Fifteenth USA/Europe Air Traffic Management Research and Development Seminar (ATM2023)

Probabilistic Pre-tactical Arrival and Departure Flight Delay Prediction with Quantile Regression

A case study for Geneva international airport using operational data

Ramon Dalmau & Paolino De Falco
EGSD/INO/ENG
EUROCONTROL Innovation Hub (EIH)
Brétigny-Sur-Orge (France)

Miroslav Spak
NMD/TRG/TDD
EUROCONTROL
Brussels (Belgium)

José Daniel Rodriguez Varela
Geneva Airport
Geneva (Switzerland)

***Abstract*—Airports plan their resources well in advance based on anticipated traffic. Currently, the only traffic information accessible in the pre-tactical phase are the flight schedules and historical data. In practice, however, flights do not always depart or arrive on time for a variety of reasons, such as air traffic flow management or reactionary delay. Because neither air traffic flow management regulations nor aircraft rotations are known during the pre-tactical phase, predicting the precise arrival and departure delay of individual flights is challenging given current technologies. As a result, probabilistic flight delay predictions are more plausible. This paper presents a machine learning model trained on historical data that learned the various quantiles of the departure and arrival delay distributions of individual flights. The model makes use of input features available during the pre-tactical phase, such as the airline, aircraft type, or expected number of passengers, to provide predictions of the delay distribution several days before operations. The performance of the model trained on operational data from Geneva airport is compared to a statistical baseline, providing evidence that machine learning is superior. Furthermore, the contribution of the various input features is quantified using the Shapely method, stressing the importance of the expected number of passengers. Finally, some examples are presented to illustrate how such a model could be applied in the pre-tactical phase.**

***Keywords*—Flight delay; machine learning; quantile regression**

I. Introduction

Flight delay is commonly defined as the difference between actual and scheduled times of departure or arrival of a flight from or to an airport, respectively[1]. As a result of the crisis related to the COVID-19 pandemic, aviation activities drastically dropped in 2020 [1]. However, the last few months have recorded a recovery of air traffic, characterised by increasing flight delays. During the 3rd quarter of 2022, the average delay per flight in the European Civil Aviation Conference (ECAC) area was 23 min, the highest value recorded in the last 5 years [2]. Flight delay is one of the key performance indicator of air transportation since it can impact negatively the airline and airport management as well as the level of passengers' satisfaction [3]. Better prediction of flight delays could aid in the implementation of mitigation measures before they occur.

Machine learning algorithms have proven to be effective in predict flight delays during the tactical phase (i.e., during the day of operations) [4], [5], when both aircraft rotations and air traffic flow management (ATFM) measures are fully (or partially) available to assist the models. A critical aspect when developing machine learning models is the type and quality of data that are available at the time horizon of interest. From the Network Manager (NM) and airports point of view, the only data available during the pre-tactical phase (i.e., several days before operations) are the flight schedules, as aircraft rotations and exact ATFM measures are still unknown. In such an uncertain time horizon, it is more reasonable to approach the flight delay prediction problem with probabilistic models capable of providing not only the expected value of the delay (arrival or departure) but also its probability distribution.

From the airline perspective, the duty manager in charge of monitoring and manage the fleet decides whether a flight should be cancelled or revised [6]. In order to perform this task, he/she needs to access information about the costs of different alternatives, which are non-linear with respect to the (uncertain) delay values [7]. Capturing not only the expected values of the flight delay but also its likelihood might improve the decision-making process of the duty managers.

From the airport perspective, airport management implies decision making under uncertainty, which becomes critical especially for long look-ahead times [8]. As an example, strategic airport capacity planning is typically not sufficiently accurate because of the inherent uncertainty of weather forecasts [9]. Although flight schedules provide an indication on when aircraft might depart from or arrive at an airport, this information always carries a certain amount of uncertainty which makes airport planning operations very challenging (e.g., to decide where and when to allocate the ground handling resources, or to efficiently plan the shifts for the staff), especially several days before the day of operations.

This paper presents a probabilistic model that utilises historical flight data to predict arrival and departure flight delays several days in advance. The model is based on multi-

[1]Throughout this paper, departure time refers to off-block time, whereas arrival time refers to in-block time.

quantile regression, which is a method for estimating how the different quantiles of a distribution (in this context, the departure and arrival delay distribution) change as a function of a set of predictors. It should be noted that all predictors used in this paper (also known as features in the machine learning jargon) are available during the pre-tactical phase. The predictions in the test set, which includes observations never seen by the model during training, are compared to a dummy baseline that assumes flights will depart and arrive on time, as well as a baseline based on standard statistics.

This paper is organised as follows: a literature review on flight delay prediction, with a particular focus on probabilistic models, is performed in Section II; Section III provides the description of the generic multi-quantile regression model that was tailored to the departure and arrival flight delay prediction problem at Geneva Airport (GVA); the details of the experiment and the results are presented in Sections IV and V, respectively; Section VI provides a discussion of the results and an overview of the implementation at GVA.

II. Literature review

In recent years, along with the development of sophisticated machine learning models, there has been a lot of interest in probabilistic flight delay prediction. The emergence of probabilistic flight delay prediction models is also likely due to the fact that point predictions are not sufficiently accurate given the uncertainty of the air transportation system, in which many agents interact (passengers, ground handlers, air traffic controllers, flight dispatchers, etc.) in addition to the weather.

For example, [10] used random forest (i.e., an ensemble of decision trees trained with the *bagging* method) and clustering algorithms to predict departure delays at US airports with a look-ahead time up to 24 hours. Promising results revealed that combining clustering and ensembles of decision trees is effective at predicting flight delays several hours in advance.

The effectiveness of alternative machine learning models has also been investigated. For instance, [11] compared the performances of random forest and recurrent neural networks (RNNs) when predicting the flight delay at Chinese airports. The authors also approached the flight delay prediction problem as a classification task, in which the model learns the probability of the delay falling into one of several predefined categories rather than forecasting the precise delay in minutes.

RNNs were also used by [4] to predict the arrival delay propagation along a sequence of flights (i.e., rotation) operated by an aircraft along the day. Specifically, the model was trained to predict the parameters of the arrival delay distribution, which was modelled as a Gaussian function for the sake of simplicity. The proposed model requires rotations data to propagate the (predicted) delay, which, as previously stated, are not available during the pre-tactical phase.

In parallel, [5] also addressed the probabilistic flight delay prediction problem. The authors presented a machine learning model to categorise flight departure times as *early*, *on-time*, or *delayed*. Similar to [4], however, the proposed model requires knowledge on the prior flight operated by the same aircraft, and thus cannot be used during the pre-tactical phase.

Recently, [12] developed two probabilistic models for individual flight delay prediction model using mixture density networks (MDN) and random forest, respectively. In reality, however, the generic random forest model was designed to perform point predictions, not probabilistic. In order to obtain the flight delay distribution from a random forest, the predictions of the individual decision trees of the ensemble were not averaged, but collected, and a kernel density estimation (KDE) was performed. Using this approach, however, the model is still trained to minimise a loss function designed for point predictions, like the mean absolute error (MAE).

