of requirements** defined in WP2 for the technical solution.

EUROPEAN PARTNERSHIP

Co-funded by the European Union

2 Interpretable Machine Learning

2.1 Interpretability in Machine Learning

The advances of **Digitalisation** enables the possibility of exploiting the advances made in recent decades in **Artificial Intelligence** to try to tackle situations that previously, due to their complexity, were only reserved for humans. **Machine Learning** models are revolutionising different industries and their usefulness is indisputable. But with their entry into service, one of the questions that has arisen the most is: **How is the model doing what it's doing?** That is why **Explainability** and **Interpretability** have become some of the most explored topics in AI in recent years, coining the term Explainable AI (XAI). Interest has arisen that the development of ML solutions should not only validate their performance, but also include some kind of interpretability/explainability analysis to build trust between the models and their users, especially when used in **critical decision-making tasks.**

Due to their nature, **Go-arounds** are **high-workload** situations and thus we consider it important in SafeOPS that any technical solution developed should dedicate an effort to analyse and study the interpretability/explainability of the models developed. As a way not only to understand which variables affect more in the prediction but also to generate **confidence** and **trust** in the tool on the part of the ATCOs.

Why is interpretability/explainability important? To answer this question, we must first define what we mean by each of these concepts. Although commonly used in the literature in equivalent terms due to their vagueness, but sometimes these terms do not refer to the same thing. Interpretability can be defined as the ability of a model to be transparent and be able understand the cause and effect within while explainability the ability of a model to provide the user some level of explanation for its predictions [4]. In SafeOPS, we have preferred to use the definition of interpretability as **the degree to which a human can understand the cause of a decision** and equate both terms in the sense that they both refer to the ability to provide information about the predictions made by a ML model [5].

One can often argue that the interpretability of the model is not always important and this is indeed the case. Interpretability is always desirable but not necessary. This gives rise to the **Interpretability-Accuracy Trade-off.** The most common belief being that those models with greater interpretability tend to perform worse than what we could call "black box" models (models which, in general, cannot be understood simply by looking at their parameters). In general, there will be situations where the predictions of a ML model are not used in critical scenarios (e.g., movie recommendation systems). In these cases, we may be more concerned with performance than interpretability as any erroneous predictions will have relatively low cost. But there are many safety-critical scenarios, such as in this case of go-arounds, where this trade-off must be re-evaluated. In these situations, it may be of interest or necessity for the human user using the prediction to have some information about the causes of this prediction. To make decisions accordingly or if necessary to discard any misleading predictions.

EUROPEAN PARTNERSHIP

Co-funded by the European Union

2.2 Model interpretability

There are a number of techniques and tools designed to provide explanations to ML models. These different methods of interpretation can be broadly classified according to the results we obtain from [6]:

- **Feature analysis:** Most modern algorithms (e.g., ensemble models) can provide certain summary statistics for each feature. This can range from a single metric related to importance to more elaborate metrics, such as pairwise feature interaction strength. In addition, it is sometimes required to visualize feature statistics for their correct interpretation. One example is "Partial dependence plots" which are curves that show a feature and the average predicted outcome.
- **Data point:** There are interpretation techniques that rely on the analysis of data points (existing or synthetic) to explain the performance of a model. An example is the method of counterfactual explanations, whereby a data point similar to the one of interest is created and some of the features are modified to find those that cause the change in prediction.
- **Model internals:** As mentioned previously, there are some types of models that by their nature are considered intrinsically interpretable. From these it is possible to extract information directly from them such as for example the weights in linear models or the learned tree structure of decision trees.
- **Surrogate models:** This technique approximates the behaviour of a complex model by using a more interpretable model. The interpretable model can be interpreted by looking at the internal parameters of the model or through its summary statistics of the features.

Furthermore, in terms of interpretability we must also distinguish between **Model-specific** vs **Model-agnostic** and **Global** vs **Local** interpretation methods. Model-specific techniques are specific to a single type of ML model. They rely on the nature and model internals to provide explanations for specific predictions. Model-agnostic techniques are those that can be used in any ML model and are applied after the model has been trained. Generally, these agnostic methods work by analysing pairs of input and output features. Model-agnostic techniques are the most versatile techniques for interpretability of ML models as they are not restricted to the nature of a model. In the following we will look at some of the most popular techniques for this type of models.

Global Model-agnostic techniques. These techniques focus on the features that have the most impact on all of the model's target outcome. Overall, it provides a high-level understanding of how the model works and its decision-making. It can be useful not only to provide general interpretability to stakeholders but can also help the analysts understand the model in the development phases and assist in the feature selection. Some examples of the most common global model-agnostic techniques are:

- **Partial Dependacy Plots (PDP):** This technique explains the global behaviour of a model by showing the relationship of the marginal effect each feature has on the predicted outcome of the model. The plot shows whether the relationship between the target and a feature variable is linear, monotonic or more complex. This technique works on the basis that the features of the model are independent and uncorrelated with each other.
- **Feature interaction:** This technique allows us to interpret the model as the sum of the interaction of the features in the prediction. An example is the H-statistic by which the strength of the interaction is estimated by measuring how much of the variation in the prediction depends on the interaction of certain features.

EUROPEAN PARTNERSHIP

Co-funded by the European Union

- **Individual Conditional Expectation (ICE):** This technique is an extension of PDP which allows us to explain heterogeneous relationships. While PDP supports explanations of two characteristics, ICE we can only explain one characteristic at a time.
- **Permuted Feature Importance:** This technique establishes the significance of the features by measuring the increase in the prediction error of the model after permuting the feature values, thus breaking the relationship between the feature and the true outcome.
- **Global surrogate models:** This technique uses an interpretable model which is trained to try to approximate the predictions of a black box model. Through this interpretable model we can extrapolate conclusions about the black box model. The use of surrogates provides flexibility as it allows us to use different types of interpretable models depending on our needs. The main disadvantage is that the surrogate model draws conclusions from the black box model and not from the actual data.

Local Model-agnostic techniques. These techniques focus on providing specific interpretability on individual predictions of the model. In the case of SafeOPS, local model-agnostic interpretability can be useful in providing real-time information to an ATCO about why the model is predicting a go-around. It can provide valuable additional information to the ATCO to better assist the flight crew and enable more effective and efficient decision-making. In addition, the local interpretability of the model is important for models that are deployed in regulated industries such as aviation. It allows to perform a kind of audit of the model and provide justification as to why a certain prediction was made. Some examples of the most common local model-agnostic techniques are:

- **Local Surrogate models (LIME):** These techniques are based on the use of local surrogate models to explain the individual predictions of the black box machine learning model. Local interpretable model-agnostic explanations (LIME) is a specific application of the local surrogate models [7]. LIME focuses on training local surrogate models to explain individual predictions. Lime works by analysing how predictions change when variations of the input data are introduced into the model. It generates a new sample dataset consisting of perturbed samples and the corresponding predictions. On this sample dataset LIME then trains an interpretable model. The weights from the interpretable model are then used to provide explanations for the black box's local behaviour.
- **Shapley Values:** The Shapley value is a solution concept in cooperative game theory [8]. In game theory, we can define a player's Shapley value as the player's average marginal contribution in a game's payoff. In ML interpretability, the players of this cooperative game are replaced by the features of the ML model and the payoff by the model output. The Shapley value of a feature for a specific data point explains the contribution of the feature to a prediction at the specified data point. One of the advantages of using Shapley values is that it does not require access to the internals of the model but just requires the model as a predictor. The main disadvantage is that the calculation of shapley values is computationally expensive.

2.3 Human-interpretability

In recent years, human-interpretability has been raised of the importance of human-interpretability in the possible ML models to be used in ATM. With black box models ATM operators tend to not know "why" or "how" a certain prediction has been made decreasing their trust in the ML solution. Making AI interpretable has been identified as of key importance to ensure **trust** and **reliability** in the interaction between humans and AI-based systems. For this reason, several projects have been financed to research and advance interpretability in ATM systems. Among these projects we highlight the following:

EUROPEAN PARTNERSHIP

Co-funded by the European Union

- **ARTIMATION:** The goal of this project is to provide a transparent and explainable AI model through the use of different techniques. This project seeks to leverage human perceptual capabilities to better understand the AI algorithm with appropriate data visualisation in support of explainable AI (XAI), exploring in the field of ATM the use of immersive analytics to display the information [9].
- **TAPAS:** The project aims to investigate a systematic exploration of AI/ML solutions to increase levels of automation in specific ATM scenarios. Through different analysis and experimentation activities it aims to provide principles of transparency to enable the application of AI supported automation in ATM [10].
- **MAHALO:** This project investigates the impact on acceptability and performance of two different approaches: AI explainability and conformity of solutions proposed by AI. To do so the project is carrying out simulations with ATCOs testing a conflict prediction and resolution tool based on AI trained to maximise specific parameters (e.g. mileage, fuel consumption) and on AI trained to mimic real ATCOs behaviour [11].
- **AISA:** This project focuses on the effect of automation tools on the level of situational awareness of ATCOs. The project explores the domain-specific application of transparent and generalisable artificial intelligence methods [12].

Based on the work done by this research as well as research in other industries [13], we can highlight the different types of properties that different ML interpretation techniques can have as well as the criteria that establish what good interpretation technique is. Starting with the properties, we can identify the following ones:

- **Expressive power:** Refers to the type of information extracted from the interpretation technique. For example, decision trees or different weighted sums.
- **Translucency:** Refers to the degree in which an interpretation relies in looking in the inner workings of a model. Intrinsically interpretable models are considered highly translucent. Interpretation techniques that rely solely on analysing the inputs and outputs of the model are considered as low translucency.
- **Portability:** Refers the range of different ML models to which an interpretability technique can be used. Generally, low translucency techniques tend to have a higher portability. Surrogate models are considered to be the interoperability technique with the highest degree of portability.
- **Algorithmic complexity:** Refers to the computational cost of the interpretability technique.

