Dispatcher3 – Machine learning for efficient flight planning

Approach and challenges for data-driven prototypes in air transport

Luis Delgado
University of Westminster
l.delgado@westminster.ac.uk

Sergi Mas-Pujol
Georgy Skorobogatov
Universitat Politecnica de Catalunya
sergi.mas-pujol@upc.edu
georgy.skorobotagov@upc.edu

Clara Argerich
Ernesto Gregori
Innaxis
ca@innaxis.aero
eg@innaxis.aero

***Abstract*—Machine learning techniques to support decision making processes are in trend. These are particularly relevant in the context of flight management where large datasets of planned and realised operations are available. Current operations experience discrepancies between planned and executed flight plan, these might be due to external factors (e.g. weather, congestion) and might lead to sub-optimal decisions (e.g. recovering delay (burning extra fuel) when no holding is expected at arrival and therefore it was no needed). Dispatcher3 produces a set of machine learning models to support flight crew pre-departure, with estimations on expected holding at arrival, runway in use and fuel usage, and the airline's duty manager on pre-tactical actions, with models trained with a larger look ahead time for ATFM and reactionary delay estimations. This paper describes the prototype architecture and approach of Dispatcher3 with particular focus on the challenges faced by this type of data-driven machine learning models in the field of air transport ranging: from technical aspects such as data leakage to operational requirements such as the consideration and estimation of uncertainty. These considerations should be relevant for projects which try to use machine learning in the field of aviation in general.**

Keywords - machine learning; challenges; pre-departure

I. Introduction

To conduct more efficient and sustainable operations an early anticipation of discrepancies between the planned and executed flights is paramount. Otherwise, those responsible for the flight operations – duty managers, dispatchers and pilots - might make sub-optimal decisions which could lead to higher costs and more emissions. The anticipation of disruptions, e.g. ATFM delay issued to flights, are required to support the creation of flight management solutions which could minimise the environmental impact of aviation, e.g. avoiding expensive tactical recovery of delay by increasing flight speed, when actions modifying the planning of rotations could have led to a lower usage of fuel if disruption were anticipated with enough look ahead time.

Flight operations generate a large set of data from different sources: from planned activities, such as flight plan, forecast weather available when dispatching the flight or expected airspace and airport congestion, to actual realisations, such as flight performance data recorded in the flight data monitoring on-board systems (FDM), actual weather or holding times. Therefore, a natural (and in trend) approach is to use these data for modelling to create predictions on operational parameters with machine learning (ML) techniques.

Dispatcher3[1], a CleanSky2 Innovative Action, aims at achieving this, developing a software prototype for the acquisition and preparation of historical flight data and use machine learning techniques to support the optimisation of future flights. Two prediction horizons are considered: longer look ahead predictions aiming at supporting the duty manager with models trained on D-1; and short-term predictions pre-departure (3 hours prior EOBT) to provide advice to the crew on what to expect in the flight aiming at supporting tactical decisions. These short-term predictions could be integrated on trajectory optimisers such as the one developed in CleanSky2 Innovation Action Pilot3 [1],[2].

The use of ML in this field, however, presents a set of challenges that will be explored in this paper. This paper focuses on the description of these challenges which could provide insight for similar projects in the field of air transport:

- Large datasets should be prepared before being used for modelling requiring data preparation pipelines.
- The training of machine learning models has its own challenges such as data leakage or data availability.
- Finally, individual predictions of complementary processes need to be integrated into a comprehensive view suitable for the end user.

The architecture the prototype is presented in Section II. The datasets used and information on how they are processed in Section III. The approach to develop the individual machine learning models and the challenges faced are summarised in Section IV. The paper then focuses on how the individual models are integrated into an advice generator system with some aspects that should be considered in this process in Section V. Finally, the paper closes with conclusions and further work laid out in Section VI.

[1] www.dispatcher3.eu - https://cordis.europa.eu/project/id/886461 (Accessed October 2022)

3AF
Association Aéronautique et Astronautique de France
TSAS2022
"Towards Sustainable Aviation" Summit
Toulouse • France - 18-20 October 2022