Regarding the MDN proposed by [12], it comprises a neural network that predicts the parameters for each Gaussian component in the mixture. The parameters (i.e., weights and biases) of the neural network are trained to minimise the negative log-likelihood. Consequently, the MDN assumes that the delay can be represented by a multi-modal Gaussian distribution. On the other hand, the model proposed in this paper does not make any specific assumptions regarding the shape of the delay distribution. Furthermore, it is important to note that neither the random forest nor the MDN proposed by [12] could used several days before operations since they rely on weather information at the destination or origin airport, which is only available 24 hours before operations.

In a similar vein, [13] explored probabilistic flight delay predictions using Bayesian artificial neural networks (ANNs) to predict aggregate flight delays in the United States, broken down by airport. Their study highlights the difficulty of predicting even aggregate-level flight delays, underscoring the importance of uncertainty quantification. Similar to [12], the model requires weather features (e.g., visibility, temperature) that are not available several days before operations.

Finally, [14] assessed the performance of various machine learning models for probabilistic flight delay prediction, including ANNs, random forest and gradient-boosted decision trees (GBDTs). Like [4], the authors assumed a Gaussian distribution that was fitted to the flight delays. The various machine learning models were then trained on historical data to predict the parameters of the distribution. For the GBDTs instance, two models were trained: one to predict the mean and the other to predict the standard deviation. Different from [4], [12] and [13], the features used by the models are available in the pre-tactical phase.

III. Generic model

Unlike classical regression models, which estimates the conditional mean of the target (i.e., the output) across features (i.e., the inputs) using the least squares approach, quantile regression determines the relationship between the features and a quantile (or quantiles) of the target distribution. It should be noted that, in contrast to previous works that parameterised the presumed delay distribution and then learned its parameters using machine learning [4], [14], quantile regression makes no assumptions about the distribution of the target and is robust to the influence of outliers.

There are various machine learning models that can be extended to quantile regression tasks. Gradient descent-based

learning algorithms, such as ANNs, can learn a specific quantile by switching from the classical MAE or mean squared error (MSE) loss to the mean pinball error (MPE):

$$\text{MPE} = \frac{1}{n_{\text{train}}} \sum_{i=1}^{n_{\text{train}}} \text{PE}\left(y_i, \hat{y}_i, \alpha\right), \tag{1}$$

where n_{train} is the number of training observations, y_i and $\hat{y}_i$ are the actual and predicted target for the i^{th} observation, respectively, $\alpha \in [0, 1]$ is the quantile to be learned, and

$$\text{PE}\left(y, \hat{y}, \alpha\right) = \alpha \max(y - \hat{y}, 0) + (1 - \alpha) \max(\hat{y} - y, 0) \tag{2}$$

is the pinball error (PE) for one observation.

In many practical applications, the goal is to determine not just one, but several quantiles. There are two methods for accomplishing this goal. The first approach involves training a separate model for each quantile. Because the models corresponding to the different quantiles are trained independently, the consistency of the predictions cannot be guaranteed [15]. Furthermore, this strategy necessitates the development and maintenance of many models, making it a time-consuming and inefficient option in practise.

The second approach consists of training just one model with a outputs, each one associated to one quantile, to minimise the mean multi-quantile pinball error (MMQPE):

$$\text{MMQPE} = \frac{1}{n_{\text{train}}} \sum_{i=1}^{n_{\text{train}}} \sum_{j=1}^{a} \text{PE}\left(y_i, \hat{y}_i, \alpha_j\right). \tag{3}$$

Many machine learning models can be configured to handle multi-quantile regression tasks. The generic model proposed in this study is based on ensemble methods, which produce a strong learner from a group of weak learners. Boosting is a well-known ensemble method that involves training a series of weak learners (e.g., rudimentary decision trees) sequentially. The training observations for the next learner in traditional adaptive boosting (AdaBoost) [16] are weighted based on how well the previous learners performed, i.e., observations that correspond to wrong predictions are assigned more weight in order to concentrate the model's attention on correcting them. Gradient boosting differs from AdaBoost in that, instead of assigning weights to observations based on performance, a new learner is trained at each iteration to fit the residual errors of the preceding learners. The ensemble is known as GBDTs model when decision trees are used as weak learners.

GBDTs can outperform ANNs in many practical applications, notably on tabular datasets where each row corresponds to one observation and each column represents a feature [17]. Furthermore, GBDTs are easier to interpret than ANNs and have very attractive properties such as the ability to handle missing data and categorical features with high cardinality. The GBDTs model was chosen for the problem addressed in this study because of the numerous benefits it provides.

Sections III-A and III-B list the features that compose the observation vector $\boldsymbol{x}$ and define the target y of the generic model developed during this research, respectively. It should be emphasised that this generic model could be trained using any of the traditional GBDTs algorithms (e.g., lightGBM, CatBoost, XGBoost) on historical data gathered by any airport. Section IV will present the specific GBDTs algorithm used to train the model as well as the dataset. Furthermore, two independent GBDTs models were trained: one to predict the quantiles of the arrival delay distribution and the other to predict the quantiles of the departure delay distribution. The set of features used by these models, however, is similar.

A. Input features

There are various limitations on the set of features that can be incorporated when building a model for usage during the pre-tactical phase. Predictions cannot, of course, be made using information from the future. For example, the majority of ATFM regulations are defined either the day before operations (so-called pre-tactical regulations) or tactically the same day. As a result, this information is unknown several days or weeks in advance. Similarly, the sequence of flights operated by each aircraft (i.e., registration number) must be known in order to anticipate rotational reactionary delays. The registration number that is going to operate a certain flight is only known when the airline submits the flight plan to the Network Manager (NM). Airlines, however, tend to wait for the most accurate weather and network information before submitting the flight plan. As a result, several days or weeks in advance, only the aircraft type that will be used for a flight can be speculated, but not which will be the inbound flight.

Based on the preceding discussion, it is understandable that the set of (more or less certain) features available for making predictions in the pre-tactical phase is rather limited. The model proposed in this paper uses the following 14 features: (1) airline, (2) handling agent who will process the flight, (3) destination (resp. origin) airport for departures (resp. arrivals), (4) aircraft type (e.g., A320), (5) flight service type, (6) type of flight (e.g., scheduled), (7) whether or not is a Schengen flight, (8) hour of the day, (9) day of the week, (10) month of the year, (11) great circle distance (GCD), as well as (12) the number of departures and (13) arrivals scheduled in the same hour. The last feature of the model is the expected number of passengers, which is estimated based on historical load factors[2] according to a model executed by the operations performance & forecasting department of GVA.

It is worth noting that the notion that circular features, such as the hour of the day, day of the week, or month of the year, always require transformation using sine and cosine functions is often misunderstood. While this transformation is commonly used in neural networks to capture periodicity, decision tree-based algorithms, such as random forest and GBDTs, can effectively handle circular features without the need for explicit transformation. The authors conducted experiments using both approaches and found that the categorical approach generally yields superior results.

Furthermore, it may seem that GCD and airport are highly correlated and provide duplicate information, but this is not the case. Different airports may have different operating

[2]The load factor is an aviation industry indicator that represents the proportion of available seating capacity that has been filled with passengers.

methods, leading to different contributions to the predictions. Meanwhile, the GCD feature was included to allow the model to learn the correlation between delay and the length or duration of the flight. Additionally, observations with airports that have few observations and are not representative in the training set could benefit from the more generic GCD feature.