Turning now to the factors that determine the quality of an explanation, we distinguish:

- **Accuracy:** The degree to which an interpretation technique generalises to unseen data.
- **Fidelity:** The extent to which the interpretation approximates the prediction of the black box model. this is one of the most important properties of any interpretation technique. If a black box model has high accuracy (performance) and the explanation has high fidelity, we can say that the explanation also has high accuracy.
- **Comprehensibility:** The degree to which the explanation of an interpretability technique is readily understandable to humans. This is one of the key elements of any explanation but one of the most difficult to measure. It is highly dependent on the user's operational knowledge and understanding of how ML solutions work.
- **Certainty:** Some ML models allow not only to make a prediction but also to provide some metric on the level of confidence in that prediction. It is generally it is preferable to use models and interpretability techniques that provide information on the uncertainty of a prediction.

EUROPEAN PARTNERSHIP

Co-funded by the European Union

- **Degree of importance:** To what extent the explanation reflects the importance of the features or parts of the explanation.
- **Novelty:** This concept is related to that of certainty. The greater the novelty, the more likely it is that a model will have low certainty due to lack of data.
- **Representativeness:** How many instances an explanation covers. As we have seen, explanations can cover the whole model (global) or represent only an individual prediction (local).

Taking all this into account, we can define from a human point of view what characterises a good explanation [5]. Human users generally prefer explanations that are **contrastive**. They are not so much interested in the explanation of a given instance as in being able to compare it with others and to establish how and why the prediction varies from one to another. In addition, ML models often use many features but the human user is not able to understand them all. So, a good explanation should be simple and provide a **small number of key factors** that affect a prediction. As mentioned before, the **user's background** must also be taken into account. Both technical and operational background must be understood when choosing how to present the explanations of a model in such a way that they are easily interpretable. Similarly, **users' prior beliefs** should be taken into account in explanations. A balance should always be sought between these prior beliefs and the model's operation. Results that go completely against the prior beliefs may generate distrust in the tool even if the performance is good but only focusing on them may prevent the discovery of new possible causes that were not taken into account before. Furthermore, there tends to be a greater interest in understanding the **extreme cases or outliers** as these are the ones that can have the greatest effect on trading. In many cases, the explanation should focus more on these extreme cases both from the perspective of the predictions and the features involved. Finally, the explanation should, as far as possible, seek **fidelity**, always in balance with the other desired objectives.

EUROPEAN PARTNERSHIP

Co-funded by the European Union

3 Case studies predictive results

3.1 Machine Learning Case Study 1 (ML_CS_01) - LightGBM model

This section presents an updated version of the ML model developed in deliverable 4.1. For this new version, some modifications have been made especially in the **correction of detected errors, new features** have been created, a **feature selection** has been made and again the **hyperparameters** of the model have been fine tuned. Furthermore, compared to the previous model, it has been extended to make predictions at different distances from the threshold (2NM, 4NM, 6NM and 8NM), thus providing additional levels of information to the ATCO. A more detailed description on the definition of this ML case study can be found in section 4 of deliverable 4.1 [3].

3.1.1 Data preparation and feature engineering updates

Error detection and correction. After a review of the results of the first version of the ML model and during subsequent EDAs (Exploratory Data Analysis), some errors in the data were discovered, that weren't picked up previously in the data preparation and processing pipeline. The main errors detected were:

- **On ground sensor:** The sensor data indicating whether the aircraft is on the ground or not is of very poor quality and very noisy. Although initially during the processing pipeline outliers are detected and eliminated, it was found that even on some occasions they still showed some kind of error. This caused, among other things, the runway to be misidentified and the aircraft landing time to be detected erroneously. The process of identifying and correcting outliers was redesigned and additional filters were added which consider that if a flight contains too many errors it should be discarded completely.
- **Multiple flights with same callsign:** One of the main assumptions in the processing pipeline was that over the course of a day there would only be unique callsigns for single flights. This is not always true and in cases where there are several flights for the same airport with the same callsign it can create flights with false or inaccurate parameters. Therefore, a flight splitting step has been established based on the timestamp of the data for the same callsign. As soon as a difference between timestamps of more than one hour is detected, the next part of the flight is considered as a new flight. Thus, establishing unique IDs per flight based on the date, the callsign and the splitting performed.

After correction, the raw data was passed back through the automated processing pipeline and a new final dataset was generated. Table 1 shows the new total number of departures, approaches and go-arounds identified in the period studied (2018-2020). It can be seen that although the total number of flights (departures and arrivals) has increased compared to the previous data set, see Table 2, there has been a decrease in the total number of go-arounds. This indicates that due to these errors the total number of go-arounds in the data was being overestimated, so it is likely that initial performance results were overestimated.

EUROPEAN PARTNERSHIP

Co-funded by the European Union

Table 1. New release of data: Number of approaches and go-arounds ML_CS_01

Airport	Number of approaches	Number of go-arounds	Go-arounds per 1000 approaches
EDDM	227044	646	2.85
EDDF	377712	1237	3.27

Table 2. Previous data release: Number of approaches and go-arounds ML_CS_01

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

New features. After several workshops with the ATCOs it was decided to explore the possibility of creating additional new features that could provide more operational information to the model. In total 6 new features were defined which can be seen in Table 3. For each point of the trajectory the **cross-wind** and **head/tail-wind** components were engineered using the available aircraft heading in ADS-B data as well as the wind speed and direction present in the METAR reports. The pseudo difference in depth of modulation (ddm) of the ILS signal was also calculated using the position information (latitude and longitude) of the aircraft as well as the altitude corrected with the QNH. The glideslope data for each runway were obtained from open sources. Finally, using the ADS-B information for each approach, it is identified if there is an aircraft in front of it and if this aircraft is in the approach phase or on the runway performing a take-off. In case there is an aircraft in front, closing time is engineered using the ground speed and heading of both aircraft as well as the haversine distance between them.

$$Closingtime(s) = \frac{distance}{V_1 * cos(brg - hdg_1) - V_2 * cos(brg - hdg_2)}$$

Where:

- V = Aircraft grounds speed (m/s);
- brg = Bearing between the two aircrafts (degrees);
- hdg = Aircraft track (degrees);
- distance = Haversine distance between aircrafts (m);

The formula is illustrated in Figure 1. It is important to mention that the formula only covers the closing time in a two-dimensional plane and does not regard vertical separation of aircraft.

EUROPEAN PARTNERSHIP

Co-funded by the European Union

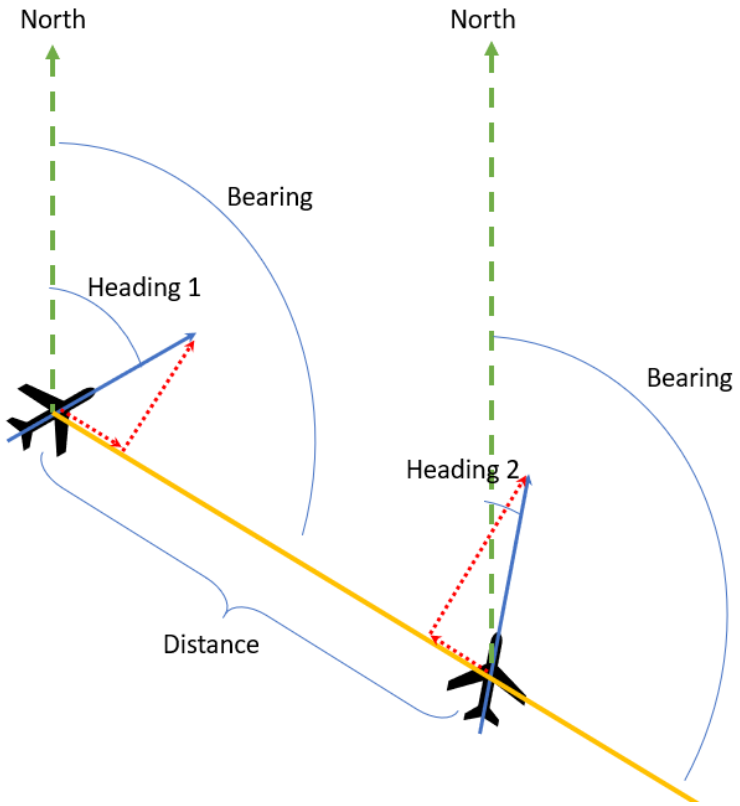

Figure 1: 2D Closing time

Table 3. New added features

Feature type	Feature name	Sampling	Source	Description
Weather data	Cross-wind	Distance from the threshold	Engineered feature	Cross-wind component
	Head/Tail-wind		Engineered feature	Head or tail wind component
Approach performance	Localizer ddm	Distance from the threshold	Engineered feature	Pseudo localizer difference in depth of modulation
	Glideslope ddm		Engineered feature	Pseudo glideslope difference in depth of modulation
Airport information	Aircraft in front	Closest available flight information	Engineered feature	Aircraft in front (approach, departure or none)
	Closing time		Engineered feature	2D Closing time in seconds with preceding approach or departure if any

Hyperparameter tuning. This process is carried out in the same way as for the first version of the model. We used a **Bayesian optimisation** for the hyperparameter tuning through the **Optuna** python library. For the hyperparameter tuning we decided to split the total data set (per airport) into a ratio

Co-funded by the European Union

of **80%-20%.** The distribution of the target variable (go-around/No go-around) in each partition was maintained. The 80% partition was then used to perform a cross-validation Bayesian optimisation of the hyperparameters. In this case as an optimisation metric, we decided to use the **Average Precision (AP)** from prediction scores. AP summarises the Precision-Recall curve as the weighted average of the precision achieved at each threshold, with the increase in recovery from the previous threshold used as the weight ([14] sklearn.metrics.average_precision_score).

$$AP = \sum_n (R_n - R_{n-1}) P_n$$

Where P_n and R_n are the precision and recall at the nth threshold. We decided to use this as the evaluation metric because it attempts to maximise the precision/recall ratio with the aim of having a model that can correctly identify all positive examples (go-arounds) and at the same time minimise the classification of negative examples as positive ones (false alarms). Once the best hyperparameters are found, they are used to re-train a final model. Finally, the 20% partition was used to validate the final model performance. Although this procedure is not the most optimal, it sufficiently reduces the bias and provides reliable model performance information. Other procedures, such as nested cross-validation, can further reduce bias but at a high computational cost.