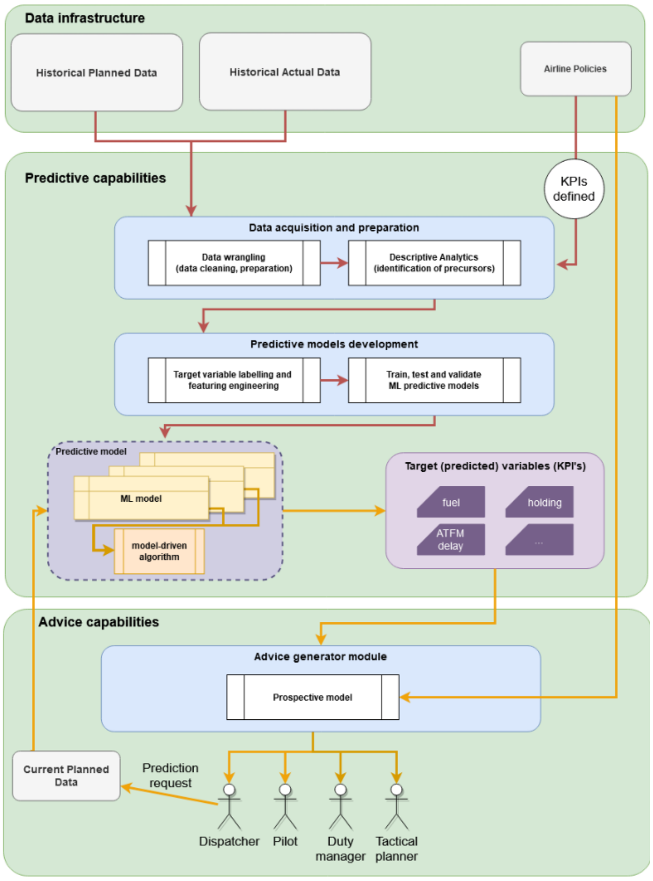

Figure 1. Dispatcher3 architecture

II. Dispatcher3 Architecture

Dispatcher3 is organised in three layers as depicted in Figure 1:

- Data infrastructure to support the storage and management of datasets required to train the models.
- Predictive capabilities, which comprises the pipelines required for the data acquisition and preparation and machine learning individual models definition, training and validation.
- Advice capabilities, which uses the individual machine learning models trained in the previous layer to present the information to the end users in a comprehensive manner. This might require the definition and execution of model-driven predictions supported by the individual machine learning models.

1) Data infrastructure

Different data sources are required in Dispatcher3, as shown in Section III. An iterative process has been used to identify and acquire datasets as required by the different models and development needs. These data are stored and managed in a data infrastructure set up in Amazon Web Services (AWS).

2) Predictive capabilities

The predictive capabilities are developed in two processes:

- Data acquisition and preparation, composed of:
  - Data wrangling (preparation and cleaning), which focuses on the acquisition of the data and their incorporation into the data lake. Once it is acquired it needs to be cleaned and prepared so that it can be used for the data analytics.
  - Descriptive analytics, using data mining techniques to extract the KPIs that will be used as target variable (variables to predict using ML models).
- Predictive model development, which consists of
  - Target variable labelling and feature engineering: Supervised machine learning algorithms work by training models based on a set of labelled data. The datasets are annotated following the KPIs defined as the result of the descriptive analytics. Also, the selected precursors must be engineered from the raw data, calculating the variables necessary.
  - Train, test and validate ML predictive model: These activities consist of the actual training of the model which provide the predictive capabilities.

Two predictions horizons are modelled:

- Day prior operations (D-1) focusing on:
  - Probability of being regulated due to ATFM
  - Location of regulation of ATFM (aerodrome or airspace) if regulated
  - Probability of being assigned a positive delay, i.e., non-zero, if regulated
  - Amount of ATFM delay
  - Block time
  - Turnaround time (if no regulated by ATFM)
- Pre-departure (H-3) with prediction on:
  - Expected runway at arrival
  - Fuel usage
  - Probability of holding at arrival
  - Amount of holding at arrival

3) Advice capabilities

The outcome of the individual models developed as part of the predictive capabilities might present some discrepancies and uncertainties that need to be considered to provide meaningful support to the end users. The advice generator, within the advice capabilities layer of Dispatcher3, focuses on how to present this information to the end user.

III. Data preparation techniques

A. Data transformation techniques: ETL and ELT pipelines

For data centred projects, the definition of an ETL (Extract, Transform, Load) pipeline is a requirement in order to properly treat the different data sources. An ETL pipeline is the set of processes used to move data from a source or multiple sources into a database, or data lake. Data shall be standardized and always be available. For this matter it is also important to have a well-structured data lake, where data can be stored raw and pre-processed.

Therefore, the objective of a ETL is to extract data from all data sources, clean them, translate them into desired formats, and normalize them for use in the case studies. Data cleaning is typically performed during the ETL phase, which aims to identify missing or erroneous data values, substitute or correct them, and then provide clean data ready for use further in the machine learning pipeline.

In Dispatcher3, we move away from the traditional ETL paradigm towards a more flexible ELT (Extract, Load, Transform) paradigm. This approach transforms the data after it has been loaded into the data lake and thus it enables the analysts and data scientists to adapt the dataset later on in the project as needed to different applications. This method is enabled by the cloud-based services which do not suffer from the possible problem of *data explosion* due to the fact that storage and computation are much cheaper than in the past. This method gets data in front of analysts much faster than ETL while simultaneously simplifying the architecture.

B. Data sources and processing techniques

Table I provides a description centred on the quality of the different data sources used. It is important to remark that several transformation techniques are required for each dataset such as: data integrity, filtering, missing values, out of range measurements and outliers identification and removal, units homogenisation, formatting of variables and key homogenization. Note that not all these processing techniques are always required (e.g. for some datasets they might not be needed in all instances (or in all case studies)) or loaded/stored in the data lake after being performed. In some cases, the data processing can be performed on-line as part of the features computation for training the models

Finally, for large files containing worldwide information (such as weather GRIB files or ADS-B files) it is important to select data according to different case-studies to reduce the dimensionality of the required datasets.