B. Output target

The departure (resp. arrival) delay prediction model outputs multiple quantiles of the predicted departure (resp. arrival) delay distribution. Specifically, the models were trained to minimise the MMQPE and predict the 5^{th}, 25^{th}, 50^{th} (i.e., the median), 75^{th} and 95^{th} quantiles of their respective targets. The quantiles were selected to represent the entire delay distribution, including both regular and extreme events.

It should be noticed that during the training phase, the model generates five values (one for each quantile) for each training observation but only requires one ground truth y (the actual delay) to compute the multi-quantile pinball error.

IV. Experiment

This section describes the experiment carried out in this research to evaluate the performance of probabilistic models in predicting departure and arrival delays during the pre-tactical phase. Section IV-A describes the datasets used for the experiment, while Section IV-B covers the specific GBDTs algorithm as well as the hyper-parameters of the models.

A. Specific dataset

A dataset is a collection of n observations $\boldsymbol{X} := (\boldsymbol{x}, y)^n$ used to train a model and assess its performance. In this work, two datasets were created: one for departures and one for arrivals, with each observation belonging to one flight departing from or arriving to GVA from the 28^{th} of October 2018 to the 11^{th} of December 2022, respectively. It should be noted that the traffic from March, 1^{st} 2020 to July, 1^{st} 2021 was excluded from the dataset because it was strongly affected by the COVID-19 pandemic. The raw data used to generate the features and target of each observation were kindly provided by GVA. A portion of the data, such as the predicted number of passengers per flight, is confidential and therefore cannot be publicly disclosed.

Table I lists and describes the features that compose the observation vector $\boldsymbol{x}$ in the two datasets. The columns of this table show basic statistics computed on the entire datasets, including both train and test sets. For each categorical (i.e., discrete) feature, like the departure airport, Table I shows the number of unique values, the most frequent value (Top) as well as its frequency (Freq.). For each numerical (i.e., continuous) feature, like the great circle distance (GCD), three quartiles are presented: 25^{th} (Q1), median (Q2) and 75^{th} (Q3).

It is worth noting that the model presented in this paper does not incorporate weather features since it is designed to be used several days before operations, when weather forecasts are often inaccurate. However, a variant of this model could be designed for use one day before operations, when Terminal Area Forecasts (TAF) for the next 24 hours are available. This variant could incorporate features such as visibility, cloud ceiling, wind speed and direction, gusts, and significant weather phenomena like thunderstorms, fog, and snow. With these additional features, this variant could better capture the effect of weather on departure and arrival delays.

Furthermore, The model does not incorporate features related to reactionary delay because the sequence of flights operated by each aircraft is only known on the day of operations when airlines submit their flight plans to the NM. Nonetheless, the authors of this paper encourage the air traffic management (ATM) research community to explore the possibility of developing a model that can predict the rotations of a particular aircraft days in advance based on its recent history. The predicted rotations could be incorporated as additional information into the model presented herein, likely improving the quality of the model's predictions.

In many applications, dataset splitting is done randomly by taking 80% of the data for training and using the rest for assessing the performance on *unseen* data (i.e., testing). When dealing with time-related and dynamically changing environments, such as the air transportation system, it is preferable to employ time-based splitting to provide statistically robust model evaluation and better imitate real-life scenarios. Accordingly, the first 80% of the flights (ordered by time) where used for training, and the rest for testing.

B. Specific algorithm

In this paper, the CatBoost implementation of the GBDTs model by Yandex [18] was used. CatBoost has gained more momentum than other GBDTs implementations (e.g., XGBoost and LightGBM) mainly because its native ability to handle high-cardinality categorical features like the departure and destination airports, as well as the use of ordered boosting and symmetric trees, which help to overcome over-fitting.

Many hyper-parameters can be used to optimise the CatBoost model, which allow to control the entire ensemble (e.g., the number of decision trees) as well as individual decision trees (e.g., the maximum depth). In the experiment conducted in this study, only the maximum depth and the number of decision trees were optimised because they were found to have the most significant impact on the loss function. The learning rate was determined automatically using the CatBoost framework's heuristic, which is dependent on the dataset attributes and the number of decision trees.

A widely used procedure to assess the performance of a model given a combination of hyper-parameters is the cross-validation (CV). The most basic k-fold CV, for instance, consists of splitting the train set into k subsets, also known as folds. Then, the following procedure is applied to each of the k folds: a copy of the model is trained using the other $k-1$ folds as train set, while the fold in hand is used as test set to compute a performance score. The average of the k scores is the CV score of the model using the combination of hyper-parameters under consideration. In this paper, the CV procedure was performed by respecting the temporal order of the observations with a `TimeSeriesSplit` [19]. Specifically, this variation returns first (order by time) i folds as train set and the $(i+1)^{\text{th}}$ fold as validation set (with $i \in \{1, k-1\}$), and averages the resulting $k-1$ scores.

TABLE I. Input features and statistics on the entire dataset (train & test)

Type	Name	Departures (170K observations)			Arrivals (170K observations)		
		Unique	Top	Freq (%)	Unique	Top	Freq (%)
Categorical	Airline	159	EZS	23	153	EZS	23
	handling agent	2	SWISSPORT	74	2	SWISSPORT	74
	Airport	266	LHR	6	267	LHR	6
	Arctyp	68	A320	42	68	A320	42
	Flight service type	14	J	97	14	J	97
	Type of flight	7	S	55	5	S	54
	Schengen flight	2	Y	66	2	Y	66
	Dayofweek	7	6	15	7	4	15
	Hour	19	9	8	20	8	7
	Month	12	12	10	12	12	10
		Q1	Q2	Q3	Q1	Q2	Q3
Numerical	Pax total (#)	82	123	155	83	123	156
	Hourly arrivals (#)	6	10	13	8	11	15
	Hourly departures (#)	9	12	15	7	10	14
	Great Circle Distance, GCD (km)	532	754	1309	532	754	1309

There exist several methods to search the hyper-parameter space for the best CV score. The most popular method is the `GridSearchCV` [19], which consists of exhaustively evaluating all the possible candidates (i.e., combinations of hyper-parameter) and returning that minimising the CV score. In this study, a more refined method called `HalvingGridSearchCV` [19] was used. The `HalvingGridSearchCV` consists of evaluating all possible candidates with a small amount of *resources* at the first iteration. In the second iteration, only some of these candidates are selected for the next iteration, which will be allocated more resources, and so on. In this paper, the number of decision trees in the ensemble was used as resource. The reader should keep in mind that when using the number of decision trees (also known as estimators) as resource, this hyper-parameter cannot be included in the search grid. It is optimised intrinsically by the `HalvingGridSearchCV` method, and including it in the search grid (which would be incorrect) will result in an exception being thrown.

For both departure and arrival delay prediction models, the best maximum depth and number of decision trees were found to be 9 and 1K, respectively.

V. Results

This section presents the results of the experiment described in Section IV. Specifically, Section V-A compares the performance metrics of the model with those of two baselines. Then, Section V-B unravels the attribution of the features according to the Shapley values computed with the trained models. Finally, Section V-C shows illustrative examples.