Feature selection is the process by which the number of input variables is reduced when developing a predictive model. It is generally considered beneficial to reduce the number of input variables both on the one hand to reduce the computational cost of modelling and on the other hand, in some cases, to actually improve the performance of the model by eliminating non-informative or redundant features. Ensemble models such as the one used (LightGBM) are quite robust in terms of non-informative or redundant features being able to internally select the best features. However, it has been considered useful to introduce a feature selection process especially for the interpretability of the model. A smaller number of features will help us to better identify the most important features involved in predicting whether an aircraft performs a go-around or not. The feature selection process uses the training dataset from the 80%-20% split to ensure that no data is leaked from the validation set to the training phase of the model. The reduced data set is then through **k-fold cross-validation** is trained using LightGBM model (without hyperparameter tuning). For each fold, the feature importance ranking is obtained using **SHAP values** and at the end averaged over all the folds. A subset of the features with the lowest average impact on the model is eliminated. This process is repeated until the performance AP score becomes significantly worse than the best one achieved through all previous iterations. Finally, the data reduced dataset is then hyperparameter tuned.

3.1.2 Predictive results

Table 4 shows the models performance for EDDM for all established prediction points as well as for the model with and without feature selection enabled. Compared to the initial version of the model (SafeOPS D4.1 [3]), we can see that there has been a slight decrease in overall performance. This supports the idea that the errors detected in the data were generating anomalous flights considered as go-arounds that the model was identifying correctly. Even so, the performance of the model can be considered as good, especially considering the unbalanced nature of the data. It can also be seen as expected that the closer to the runway threshold the better the performance of the models, with a gradual decrease in performance as we move away. The best performance results are obtained at 2NM and where we are able to detect 34% of all the go-arounds but with a precision of 88%. This means that although most go-arounds are not detected for those that are, the model presents a high confidence in the prediction. This translates significantly into a low level of false alarms which would be a distraction and annoyance for ATCOs. We can see that for the prediction points at 6NM and 8NM

EUROPEAN PARTNERSHIP

Co-funded by the European Union

the total number of go-arounds detected is below 10%, although in a positive way it can be seen that in both cases there is a high precision, 91% and 70% respectively. Again, seeing how the model prioritises minimising the number of false alarms. In the models where an initial selection of features is applied, it can be seen that equivalent performance values can be obtained. In the case of the 2NM prediction point, it can be seen that with only 22 (less than a quarter) features an equivalent performance can be obtained, allowing the possibility of using a lighter and simpler model which will be easier to interpret.

Table 4. ML_CS_01 EDDM model results

EDDM

All features results							
Number of features	**Prediction point**	**Go-around**	**Precision**	**Recall**	**F1-score**	**ROC-AUC**	**PR-AUC**
152	2 NM	True	0.8800	0.3411	0.4916	0.8908	0.4598
		False	0.9981	0.9999	0.9990		
122	4 NM	True	0.8710	0.2093	0.3375	0.7210	0.2413
		False	0.9977	0.9999	0.9988		
96	6 NM	True	0.9091	0.0775	0.1429	0.6551	0.1243
		False	0.9974	0.9999	0.9987		
66	8 NM	True	0.7000	0.0543	0.1007	0.6826	0.1032
		False	0.9973	0.9999	0.9986		
Feature selection results							
Number of features	**Prediction point**	**Go-around**	**Precision**	**Recall**	**F1-score**	**ROC-AUC**	**PR-AUC**
22	2 NM	True	0.8235	0.3256	0.4667	0.8475	0.4301
		False	0.9981	0.9998	0.9989		
102	4 NM	True	0.8667	0.2016	0.3270	0.7690	0.2447
		False	0.9977	0.9999	0.9988		
66	6 NM	True	0.8889	0.0620	0.1159	0.6548	0.1265
		False	0.9973	0.9999	0.9987		
36	8NM	True	0.6363	0.0543	0.1000	0.6700	0.0992
		False	0.9973	0.9999	0.9986		

Table 5 shows the models performance for EDDF for all established prediction points as well as for the model with and without feature selection enabled. Like for EDDM we can see that compared to the initial version of the model (SafeOPS D4.1 [3]), there has been a slight decrease in overall performance. However, for the prediction point at 2NM in the EDDF case, the model is able to detect 40% of all go-arounds with high precision 89%. Compared to EDDM at the 6NM and 8NM prediction points in the

EUROPEAN PARTNERSHIP

Co-funded by the European Union

worst case the model is able to predict up to 15% of all go-arounds. It can also be seen that for the different prediction points a high level of precision is maintained ensuring that the model has low number of false alarms. The use of the feature selection method has also allowed the generation of simpler and lighter models with similar performances. In the case of the 2NM prediction point, the model with less than half of the available features is able to detect 37% of all go-arounds with a 89% precision. As discussed above, this helps us to reduce the noise input into the model from non-value adding features and facilitates subsequent interpretability. An analysis of the interpretability of the models (EDDM and EDDF) through the feature importance feature importance ranking can be found in the next section (section 4).

Table 5. ML_CS_01 EDDF model results

EDDF

All features results							
Number of features	**Prediction point**	**Go-around**	**Precision**	**Recall**	**F1-score**	**ROC-AUC**	**PR-AUC**
152	2 NM	True	0.8850	0.4049	0.5556	0.8448	0.4511
		False	0.9980	0.9998	0.9989		
122	4 NM	True	0.9118	0.2510	0.3937	0.7495	0.2976
		False	0.9975	0.9999	0.9987		
96	6 NM	True	0.7846	0.2065	0.3269	0.7096	0.2320
		False	0.9974	0.9989	0.9986		
66	8 NM	True	0.9024	0.1498	0.2569	0.6776	0.2024
		False	0.9972	0.9999	0.9986		
Feature selection results							
Number of features	**Prediction point**	**Go-around**	**Precision**	**Recall**	**F1-score**	**ROC-AUC**	**PR-AUC**
62	2 NM	True	0.8846	0.3725	0.5242	0.8559	0.4512
		False	0.9979	0.9998	0.9989		
112	4 NM	True	0.8611	0.2510	0.3887	0.7989	0.3119
		False	09975	0.9999	0.9987		
86	6 NM	True	0.7937	0.2024	0.3225	0.7320	0.2367
		False	0.9974	0.9998	0.9986		
46	8NM	True	0.7647	0.1579	0.2617	0.6867	0.1896
		False	0.9972	0.9998	0.9985		

EUROPEAN PARTNERSHIP

Co-funded by the European Union

3.2 Machine Learning Case Study 2 (ML_CS_02) - Predictive results

This section presents the ML model developed for the second defined case study (ML_CS_02). Initially, this ML solution was intended to provide predictions every "x" number of seconds. Eventually it was decided to re-define the continuous prediction as the possibility to make predictions **at constant intervals of distance from the runway's threshold**. Primarily to maintain consistency and allow comparison with ML_CS_01 and secondly to facilitate the development of the neural network based solution. A more detailed description of the definition of this ML case study can be found in section 4 of deliverable 4.1 [3].

3.2.1 Data preparation and Feature engineering

Redefining the problem so that predictions are made at periodic distance intervals allowed us to take advantage of the infrastructure and **automated data pipelines** already developed for ML_CS_01. We decided to use the **0.5NM sampling** for this case study. This means that for an aircraft flying at an average speed of 60-90 m/s in the final approach phase the ATCO can expect a prediction **approximately every 15-10 seconds** giving a high level of granularity in the results. The Long Short Term Memory (**LSTM)** is a type of recurrent neural network (RNN) which uses sequential (or time series) data. This type of model is very powerful as it is often able to automatically extract features from past events. The main peculiarity of LSTMs is that they are a bit more demanding than other models in terms of **data preparation** and formatting. In contrast to models such as LightGBM which uses tabular format data (number of samples x number of features) an LSTM model requires a 3D matrix with the following form:

- **Samples:** refers to the total number of observations fed into the LSTM. In this case, the series has one observation every 0.5NM for every approach
- **Sequence:** refers to the past data window the LSTM has access to make a prediction.
- **Features:** number of data columns selected as potential features.

As for the features used, a prior selection of features had to be made for the case study. Table 6 shows the final list of features used. Unlike LightGBM; LSTMs are very sensitive to the type of data and format that is entered as a feature. It doesn't handle categorical features naturally so these must be encoded. There are several techniques such as One-hot-encoding or Ordinal encoding. After exploring them it was decided that none of them were suitable for our data and that any type of encoding could introduce noise to the model. It was therefore decided to remove these from the final list of features. Therefore, only numerical features were used. In addition, certain features such as "Airport information" are calculated for the specific prediction point. In this case and with a dynamic prediction point it means that they would require to be calculated for all possible distances. These will introduce a high computational cost so they were also discarded. Finally, the selection is focused on the "Approach performance" features that as seen in ML_CS_01 (see section 4) are the generally the most relevant in the prediction at 2NM, 4NM, 6NM and 8NM. Finally, the LSTMs do not handle varying ranges in feature values, so a **Standard scaler** was applied which standardize features by removing the mean and scaling to unit variance ([14] sklearn.preprocessing.StandardScaler).

EUROPEAN PARTNERSHIP

Co-funded by the European Union

Table 6. ML_CS_02 Description of features

Feature type	Feature name	Sampling	Source	Description
Flight information	Approach attempt	Static	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	Nearest available METAR report	Available in data	Measured wind speed
	Wind direction		Available in data	Measured wind direction
	Temperature		Available in data	Measured Temperature
	Visibility		Available in data	Measured visibility
	Dew point temperature		Available in data	Based of the lowest clouds that cover more than half of the sky relative to the ground
	Ceiling height		Engineered feature	Based of the lowest clouds that cover more than half of the sky relative to the ground
	Cross-wind	Every 0.5NM from runway threshold	Engineered feature	Cross-wind component
	Head/Tail-wind		Engineered feature	Head or tail wind component
Approach performance	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	Descent vertical rate 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

Co-funded by the European Union

	Localizer ddm		Engineered feature	Pseudo localizer difference in depth of modulation
	Glideslope ddm		Engineered feature	Pseudo glideslope difference in depth of modulation
	Distance		Engineered feature	Haversine distance to threshold in nautical miles

3.2.2 Long Short-Term Memory (LSTM) model

Recurrent Neural Networks (RNN) are a special kind of neural networks specially design to work with sequence/temporal data. RNN neurons work by having a cellular state/memory, and the input is processed according to this internal state, which is achieved with the help of loops with in the neural network, see figure 2. LSTM networks are a type of RNN capable of learning order dependence in sequence prediction problems. A RNN/LSTM network layers can be designed in a multitude ways which can impact how features are learned and extracted from the data. Among the different types of layers, the most important are usually:

- **LSTM:** RNN layer that works with sequential data and is able to learn features of events from long to short time data.
- **Dropout:** This consists of randomly setting a fraction rate of input units to 0 at each update during training phase. It is useful to try and prevent overfitting of the model.
- **Dense:** A layer with all its neurons are connected to every neuron of its preceding layer. Neurons of the dense layer perform matrix-vector multiplication.

Due to the ability of LSTMs to work with sequences it is believed to be well suited to our case study. The architecture of the RNN/LSTM has been designed from scratch, iterating and adjusting the hyper parameters of the network and its design to adequately minimise the maximum loss error. Finally, it was decided to set the amount of information passed to the LSTM from past observations to be 4 data points (previous 2 NM).

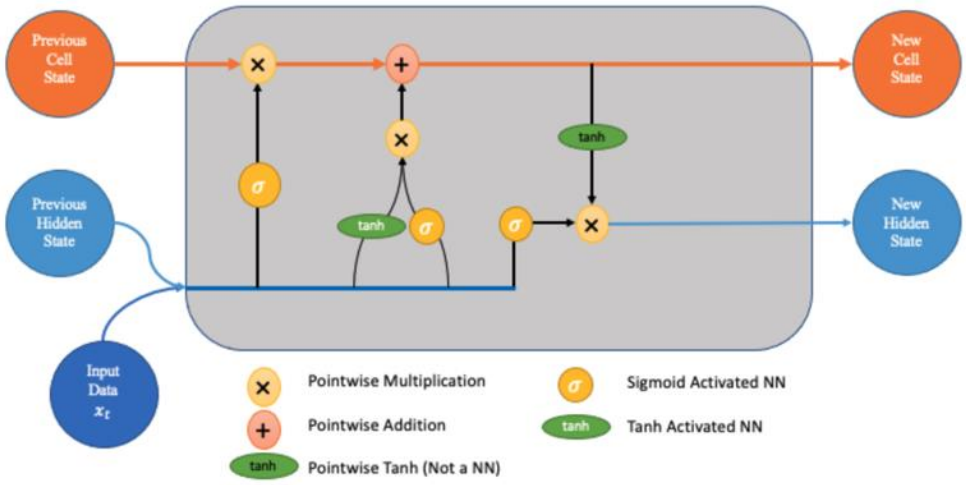

Figure 2: Basic LSTM Layout from [15]

Co-funded by the European Union

3.2.3 Predictive results

Initially and in order to validate the possibility of using a LSTM network for go-around prediction, we trained a LSTM network to perform **2NM threshold predictions** for EDDM and EDDF and compared it with the results obtained using the hyperparameter tuned LightGBM model. The results of this validation can be seen in Table 7 and Table 8. We can see that with a LSTM network without any unbalanced data handling techniques (e.g., under or over sampling) and with a total number of **26 features** we achieve very similar performance results to those of the LightGBM models. This validates the idea of the possibility of using such networks for gait prediction.

Table 7. EDDM - LSTM validation 2 NM

EDDM					
Prediction point	**ML model**	**Precision**	**Recall**	**ROC-AUC**	**PR-AUC**
2 NM	LightGBM	0.88	0.33	0.89	0.46
	LSTM	0.85	0.35	0.76	0.45

Table 8. EDDF - LSTM validation 2 NM

EDDF					
Prediction point	**ML model**	**Precision**	**Recall**	**ROC-AUC**	**PR-AUC**
2 NM	LightGBM	0.89	0.41	0.84	0.45
	LSTM	0.92	0.42	0.79	0.50

We then proceed to develop for EDDM and EDDF a **global prediction model** based on LSTMs. This model will be able to generate predictions every 0.5NM from 8NM to 2NM from the runway threshold. For this purpose, for each flight, sequences are generated every 0.5NM in which information relating to the last 2 NM is included. For example, at 3NM the sequence will contain information related to the flight features from 5NM to 3NM. For the training and validation of the LSTM, an 80-20 data split was used. Table 9 presents the results for what we have termed **"Global performance"**. When interpreting these results, one should bear in mind that they refer to the predictions made for all the splitted flight segments under study. Therefore, a recall of xx% does not mean that only that percentage of go-arounds are identified. For this, one has to look at the results of the so-called **"Local performance"**.

Table 9. LSTM Global performance

Global LSTM performance				
Airport	**Precision**	**Recall**	**ROC-AUC**	**PR-AUC**
EDDM	0.8608	0.1401	0.6655	0.1717
EDDF	0.8655	0.2454	0.7217	0.2883

EUROPEAN PARTNERSHIP

Co-funded by the European Union

The results shown in Table 10 refer to specific predictions made at particular distance points in the final approximation phase. These results can be directly compared to those of ML_CS_01. It is interesting to see how, despite being a general model for the whole final approach phase, the performance obtained at 2NM, 4NM, 6NM and 8NM is similar or better to the **specific LightGBM models** developed both in EDDM and EDDF. Although the vast majority of go-arounds go undetected, the possibility of using a unique sequencing model such as the LSTM opens up the possibility of **exploiting the predictions** it generates as an additional input in the go-around analysis process. The trend could be studied and we could alert the ATCO of those flights that, without being classified as go-arounds, show an upward trend in prediction and could therefore be worthy of additional checks. In any case, these lines of reasoning are further elaborated in the work done in WP3 related to the **Risk Framework**.

Table 10. Table 10. LSTM Local performance

Local LSTM performance

EDDM				
Prediction point	**Precision**	**Recall**	**ROC-AUC**	**PR-AUC**
2NM	0.9167	0.3411	0.7925	0.4102
4 NM	0.8571	0.1395	0.6681	0.1732
6 NM	0.8182	0.0698	0.6302	0.0896
8 NM	0.8333	0.0388	0.6138	0.0639
EDDF				
Prediction point	**Precision**	**Recall**	**ROC-AUC**	**PR-AUC**
2NM	0.8718	0.4130	0.7992	0.4695
4 NM	0.8667	0.2632	0.7359	0.3061
6 NM	0.8727	0.1943	0.6842	0.2282
8 NM	0.7660	0.1457	0.6761	0.1807

EUROPEAN PARTNERSHIP

Co-funded by the European Union

4 Model Interpretability results

4.1 SHAP (SHapley Additive exPlanations)

SHAP (SHapley Additive exPlanations) is an interpretability technique based on Shapley values developed to provide explanations of individual predictions [16]. SHAP focuses on trying to explain the prediction of a particular instance by calculating the contribution of each feature to the prediction. It calculates Shapley values from **coalition game theory** where the feature values of a data instance act as players in a coalition. The Shapley values tell us how to fairly distribute the prediction among the different features. An innovation that SHAP brings over other similar techniques (e.g., LIME) is that the Shapley value explanation is represented as an additive feature attribution method, a linear model [6]. In addition, SHAP also features an implementation for tree-based machine learning models, such as decision trees, random forests and gradient boosting trees [17].

For the interpretation of the results of the models developed we will mainly use two types of visualisations: **Global feature importance** and **Local explanation summary**. Global feature importance is computed as the average of the absolute Shapley values per feature across all the data. This way of calculating global feature importance in SHAP is an alternative to permutation feature importance (see section 2). Permutation feature importance is based on the decrease in model performance whereas SHAP is based on the magnitude of feature attributions. Local explanation summary plots provide a combination of feature importance and feature effects. Each point on the plot is a Shapley value for a feature and a individual instance. The y-axis shows the different features while the x-axis shows the calculated Shapley value. The colour represents the actual value of the feature for each instance from low to high. Moreover, if overlapping occurs jitter in the y-axis is added to get a sense of the distribution of the Shapley values per feature.

4.2 Interpretability ML_CS_01

4.2.1 EDDM interpretability analysis

To carry out the interpretability study of the models developed, we have used the models with the **feature selection**, thus reducing the original characteristics used in the training of the models. This way simpler models will be studied, facilitating the calculation of the shapley values, as well as reducing the introduction of possible features that introduce noise in the model and that, due to the stochastic nature of the models, could misrepresent the performance of the model.

Starting with the model developed for **EDDM at 2NM**, we can see the SHAP feature importance results in Figure 3. It is important to always bear in mind in these analyses that from the results of the interpretability techniques it is **not possible to derive causal relationships** between the features and the predicted event (go-arounds), but merely to infer the same. This is why the analysis should not focus on specific features and/or on the ranking generated by SHAP, but rather on a more **global perspective**. In this way, more reliable conclusions can be drawn. In Figure 2, we can see the preponderance of features related to the **"Approach performance"**, indicating that they have the strongest cause-effect relationship in the go-around prediction for our model. In particular it is interesting to see how in general for the different sampling points the same "Approach performance" feature types are repeated showing the importance especially of the vertical speed (hdot_mds) and energy level in the go-around scenarios. In addition, we can also observe interesting situations such as

EUROPEAN PARTNERSHIP

Co-funded by the European Union

that for a number of flights the cause of their being labelled as go-arounds is influenced by their high vertical speed or energy levels at 2NM or by having a small closing time with the aircraft in front of them. Features related to the number of approaches on the same runway in the last 30 minutes (rwy_ARR_10_mins) is also observed as relevant, indicating the effects airport traffic volume hast as well as the flight's callsign indicating some perceived relationship between it and the probability of a go-around.