A. Performance

The performance of the proposed models must be compared to some reference values, i.e., the baseline. The simplest baseline is to assume that all flights will depart and arrive on time at their scheduled departure and arrival time, respectively. That is, regardless of the values in the observation vector $\boldsymbol{x}$, the predicted quantiles are zero for all observations in the test set. For the remainder of the paper, this baseline will be referred to as the 'zero delay'. This baseline is practical because it is extremely simple, and it also nearly replicates the current system when delays are completely disregarded.

A more principled baseline consists of predicting the quantiles of the delay distribution based on historical data. In this paper, flights in the train set were grouped by season (winter, spring, summer or autumn), period of the day (morning, afternoon, evening, late or late night) as well as airport (departure or destination depending on the model). For each one of these groups, the various quantiles of the departure and arrival delay distributions were computed, and these quantiles were used as predictions for the observations in the test set belonging to the same group. It is worth mentioning that groups with fewer than 250 observations were declared underrepresented, implying that the predicted quantiles are not statistically significant. Observations in the test set that were missing predictions because their group was underrepresented in the train set were assigned the quantiles of the delay distribution resulting from grouping by season and period of year, omitting the airport. For the remainder of the paper, this baseline will be referred to as the 'statistics'.

The authors selected these two rather naive baselines because, to the best of their knowledge, none of the models proposed in the literature, except for [14] - who also used a GBDTs model, but assumed a Gaussian distribution - are able to perform probabilistic predictions several days before operations due to the need for weather data, information about aircraft rotations, and/or ATFM regulations.

Table II shows the performance metrics on the test set for the two baselines and the machine learning models. Results indicate that the statistics baseline outperforms the zero delay baseline, particularly for the high quantiles. It reduces the 95^{th} quantile's MPE of the departure and arrival delay distributions by 11.1 min (64%) and 9.6 min (63%), respectively. However, the performance is very similar at the low quantiles. The statistics baseline reduces the MMQPE of the zero delay baseline by 15.9 min (33%) and 15 min (31%) for the departure and arrival delay predictions, respectively.

TABLE II. Performance metrics on the test set (min)

Operation	Model	MPE for the various quantiles					MMQPE
		5th	25th	Median	75th	95th	
Departures	Zero delay	1.9	5.3	9.6	13.9	17.4	48.1
	Statistics	1.3	5.3	8.9	10.3	6.3	32.2
	CatBoost	1.2	4.9	7.9	8.7	5.2	28.0
Arrivals	Zero delay	4.2	6.6	9.7	12.8	15.3	48.6
	Statistics	1.8	6.3	9.7	10.2	5.7	33.6
	CatBoost	1.6	5.8	8.8	9.2	5.0	30.3

The machine learning models proposed herein perform even better. It should be noted, however, that the performance gap between the statistics and zero delay baselines is higher than the performance gap between the CatBoost regressor and the statistics baseline, indicating that relatively simple statistical methods can indeed deliver decent predictions.

Specifically, the CatBoost model reduces the MMQPE of the zero delay baseline by 20.1 min (42%) and 18 min (38%) for the departure and arrival delay prediction tasks, respectively. The relative benefit in comparison to the statistical baseline, however, is not that extraordinary: the MMQPE for the departure delay prediction task improves by 4.2 min (13%), whereas the improvement is about 3.3 min (11%) for the arrival delay prediction task. In any instance, the machine learning approach yields more reasonable estimates for all individual quantiles, particularly for the high quantiles.

The reader should be aware that the interpretation of the MMQPE may be misleading, since it is simply the sum over all the individual quantile's MPE. Adding more quantiles would increase each of the MMQPE values accordingly, (potentially) leading to more dramatic differences across models.

B. Feature attribution

Principles from game theory can be used to interpret the prediction of a model for a given observation vector $\boldsymbol{x}$, assuming that each one of the d features is a player and the model output $\hat{y}$ is the payout. Let us consider the following scenario: all features participate in the game (i.e., contribute to the model output), and the features enter the room where the game is played in a random order. The contribution of a feature could be calculated as the average change in the payout received by the coalition already in the room when the corresponding player (feature) joins them. This contribution measure is commonly known in the literature as the Shapley value. Specifically, the Shapley value $\phi_i(\boldsymbol{x})$ of the feature i for a given observation vector $\boldsymbol{x}$ represents the average marginal contribution of i on the output of the model across all possible combinations of features. It can be proven that the Shapley value is the only contribution measure that simultaneously satisfy local accuracy, consistency, and missingness [17].

In practical applications, however, Shapley values can only be approximated because computing them precisely is an NP-hard problem. `TreeExplainer` is a novel explanation method for tree-based models (including GBDTs) that allows for the tractable computation of Shapley values in polynomial time [17]. The `TreeExplainer` was used in this paper.

Figures 1 and 2 aggregate Shapley values for all the features and observations in the test sets, which were computed by using the `TreeExplainer` with the trained models. Because each model produces five outputs (the various quantiles of the predicted delay distribution), Shapley values can be computed independently for each quantile. Only the 5th, median and 95th quantiles are examined for the sake of clarity.

In Figs. 1 and 2, the vertical axis indicates the name of the features, in order of importance from the top to the bottom in terms of mean absolute Shapley value. Each dot in the horizontal axis shows the Shapley value of the associated feature on the prediction for one observation, and the colour indicates the magnitude of that feature: red indicates high, while blue indicates low. Note that colour has no meaning for categorical features such as the airline or the aircraft type.

According to Fig. 1a, the most influencing feature when predicting the 5th quantile of the departure delay distribution, in terms of mean absolute Shapley value, is the (expected) number of passengers. Results indicate that the higher the number of passengers, the greater the output of the model. It is important to remember that the expected value (i.e., mean) of the target in the train set plus the Shapley values of the individual features equals the model's output. As a result, a positive Shapley value indicates that the corresponding feature is influencing the model's output to be greater than the expected value in the train set. The month of the year and the aircraft type are closely followed by the airline, hour of the day and airport in the list of the most relevant features.

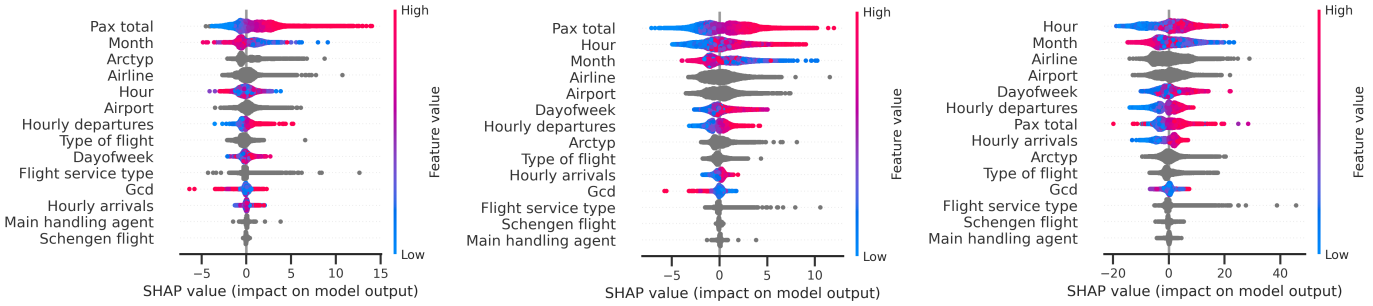
Figure 1b shows how the feature ranking changes when predicting the median. The findings indicate that calendar features are extremely important for higher quantiles.