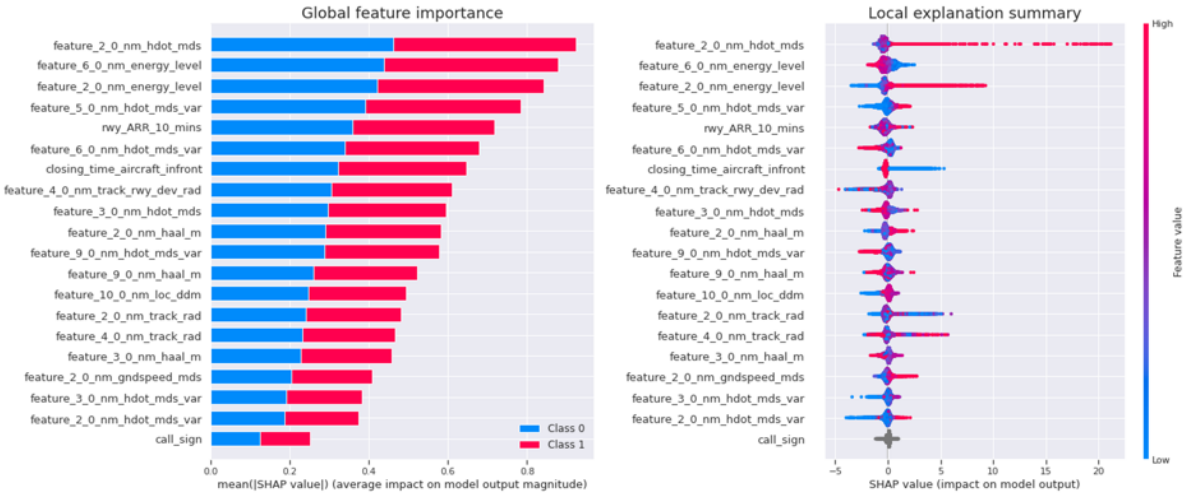

Figure 3: EDDM 2NM - interpretability results (Top 20 features)

Figure 4 shows the SHAP features ranking for the **4NM model at EDDM**. Comparing with the results for 2NM we can see that in this case "Flight information" and "Airport information" features have a higher relevance in the prediction. We see how the main features are the callsign and the runway id indicating the possible effect that airline/crew familiarity with the airport has or indicating how the approach complexity in some runways can be higher than in others. In this case we can see many "Airport information" features, especially the total number of previous go-arounds at the airport (airport_GA_60_mins) and the number of previous arrivals at the approaching runway (rwy_ARR_10_mins). In both cases, high volumes generally favour the positive prediction of go-arounds. "Approach performance" features are also identified as in the case of 2NM, although with lesser relevance. It is still interesting to note that these are generally related to vertical speed, ground speed and aircraft track.

EUROPEAN PARTNERSHIP

Co-funded by the European Union

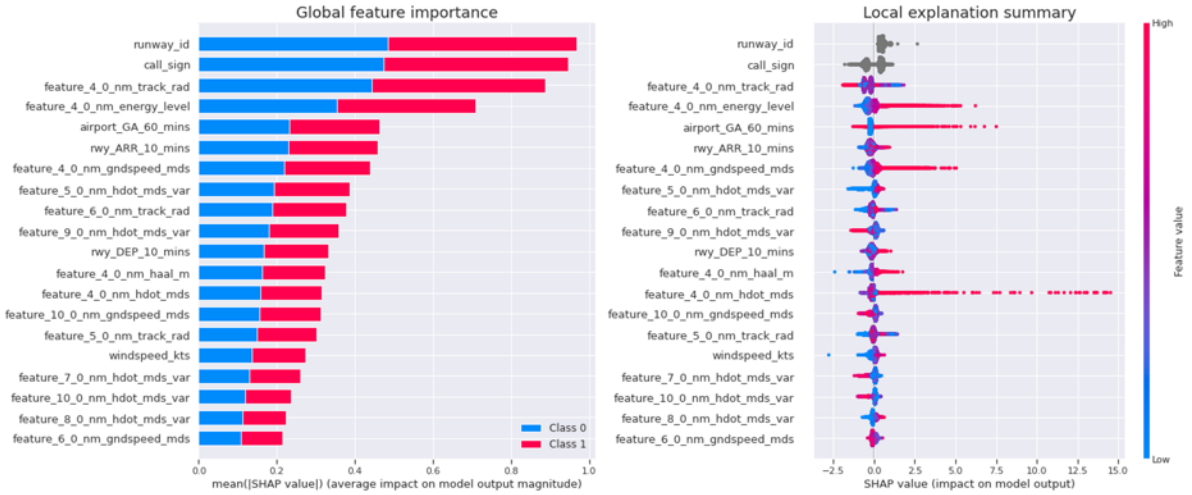

Figure 4: EDDM 4NM - interpretability results (Top 20 features)

Figure 5 shows the SHAP features ranking for the **6NM model at EDDM**. At 6NM it can be confirmed that there is a trend the further away from the threshold where the "Approach performance" features loose relative importance in the prediction. This is because the further away from the threshold, in most cases, the aircraft performance values (e.g., ground speed or vertical speed) are very similar between aircraft with and without go-arounds. Although we can still see the importance of features such as the aircraft track and energy level. We can also see how the "Flight information" and "Airport information" features gain relative importance. As with 4NM the callsign and runway id presents a high feature importance for the model. There are also temporal features (week and hour) showing a possible temporality in the occurrence of go-arounds throughout the day and year.

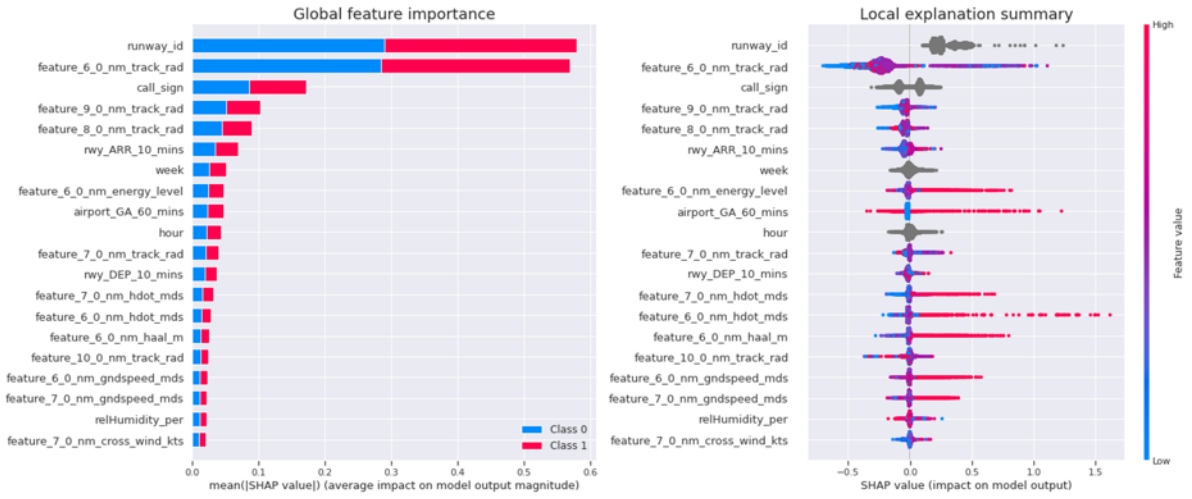

Figure 5: EDDM 6NM - interpretability results (Top 20 features)

Finally, Figure 6 shows the SHAP features ranking for the **8NM model at EDDM**. In this case we see with the feature wind speed appears as the most important. The "Weather data" features had only been relatively important so far, but here again we see that the further away from the threshold the "Approach performance" features become less important. Features related to "Weather data", "Airport information" and "Flight information" gain importance. From these it can be inferred that the model tries to make predictions more based on the operational context information (weather and

EUROPEAN PARTNERSHIP
Co-funded by the European Union

airport traffic) rather than on the aircraft performance data of each aircraft. This may be one of the explanations why the model performance decreases with distance to the threshold.

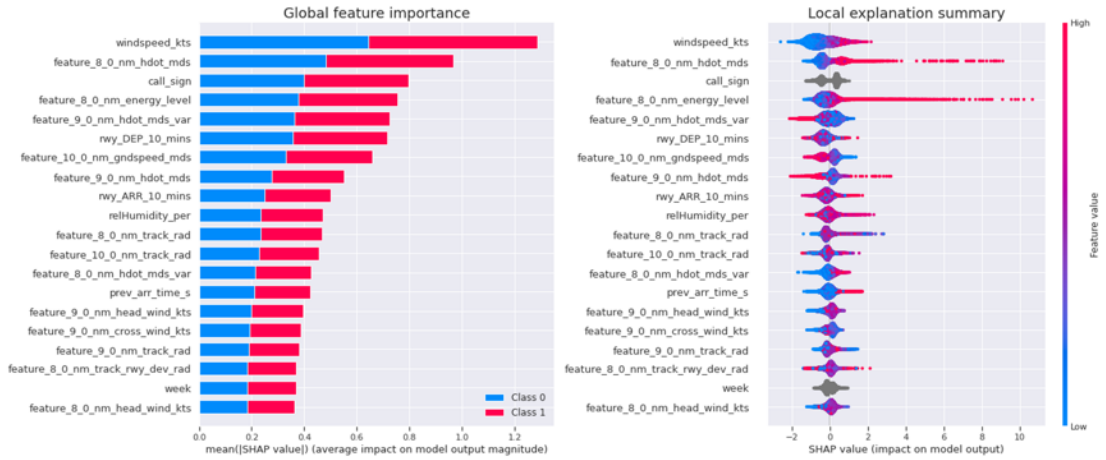

Figure 6: EDDM 8NM - interpretability results (Top 20 features)

4.2.2 EDDF interpretability analysis

Analogously to the case of EDDM, the models to which the feature selection was applied were used for the EDDF interpretability analysis. Figure 7 shows the SHAP feature for the **2NM model at EDDF.** Similar to the EDDM case, the "Approach performance" features show the highest importance, especially vertical speed, energy level and aircraft track. "Airport information" features such as the total number of previous go-arounds at the airport and number of previous arrivals at the approaching runway also present high importance. On the other hand, also important "Flight information" features are shown, especially the week, and "Weather data" such as wind speed. Also, it can be seen how in general for the different sampling points the same types of "Approach performance" characteristics are also repeated showing the importance especially of vertical speed (hdot_mds), energy level and aircraft track in the go-around scenarios. In addition, we can also observe that for a number of flights the cause of being labelled as go-arounds is influenced by their high vertical speed or energy levels at 2NM.