The preceding statement is further supported by Fig. 1c, which shows that the most important features when predicting extreme departure delays are the hour, the month and the airline. Figure 1 reveals that there is no dominant feature (also known as *golden* feature) in the model, and that multiple features contribute to the output with about the same impact.

Another conclusion that can be derived from Fig. 1 is that, as expected, the higher the number of hourly departure operations (a proxy for airport congestion), the greater the predicted departure delay, albeit with a relatively minor contribution.

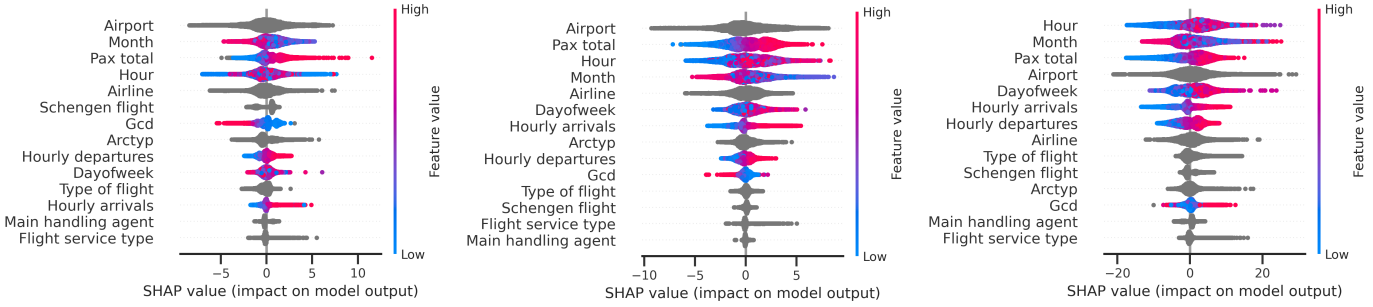
Curiously, Fig. 2 shows that the most important features of the arrival delay prediction model are ranked differently. Specifically, Fig. 1a indicates that the departure airport plays the most important role when predicting the 5th quantile of the arrival delay distribution, while the number of passengers is placed 3rd in terms of mean absolute Shapley value. Similar to Fig. 1, Fig. 2 indicates that calendar features are increasingly crucial at higher quantiles, and that the higher the number of arrival operations, the greater the predicted arrival delay.

Last but not least, Figs. 1 and 2 show that the absolute value of the Shapley value increases with the GCD. This indicates that, for long-haul flights, the delay strongly depends on the distance - according to the data used for training the models. The results presented in this section are only applicable to GVA, and other airports or countries may exhibit different figures. For example, in the United States, delays on connected short-haul flights are often more uncertain as the domino effect can grow from one leg to the next.



(a) 5th quantile (b) Median (c) 95th quantile

Figure 1: Feature attribution distribution for departure delay prediction model. It should be noted that the x-axes of the figures are represented with different scales to enable a visual assessment of the trends



(a) 5th quantile (b) Median (c) 95th quantile

Figure 2: Feature attribution distribution for arrival delay prediction model. It should be noted that the x-axes of the figures are represented with different scales to enable a visual assessment of the trends

The reader should keep in mind that each point in Figs. 1 and 2 corresponds to a single observation, i.e., flight. Accordingly, these figures represent the global interpretation of the departure and arrival delay prediction models, respectively, based on local feature attributions. Figures 3 and 4, on the other hand, illustrate the Shapley value of the expected number of passengers feature as a function of its value for the departure and arrival delay prediction models, respectively. As in the previous figures, each point represents one observation.

Figure 3a shows that, for the 5th quantile of the departure delay distribution, the relationship between the number of passenger and its Shapley value is linear. The results also show that when the number of passengers is below (roughly) 100, the Shapley value corresponding to this feature is negative (i.e., this feature contributes to predicting early departure when compared to the expected value in the training set), whereas it is positive when the number of passengers is above. A similar pattern can be observed for the median in Fig. 3b. For the 95th quantile (see Fig. 3c), however, the relationship is linear only when the number of passengers is below 200. The attribution of this feature is similar for the arrival delay prediction model, according to Fig. 4.

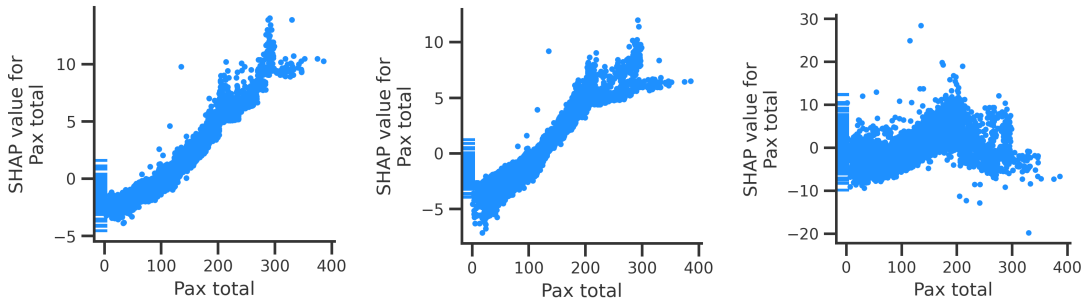
Figures 5 and 6 show the Shapley value of the hourly departures and arrival features as a function of their value for the departure and arrival delay prediction models, respectively. As expected, Figs. 5 and 6 indicate that the higher the number of hourly departures and arrivals, the higher the quantiles of the predicted departure and delay distribution, respectively. For instance, Figs. 5b shows that when the number of hourly departures is lower than around 10, the contribution of this feature is null or negative, whereas higher values tend to increase the median of the predicted delay distribution. Similar conclusions can be derived for the arrival delay prediction as well as for the rest of quantiles.

The dispersion of Shapley values for a specific value of a feature in Figs. 3 to 6 is due to the fact that the Shapley value depends on the value of the other features.

C. Illustrative applications

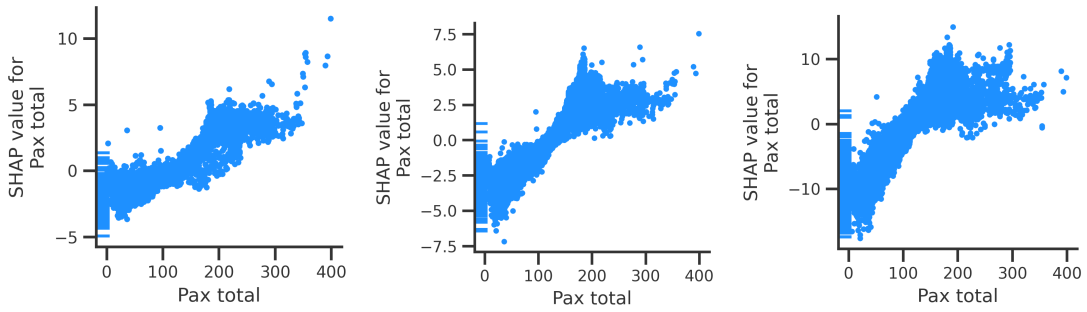
This section presents some illustrative examples of how the departure and arrival delay prediction models covered in previous sections could be used in real operations. In hierarchical order, Sections V-C1 to V-C3 show how to pinpoint flights that are likely to not depart or arrive on time starting for an aggregated prediction over the next months.

1) Detection of problematic days: Let us start with the most basic use case, in which airport operators plan their resources (like number of staff and handling agents, stand and gate allocation, etc.) several days in advance. Figure 7 (resp. 8) shows the mean absolute hourly mismatch between scheduled and potential number of departures (resp. arrivals). For instance, a date marked with the number 3 means that, in average during that day (considering the 24 hours), the absolute difference (positive or negative) between the number of scheduled and potential hourly operations is 3.



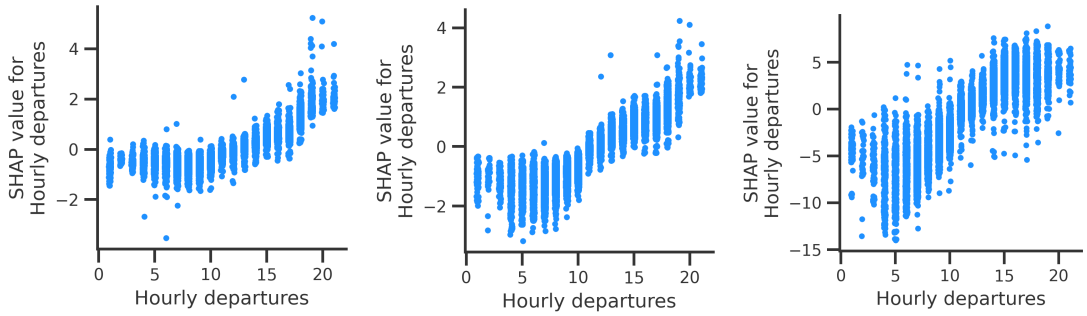
(a) 5th quantile (b) Median (c) 95th quantile

Figure 3: Attribution of the total pax. feature for departure delay prediction model as a function of its value. It should be noted that the y-axes of the figures are represented with different scales to enable a visual assessment of the trends



(a) 5th quantile (b) Median (c) 95th quantile

Figure 4: Attribution of the total pax. feature for arrival delay prediction model as a function of its value. It should be noted that the y-axes of the figures are represented with different scales to enable a visual assessment of the trends



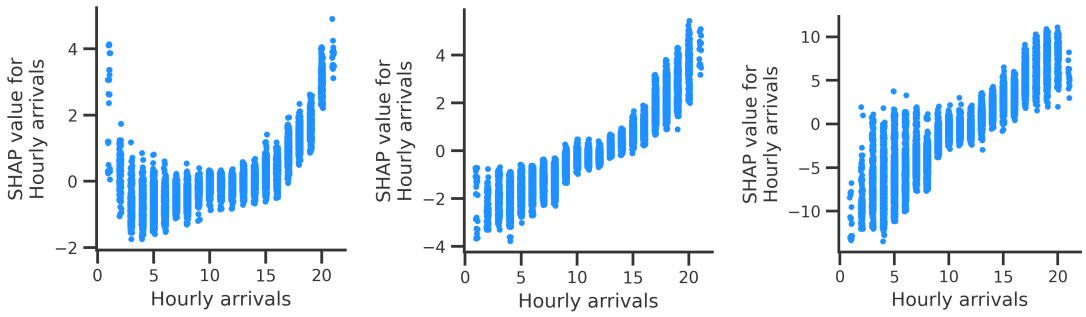
(a) 5th quantile (b) Median (c) 95th quantile

Figure 5: Attribution of the hourly departures feature for departure delay prediction model as a function of its value. It should be noted that the y-axes of the figures are represented with different scales to enable a visual assessment of the trends

Each cell within Figs. 7 and 8 shows the mismatch considering that flights could arrive at any time within a predicted quantile range. For example, a flight with a scheduled departure time of 10:30 and which 5th and 95th quantiles of the predicted departure delay distribution are -45 and 60 min, respectively, may depart at any time between 9:45 and 11:30, and thus should be considered when computing counts for the windows [9:00,10:00), [10:00,11:00), and [11:00,12:00). Accordingly, depending on the quantile range under consideration, a single flight could be counted in several windows.

Figures 7a and 8a show, respectively, the mismatch when flights depart and arrive as late (or early) as the median of the predicted delay distribution. It should be noted that because the median is a single value rather than a range, flights are counted only in one window. In this situation, Figs. 7a and 8a show that the mean absolute hourly mismatch never exceeds two operations (either departures or arrivals), and, as expected, dates across the whole summer season and on weekends are the most uncertain, particularly for the arrivals.

When considering that flights could depart or arrive at any time between the 25th and 75th quantiles of the predicted delay distribution, the discrepancy between the scheduled number of hourly operations and the potential number of operations begins to increase (see Figs. 7b and 8b). Obviously, the most

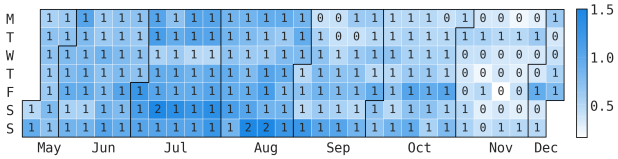


(a) 5th quantile (b) Median (c) 95th quantile

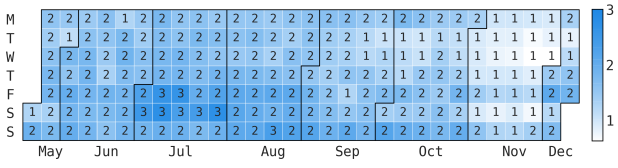
Figure 6: Attribution of the hourly arrivals feature for arrival delay prediction model as a function of its value. It should be noted that the y-axes of the figures are represented with different scales to enable a visual assessment of the trends

extreme difference is observed when considering that flights could depart or arrive at any time between the 5th and 95th quantiles of the predicted delay distribution. In that situation, which is shown in Figs. 7c and 8c, the mean absolute hourly mismatch could be as high as 8 departures and 9 arrivals, respectively.

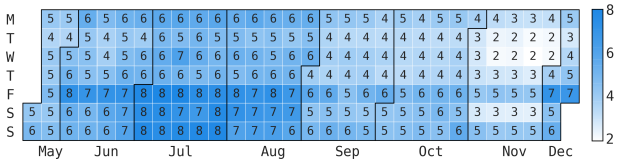
Airport operators could use this simple calendar view to identify dates with potential pitfalls caused by a difference between the number of scheduled and potential operations per hour. It is worth noting that because all of the model's features are accessible during the pre-tactical phase, airports might do this assessment months in advance. Once the most critical days have been found, operators could zoom in and identify the hours with the most (predicted) disparities.