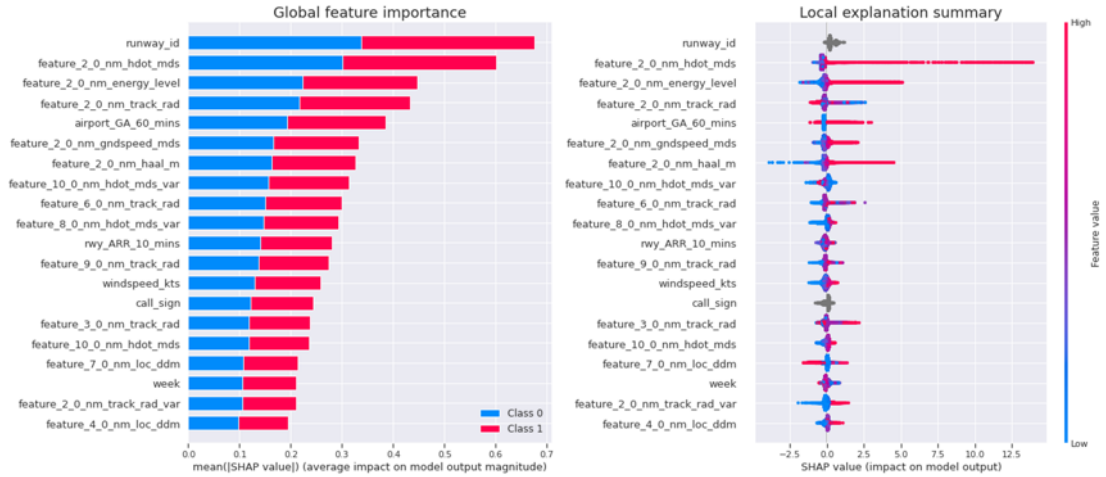

Figure 7: EDDF 2NM - interpretability results (Top 20 features)

Co-funded by the European Union

Figure 8 shows the SHAP features ranking for the **4NM model at EDDF**. Again, comparing with the results for 2NM we can see that "Flight information" and "Airport information" features start to show a higher relevance in the prediction. We see how the among the main features are the callsign and the total number of previous go-arounds at the airport (airport_GA_60_mins). We also see the relevance of the week of the year as well as "Weather data information" such as wind speed and dew point. As in all other prediction points the "Approach performance" features still dominates the top 20 especially with vertical speed, energy level and aircraft track. It is interesting to see how the closing time also appear here and show that a small value of the closing time favours the go-around classification of a flight.

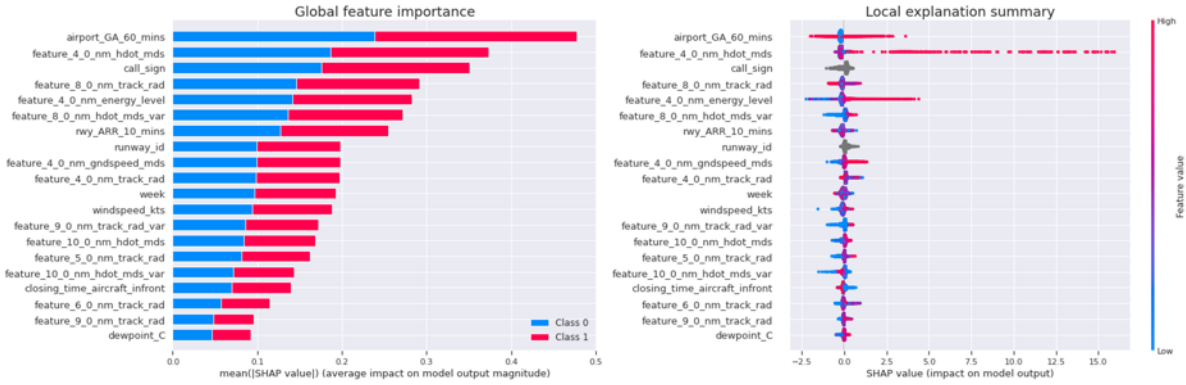

Figure 8: EDDF 4NM - interpretability results (Top 20 features)

Figure 9 shows the SHAP features ranking for the **6NM model at EDDF**. it can be seen again that as we move away from the runway threshold, the "Approach performance" feature becomes less important. Among the main features we have "Weather information" such as wind speed, "Flight information" such as the aircraft callsign and "Airport information" such as the total number of previous go-arounds at the airport and the number of approaches on the same runway. In this case we can see how a high vertical speed especially at 6 NM is marked as very relevant in a series of flights for their classification as go-arounds.

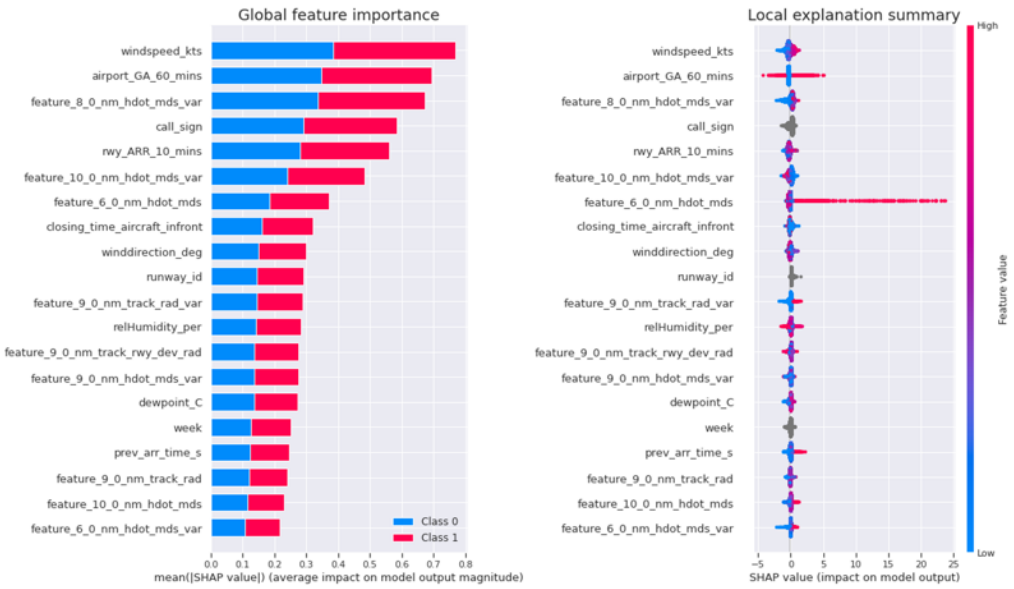

Figure 9: EDDF 6NM - interpretability results (Top 20 features)

EUROPEAN PARTNERSHIP

Co-funded by the European Union

Finally, Figure 10 shows the SHAP features ranking for the **8NM model at EDDF**. In this case as in that of EDDM the feature wind speed appears as the most important and again we see that the further away from the threshold the "Approach performance" features become less relevant. Features related to "Weather data", "Airport information" and "Flight information" gain importance. From these it can be again inferred that the model tries to make predictions more based on the operational context information (weather and airport traffic) rather than on the aircraft performance data of each aircraft. This may be one of the explanations why the model performance decreases with distance to the threshold. Lastly, it should be noted how in this, similar to the previous one, high vertical speeds at 8NM are identified as very relevant in a series of flights for their classification as go-arounds

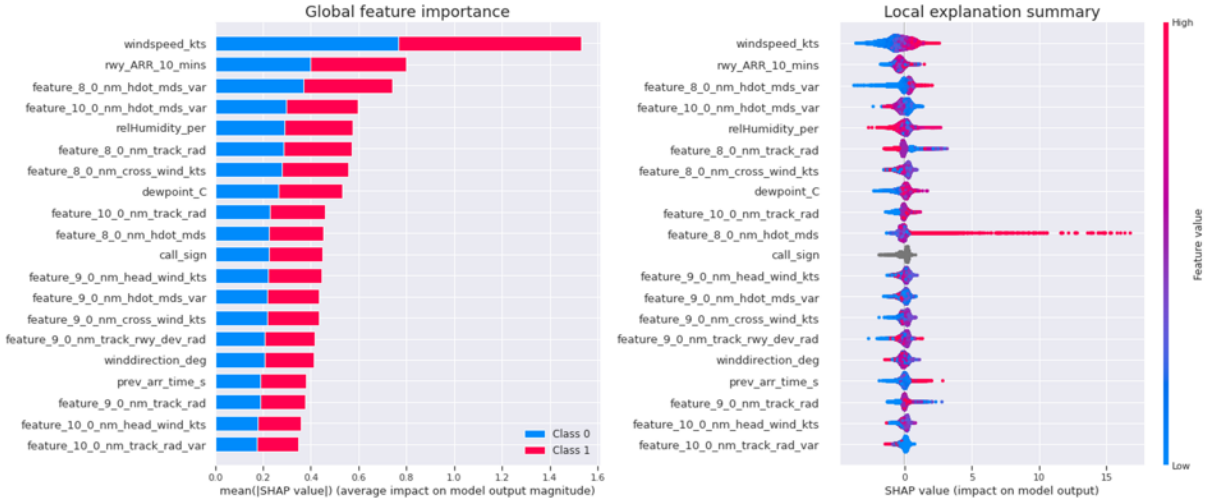

Figure 10: EDDF 8NM - interpretability results (Top 20 features)

4.3 Interpretability ML_CS_02

Figure 11 and Figure 12 show the feature importance ranking of the LSTM models developed for EDDM and EDDF respectively. As this is a more complex type of ML model, a technique from the SHAP library called **"Expected gradients"** was used to extract the feature importance. This technique is an extension of the integrated gradients method [18], which is a feature attribution method designed for differentiable models based on an extension of Shapley values to infinite player games (Aumann-Shapley values).