(a) Median



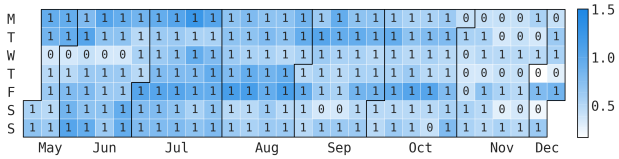
(b) Any time between the 25th and 75th quantiles



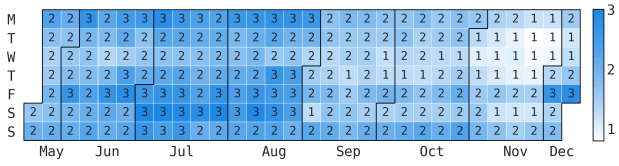
(c) Any time between the 5th and 95th quantiles

Figure 7: Mean absolute hourly mismatch between scheduled and potential number of departures in the test set

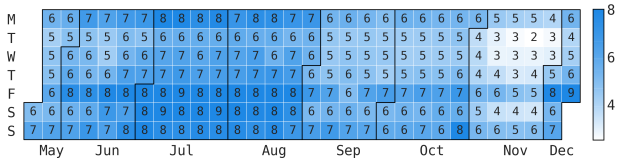
2) Detection of problematic hours: Based on Figs. 7c and 8c, the 30th of July has been selected to illustrate how problematic hours could be identified. This day showed the highest mean absolute hourly mismatch between scheduled



(a) Median



(b) Any time between the 25th and 75th quantiles

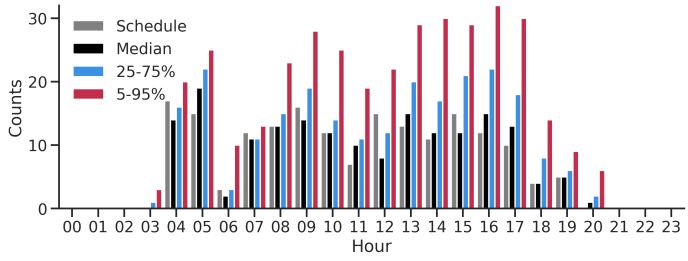


(c) Any time between the 5th and 95th quantiles

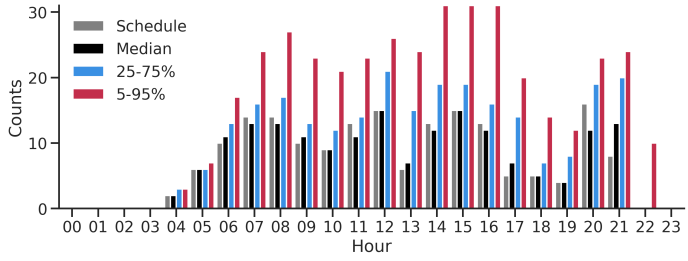
Figure 8: Mean absolute hourly mismatch between scheduled and potential number of arrivals in the test set

and potential number of operations. Figures 9a and 9b show the detailed hourly departures (resp. arrivals), considering that flights depart (resp. arrive) at the scheduled departure (resp. arrival) time, that realise the median delay, and that depart (resp. arrive) at any time between the 25th and 75th quantiles as well as between the 5th and 95th quantiles.

In Figure 9, the extension of the bar showing the number of probable events included by the 5th and 95th quantile values may be of particular interest (i.e., red bar) when conservative decisions might be taken by the user. Even more interesting could be to detect the most critical periods of a day by looking at the difference between the planned and predicted amount of operations (i.e., difference between the extension of the grey and red bars). However, the user might decide to plan an action without considering any uncertainty. In this case, the count provided by the median predictions (i.e., the black bar) should be adopted. An intermediate approach could be to consider a more likely range of predictions that are provided

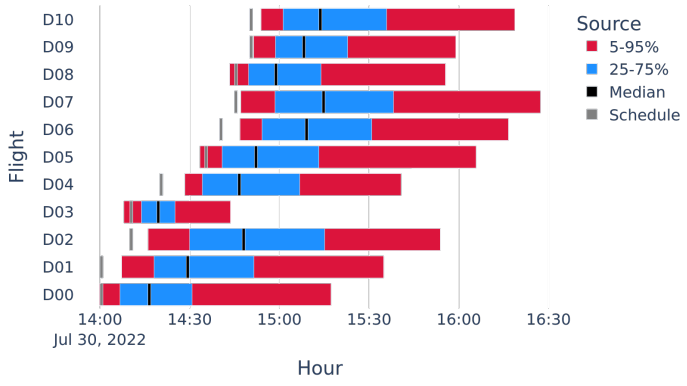


(a) Departures

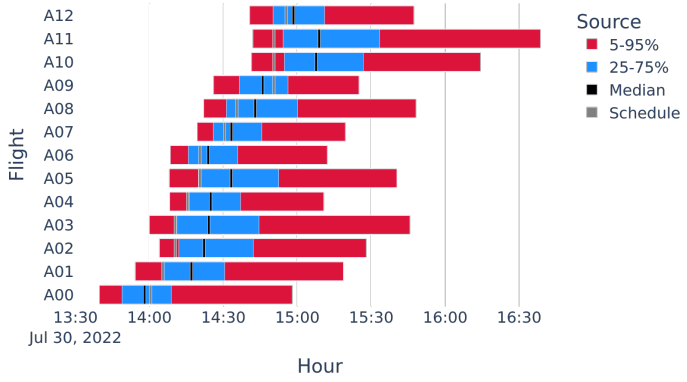


(b) Arrivals

Figure 9: Potential number of departure and arrival operations at GVA during the 30th of July 2022



(a) Departures



(b) Arrivals

Figure 10: Predicted quantiles of the flights scheduled to depart or arrive at GVA during the 30th of July 2022 from 14:00 to 15:00

by the 25-75th quantile values (i.e., blue bar).

3) Detection of problematic flights: Once a critical period of the day has been identified, it might be desired to detect which flights require more attention since their arrivals or departures predictions incorporate more uncertainty and present higher mismatched with the scheduled in-block and off-block times. As an example in Figure 10, on a day characterised by high delays such as the 30th of July 2022, flights D10, D09, D07, D06, D04, D02 are very likely to depart later than scheduled since their planned time does not even fall within the red bar (5-95th quantile) which covers 90% of possible occurrences (Figure 10a). On the contrary, it is possible to observe that the mismatch between planned and predicted arrival times is lower since the grey lines fall within or are in close proximity to the blue bars (Figure 10b) for most of the represented flights. An intuitive indication of criticality is the distance between the schedule time (grey lines) and both, the ends of the red bar (5-95th quantile) and the median value (black lines). An undesired scenario is represented indeed by flights scheduled much earlier than these two values. With these ad hoc predictions the assignment of airport resources for each single flight could be more efficient.

VI. Discussion & Conclusions

Once a machine learning model is trained, specific metrics, such as the average error between predicted and actual realisations of the target variable, can be computed accounting for both aleatory and epistemic uncertainty [20]. However, while these averaged statistics of the error can be used to assess the overall quality of the model, they do not provide a quantification of the uncertainty of a single prediction.