The first thing to take into account when evaluating the importance of the features in this case study is that, unlike those of ML_CS_01, they are not located at a single point but rather throughout the 8NM to 2NM. That is why it is not easy to draw conclusions about high or low values of this feature and their effect in the go-around prediction: This will be dependant on where it is located. Comparing the results of both models we can see how in both cases the hour and temperature ranked in the top five features. Of particular interest is the latter because although at first sight it may appear to it should have a lesser impact than another such weather information such as wind, it may hide relationships with other important meteorological phenomena (e.g. stormy weather). Also interestingly in the case of EDDM one can see the high importance of height (haal_m). The latter for EDDF shows, on the contrary, low importance. In the case of EDDF we can see that wind speed and head wind show a high importance while not so high for EDDM.

EUROPEAN PARTNERSHIP

Co-funded by the European Union

Comparing the results of ML_CS_02 with those of ML_CS_01, some interesting variations can be seen. For example, in ML_CS_01 aircraft track generally showed a high relevance but not so much for the LSTM based models used in ML_CS_02. In contrast, for the LightGBM models from ML_CS_01, the many of the "Weather information" features such as temperature or dew point do not appear at any prediction point as one of the top 20 features. Where both types of models coincide is that in all cases ground speed is usually one of the most important features. This allows us to infer with some degree of certainty its importance in detecting and predicting a go-around approach.

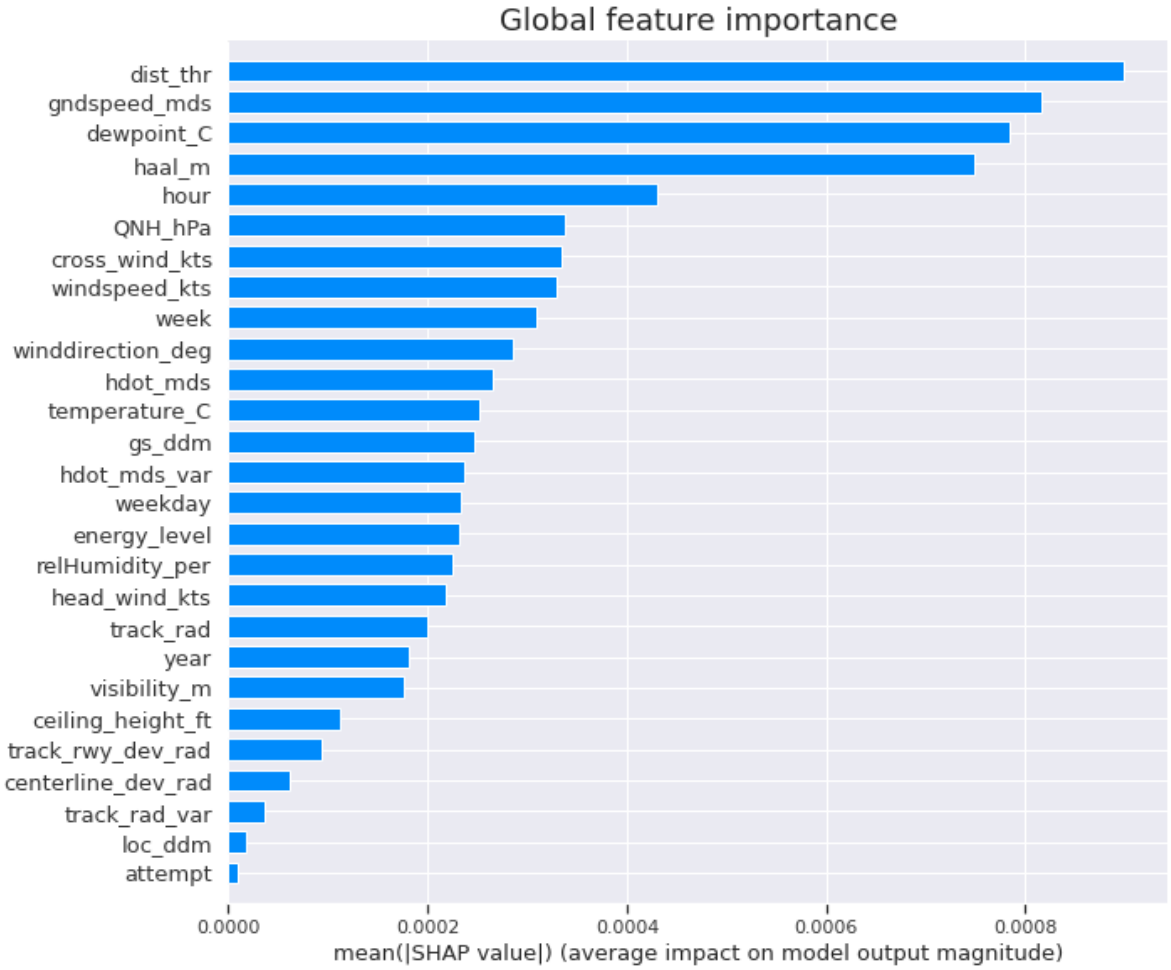

Figure 11: EDDM - LSTM Feature importance

Co-funded by the European Union

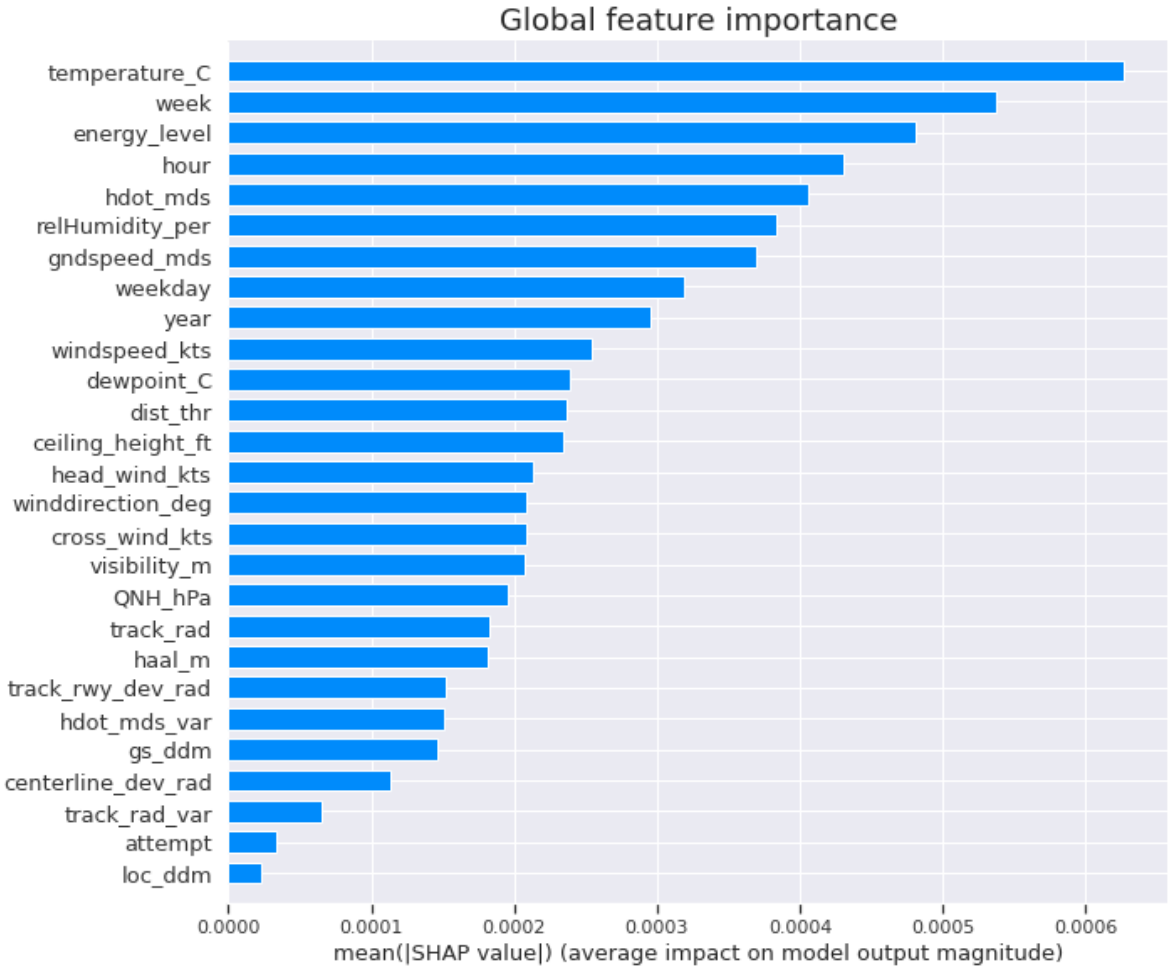

Figure 12: EDDF - LSTM Feature importance

Co-funded by the European Union

5 Conclusions

In this deliverable a technical summary of the second phase of **Work Package 4 (WP4)** of the SafeOPS project has been presented. WP4 is dedicated to the development of all tasks related to the **technical development of a data-driven predictive solution** for the prediction of go-around situations in airport operations. The work carried out in this second phase consisted in the development of the **final predictive results** for both ML_CS_01 and ML_CS_02 as well as an analysis of the **interpretability** and **explainability** of the ML solutions developed.

With respect to the objectives set out for WP4, we can say that we successfully developed different ML solutions which, taking into account the **unbalanced nature of the go-around events**, have been able to adequately provide predictions in advance. At the two airports investigated (**EDDM** and **EDDF**), models developed have been able to identify in some cases 34%-40% of all go-arounds with a precision of close to 90%. Although the recall is not particularly high, through previous work both in (SafeOPS D2.1 [1]) and (SafeOPS D4.1 [3]) we identify that due to the limitations of data used, such as ADS-B, it would be very difficult or impossible to have high levels of recall, as **ATC induced go-arounds** (e.g. the runway is blocked or other traffic requiring immediate priority) would not be adequately reflected in the data and therefore the model would not be able to identify these events correctly. In addition, the work has also focused on investigating the interpretability of the models developed. For this purpose, a study of the **feature importance,** using the **SHAP** python package, for each model and airport was carried out. From these, of the main conclusions drawn we find the preponderance of features related to **"Approach performance"** such as vertical speed, ground speed or energy level. During the development of the **Predictive Layer** for SafeOPS the main takeaways and lessons learned for future developments of ML solutions are:

- **Data quality and validation:** we have seen first-hand the importance of data in ML models. On the one hand, if the event to be predicted is not fully captured in these, the performance of the models will be limited. In addition, the importance of continuous validation of data quality to ensure the veracity of the model's output. Thanks to this we were able to detect previous errors in the data processing stages, correct them and re-evaluate the models. We believe that data validation will be a key process in all future developments of possible ML tools in ATM in the field of certification.
- **Human-centered design and Performance assessment:** From the outset, it has been important to maintain a user-centred development. For example, establishing the need for low false alarms from a user perspective often led us to focus model development on precision optimisation in a kind of trade-off between precision and recall. Understanding the needs of the user and their operational context is vital to the successful development of an ML tool.
- **Human-Interpretability:** As we have seen throughout this report, the interpretability of models can be very important when they are helping to make decisions in critical situations. It is important that stakeholders have some understanding of how the model is generating the predictions, what it is based on and what they can expect from it. In this way, the user can more effectively incorporate these predictions into their decision-making process and develop trust with the ML tool.