There are various models and methodologies for predicting flight delays in the literature. However, some of them provide punctual predictions, leaving to the user the assessment of possible deviations from the predicted values that might derive from the complex and uncertain environment in which flights are operated. Other approaches have been suggested in the literature to estimate the uncertainty of the individual predictions, such as sensitivity analyses [21], bootstrapping methods [22], Bayesian methods [23] and Gaussian processes [12]. Most of these methods provide an estimation of the variance of the error but they are not able to provide a range of probabilistic occurrences of predicted values.

In this paper, the uncertainty of individual predictions can be quantified by the quantile values provided by the model's outcomes. As an example, the difference between two (predicted) quantiles, one relatively high and another relatively low (e.g., 95th and 5th, respectively), represents the extension of the time domain over which departures and arrivals are predicted to take place. This quantification allows to assess the *criticality* (or miss-match between plan and prediction) of specific periods of the year at an aggregated level (see Figs. 8 and 7) and, more in detail, of specific days of the year (see Fig. 9) and of individual flights (see Fig. 10)

An important methodology to quantify the contributions of each single input feature to the predictions is the Shapley analysis, which results are shown in Section V. As a main

outcome of this analysis, it has been observed that the number of passenger highly affects the predictions. Specifically, the higher the number of passengers (*Pax total* in Figs. 1 and 2'), the higher the arrival and departure delays, showing that particular attention should be paid to operations involving passengers, such as boarding and de-boarding. For a flight showing high positive Shapley values of the total passenger input feature, further analysis and sets of data might allow to identify the operations involving passengers that are more likely to cause delays.

This study has been developed in response to a proposal of the operations performance & forecasting department of Geneva airport (GVA) within one of the EUROCONTROL Air Transport Innovation Network (EATIN) initiative (https://www.eurocontrol.int/project/eatin). As such, it is of particular interest to understand how this probabilistic approach can satisfy the needs of already complex and demanding airport operations. The model is currently under trial at GVA, and in the following months a survey will be conducted to study the impact that the model is making on the planning of the operations at GVA. As a result of the survey, a suitable human-machine interface might be developed and implemented. Alternatively, the schedule arrival and departure values could be replaced in the systems (planning, demand & capacity balance in the land side as well as in the air side, etc.) by the model values. This approach could be extended to other airports and adjusted to serve the needs of any other ATM stakeholder.

In future work, the performance improvement of incorporating weather features into the model, such as visibility, cloud ceiling, or wind speed, could be assessed. However, this variant would only be usable when weather forecasts for the airport are available, which is typically 24 hours before operations. A similar discussion applies to aircraft rotations and ATFM regulations.

REFERENCES

[1] A. Abu-Rayash and I. Dincer, "Analysis of mobility trends during the covid-19 coronavirus pandemic: Exploring the impacts on global aviation and travel in selected cities," *Energy research & social science*, vol. 68, p. 101693, 2020.

[2] C. Walker, "All-causes delays to air transport in europe - quarter 3 2022," *CODA DIGEST Q3 2022*, pp. 1–15, 2022.

[3] A. Sternberg, J. Soares, D. Carvalho, and E. Ogasawara, "A review on flight delay prediction," *arXiv preprint arXiv:1703.06118*, 2017.

[4] R. Dalmau, G. Murgese, Y. De Wandeler, R. Correia, and A. Marsden, "Early Detection of Night Curfew Infringements by Delay Propagation with Neural Networks," in *14th USA Europe Air Traffic Management Research and Development Seminar*, (Virtual Event), 2021.

[5] D. Sahadevan, P. Ponnusamy, M. K. Nelli, and V. P. Gopi, "Predictability improvement of scheduled flights departure time variation using supervised machine learning," *International Journal of Aviation, Aeronautics, and Aerospace*, vol. 8, 2021.

[6] H.-W. M. Vos, B. F. Santos, and T. Omondi, "Aircraft schedule recovery problem–a dynamic modeling framework for daily operations," *Transportation Research Procedia*, vol. 10, pp. 931–940, 2015.

[7] A. J. Cook and G. Tanner, "European airline delay cost reference values," 2011.

[8] A. Spencer, P. J. Smith, and C. Billings, "Airport resource management and decision aids for airlines," in *2005 International Symposium on Aviation Psychology*, p. 700, 2005.

[9] R. Kicinger, J.-T. Chen, M. Steiner, and J. Pinto, "Airport capacity prediction with explicit consideration of weather forecast uncertainty," *Journal of Air Transportation*, vol. 24, no. 1, pp. 18–28, 2016.

[10] J. J. Rebollo and H. Balakrishnan, "Characterization and prediction of air traffic delays," *Transportation research part C: Emerging technologies*, vol. 44, pp. 231–241, 2014.

[11] G. Gui, F. Liu, J. Sun, J. Yang, Z. Zhou, and D. Zhao, "Flight delay prediction based on aviation big data and machine learning," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 1, pp. 140–150, 2019.

[12] M. Zoutendijk and M. Mitici, "Probabilistic flight delay predictions using machine learning and applications to the flight-to-gate assignment problem," *Aerospace*, vol. 8, no. 6, 2021.

[13] T. Vandal, M. Livingston, C. Piho, and S. Zimmerman, "Prediction and uncertainty quantification of daily airport flight delays," in *Proceedings of The 4th International Conference on Predictive Applications and APIs*, vol. 82 of *Proceedings of Machine Learning Research*, pp. 45–51, October 2017.

[14] Z. Wang, C. Liao, X. Hang, L. Li, D. Delahaye, and M. Hansen, "Distribution prediction of strategic flight delays via machine learning methods," *Sustainability*, vol. 14, no. 22, 2022.

[15] S. J. Moon, J.-J. Jeon, J. S. H. Lee, and Y. Kim, "Learning multiple quantiles with neural networks," *Journal of Computational and Graphical Statistics*, vol. 0, no. 0, pp. 1–11, 2021.

[16] R. E. Schapire, "Explaining adaboost," in *Empirical inference*, pp. 37–52, Springer, 2013.

[17] S. M. Lundberg, G. Erion, H. Chen, A. DeGrave, J. M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal, and S.-I. Lee, "From local explanations to global understanding with explainable ai for trees," *Nature machine intelligence*, vol. 2, pp. 56–67, 2020.

[18] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "Catboost: unbiased boosting with categorical features," in *Proceedings of the 31st Advances in Neural Information Processing Systems (NIPS)*, (Montréal, Canada), 2018.

[19] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[20] Y. Li, J. Chen, and L. Feng, "Dealing with uncertainty: A survey of theories and practices," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 25, pp. 2463–2482, 11 2013.

[21] Z. Bosnić and I. Kononenko, "Estimation of individual prediction reliability using the local sensitivity analysis," *Applied Intelligence*, vol. 29, pp. 187–203, 2008.

[22] T. Heskes, "Practical confidence and prediction intervals," in *Proceedings of the 9th International Conference on Neural Information Processing Systems*, NIPS'96, (Cambridge, MA, USA), p. 176–182, The MIT Press, 1996.

[23] A. Khosravi, S. Nahavandi, D. Creighton, and A. F. Atiya, "Comprehensive review of neural network-based prediction intervals and new advances," *IEEE Transactions on Neural Networks*, vol. 22, no. 9, pp. 1341–1356, 2011.