The results of all the technical work done in WP4 will be delivered to **WP3**, presented in workshops with stakeholders and will help to further develop and define the **Risk Framework**. In addition, it will also support the **Experimental/validation plan** developed in **WP2**. Finally, at the beginning of the

EUROPEAN PARTNERSHIP

Co-funded by the European Union

project, **functional** and **non-functional requirements** were defined for the prediction tool. A final assessment of the fulfilment of these requirements is provided in Table 11.

Table 11. ML related Functional and Non-Functional Requirements

Requirement type	Requirement category	Requirement ID	Requirement definition	Additional information	Status
Functional Requirements	High Level Functionality	FR.C.01	The system shall output a probability of an approaching A/C performing a GA (also referred to as prediction), given data that describes the A/C's approach and the conditions thereof as input.	Achieved	Deliverable 4.1 - Section 5 "Predictive modelling" Deliverable 4.2 - Section 4 "Model Interpretability results"
		FR.C.03	The system shall provide quantifiable metrics on the performance quality of the prediction.	Achieved	Deliverable 4.1 - Section 5 "Predictive modelling" Deliverable 4.2 - Section 3 "Case studies predictive results"
		FR.C.04	The system shall provide information on the contributing factors, responsible for the prediction.	Achieved	Deliverable 4.2 - Section 4 "Model Interpretability results"
	Timing of Prediction	FR.T.01	The prediction shall be computed every prediction update rate seconds in between a minimum distance and maximum distance measured from the runway threshold.	Achieved	Deliverable 4.2 - Section 3 "Case studies predictive results"
		FR.T.02	The prediction shall be computed at specified distance increments in between a minimum distance and maximum distance measured from the runway threshold.	Achieved	Deliverable 4.2 - Section 3 "Case studies predictive results"

Co-funded by the European Union

Requirement type	Requirement category	Requirement ID	Requirement definition	Additional information	Status
	Big Data and Machine Learning Requirements	FR.D.01	The data sets available to the system shall be stored in a data lake, where they can be accessed as input for the data pipeline.	Achieved	Deliverable 4.1 - Section 2 "Data infrastructure"
		FR.D.02	The system shall contain a data processing pipeline that automates data cleaning and data preparation tasks.	Achieved	Deliverable 4.1 - Section 3 "Data preparation"
		FR.D.03	The system shall contain a data cleaning process, that automates the following tasks: • outlier detection • filtering / missing value handling for the data sets available in the data lake.	Achieved	Deliverable 4.1 - Section 3 "Data preparation" Deliverable 4.1 - Section 5 "Predictive modelling"
		FR.D.04	The system shall contain a data preparation process, that automates the following tasks: • data fusion • target labelling • feature engineering for the data sets available in the data lake, and generates training data sets, test data sets and validation data sets.	Achieved	Deliverable 4.1 - Section 3 "Data preparation" Deliverable 4.1 - Section 5 "Predictive modelling"
		FR.M.01	The system shall contain a machine learning model training process, that optimizes	Achieved	Deliverable 4.1 - Section 5 "Predictive modelling"

EUROPEAN PARTNERSHIP

Co-funded by the European Union

Requirement type	Requirement category	Requirement ID	Requirement definition	Additional information	Status
			the prediction of a machine learning model, given a training data set.		Deliverable 4.2 - Section 3 "Case studies predictive results"
Non-Functional Requirements	Input Data	NF.D.01	The data set provided as input to the system shall contain information on: • A/C performance • meteorological conditions • pilot inputs to the A/C WTC of the A/C	Achieved	Deliverable 4.1 - Section 3 "Data preparation" Deliverable 4.1 - Section 5 "Predictive modelling" Deliverable 4.2 - Section 3 "Case studies predictive results"
	Computational Efficiency	NF.C.01	The information about the probability of a GA prediction should be provided in real time (less than 0.5s after provision of input data)	Achieved	In both case studies, model prediction can be performed in less than 0.5 seconds. The main bottleneck is the processing of the data, although this can be achieved in under a second per flight.
	Model Training	NF.M.01	The performance assessment of the system shall include quantifiable metrics on: • true positive, true negative, false positive and false negative ratios	Achieved	Deliverable 4.1 - Section 5 "Predictive modelling" Deliverable 4.2 - Section 3 "Case studies predictive results"

Co-funded by the European Union

Requirement type	Requirement category	Requirement ID	Requirement definition	Additional information	Status
			accuracy, precision, recall and specificity		
		NF.M.02	The model training shall be able to cope with imbalanced training data sets	Achieved	Deliverable 4.1 - Section 5 "Predictive modelling" Deliverable 4.2 - Section 3 "Case studies predictive results"

EUROPEAN PARTNERSHIP

Co-funded by the European Union

6 References

[1] SafeOPS Consortium, „D2.1 User, functional and data requirements," 2021. [Online]. Available: https://ec.europa.eu/research/participants/documents/downloadPublic?documentIds=080166e5e0f1c687&appId=PPGMS. [Zugriff am 15 7 2022].

[2] SafeOPS Consortium, „D3.3 Human Factors Analysis of the Impact of Providing Probabilistic Risk Information in Real Time," 2021. [Online]. Available: https://ec.europa.eu/research/participants/documents/downloadPublic?documentIds=080166e5ecb7a660&appId=PPGMS. [Zugriff am 15 7 2022].

[3] SafeOPS Consortium, „D 4.1: Complete data pipeline description and ML solution," 2021. [Online]. Available: https://ec.europa.eu/research/participants/documents/downloadPublic?documentIds=080166e5ec5dac75&appId=PPGMS. [Zugriff am 15 7 2022].

[4] T. T. Tran, „Explainability vs. Interpretability and methods for models' improvement," Hamburg University of Applied Science, Hamburg, 2020.

[5] T. Miller, „Explanation in artificial intelligence: Insights from the social sciences," *Artificial Intelligence,* Bd. 267, p. 1–38, 2019.

[6] C. Molnar, Interpretable machine learning, Second edition Hrsg., Munich: Christoph Molnar, 2022, p. 318.

[7] M. T. Ribeiro, S. Singh und C. Guestrin, „"Why Should I Trust You?"," New York, NY, ACM, 2016, p. 1135–1144.

[8] L. S. Shapley, Notes on the N-Person Game - I, RAND Corporation, 1951.

[9] Artimation Consortium, „ARTIMATION," [Online]. Available: https://www.artimation.eu/. [Zugriff am 15 7 2022].

[10] TAPAS Consortium, „TAPAS," [Online]. Available: https://tapas-atm.eu/. [Zugriff am 15 7 2022].

[11] MAHALO Consortium, „MAHALO," [Online]. Available: http://mahaloproject.eu/. [Zugriff am 15 7 2022].

[12] ASIA Consortium, „AISA," [Online]. Available: https://aisa-project.eu/. [Zugriff am 15 7 2022].

[13] M. Robnik-Šikonja und M. Bohanec, „Perturbation-Based Explanations of Prediction Models," in *Human and machine learning*, Cham, Switzerland, Springer, 2018, p. 159–175.

[14] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot und Édouard Duchesnay, „Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research,* Bd. 12, Nr. 85, p. 2825–2830, 2011.

[15] R. Dolphin, „LSTM Networks | A Detailed Explanation," 2020. [Online]. Available: https://towardsdatascience.com/lstm-networks-a-detailed-explanation-8fae6aefc7f9. [Zugriff am 15 7 2022].

[16] S. M. Lundberg und S.-I. Lee, „A Unified Approach to Interpreting Model Predictions," Bd. 30, Curran Associates, Inc, 2017.

[17] S. M. Lundberg, G. Erion, H. Chen, A. DeGrave, J. M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal und S.-I. Lee, „From Local Explanations to Global Understanding with Explainable AI for Trees," *Nature machine intelligence,* Bd. 2, Nr. 1, p. 56–67, 2020.

[18] Mukund Sundararajan, Ankur Taly und Qiqi Yan, „Axiomatic Attribution for Deep Networks," Bd. 70, PMLR, 2017, p. 3319–3328.

7 List of abbreviations

ADS-B	Automatic Dependent Surveillance–Broadcast
AI	Artificial Intelligence
AP	Average Precision
ATCO	Air Traffic Control Officer
ATM	Air Traffic Management
AUC	Area Under the Curve
EDA	Exploratory Data Analysis
EDDF	Frankfurt Airport
EDDM	Munich Airport
HMI	Human-Machine Interface
ICE	Individual Conditional Expectation
LIME	Local Interpretable Model-agnostic Explanations
LSTM	Long Short-Term Memory
METAR	Meteorological Terminal Air Report
ML	Machine Learning
ML_CS	Machine Learning Case Study
NM	Nautical Miles
OL	Operational Layer
PDP	Partial Dependacy Plots
PL	Predictive Layer
PR	Precision-Recall
RF	Risk Framework
RNN	Recurrent Neural Networks
ROC	Receiver Operating Characteristic
SHAP	Shaley Additive Explanations
WP	Work Package
XAI	Explainable AI

EUROPEAN PARTNERSHIP

Co-funded by the European Union

-END OF DOCUMENT-

EUROPEAN PARTNERSHIP

Co-funded by the European Union