IV. Predictive modelling

A. Machine learning model developement

The typical machine learning model development pipeline consists of the following steps:

1) Data wrangling: The first step is to filter and merge the different sources of data (previously cleaned in the data preparation stage) to produce a unique dataset. Each task may need a different subset of data, depending on the target variable.
2) Data visualization: Visualizing data is a great way to gain insights. In particular, correlation plots and pair plots allow us to obtain information about the relations among variables and identify the degree of correlation between some features and the target variable.
3) Feature and target engineering: Once the training set has been generated, the data has to be processed so their representation is optimal for the estimators. At this stage, the features are usually divided in numerical and categorical to apply different techniques to each group: The numerical columns

TABLE I. Datasets

Data set	Description	Quality and characteristics
FDM provided by Vueling	Contains performance data and static data directly collected by the airlines	Sampling period very low Parameters recorded with different sampling rate The information is trustworthy and quality is overall good, however there are variables whose values are missing Required decoding before being used
ADS-B data by OpenSky	Radar data, contains trajectories	Noisy data Sampling period low
METAR	Contains weather information at different European Airports	Sampling period high, 30 minutes Missing values often (might rely on manual definition/recording) Units might differ for different data sources Dynamic variables might often be filled with same value (data integrity issues)
NOAA weather forecasts	GRIB file containing meteorological information in a 4D space (latitude, longitude, altitude and time)	Large files (GRIB files contain a lot of information) It is important to filter them according to different case-studies in order to reduce dimensionality
ALLFT+	Network and flight plan and trajectories information from DDR2 by EUROCONTROL	Allft+ files contain large information, it is important to select the desired features to reduce dimensionality
ECTL R&D Archive	Network information and flight information provided by EUROCONTROL	Sampling period high Good quality
Vueling flight data and flight plans	Flight information and flight plans from Vueling flights.	Sampling period high Good quality

have their missing values input with different strategies (e.g. mean, median, a fixed value) and are scaled so the algorithms that rely on distances (like K-means) can be correctly used. The categorical values also have their missing values imputed, but with different techniques (like using the most frequent value) and then have to be encoded following strategies like ordinal encoding, one-hot encoding or target encoding.

4) At this point, the models can be trained. Regardless of the chosen model, a set of values called hyper parameters must be tuned. These parameters affect the performance of the model and its behaviour. Therefore, they are crucial to correctly perform the chosen task. To check if the chosen features are significant to the task, a feature importance stage can be added. This ensures that the chosen features are tested so their relevance to the predictor can be assessed.

5) Once the model returns an acceptable result, it is stored along with any other useful information (e.g. parameters, metrics, graphics). This way, it can be deployed and integrated in a production environment.

It is worth noticing that steps 3 to 5 are iterative, that is, they are repeated over time. This is done to ensure the quality of the final model, since models are continuously tested to find improvements.

B. Machine learning challenges in the context of Dispatcher3

Novel AI techniques have been developed in the past years that push the boundaries of the accuracies of the machine learning models, computational cost, data representation and information extraction and automatisation of the ML pipelines, among others. GANs (Generative Adversarial Networks) have been successfully applied to generate novel realistic datasets with the same statistical properties as the training set through two competing neural networks against each other. A lot of progress has been made in the area of machine learning over graphs, especially in creating memory-optimised ways to encode information and feed it to the machine learning models. AutoML is enabling non-machine learning experts to successfully use and train simple machine learning models in various applications, or to quickly generate a baseline model in a project and get the team started, by automatising a lot of the time-consuming, iterative tasks of machine learning model development. It allows data scientists, analysts and developers to build ML models with high scale and efficiency while sustaining model quality.

However, these techniques need to be applied with caution and there are a set of challenges that need to be addressed. Some of them are specific to the domain in which Dispatcher3 applies. These can be summarised in:

1. Avoiding data leakage and data availability.
2. Machine learning model selection and tuning.
3. Uncertainty modelling considerations.
4. Prediction of non-observed in historical dataset actions.
5. Models explainability.
6. Data drift on exploitation models.

1) Avoiding data leakage and data availability

a) Data leakage

Data leakage refers to a situation in which a model uses information that it is not supposed to have or would not have in real-world settings, and it can occur in many ways, some more obvious than others. In recent years, more research on data leakage has been performed and firmer practices were established to prevent it. These include, but are not limited to:

- The need of being aware of the date and time of the availability of the data that is fed into machine learning models. That is something we strive to document exhaustively under model assumptions in Dispatcher3 due to the fact that often we will not be able to have exact information on when some data sources will be available to the potential users of the models developed in Dispatcher3, where the models put into operation in real-world settings;

- Beware of temporal leakage. Temporal leakage can enter the datasets when constructing training and testing datasets by sampling them in a way that would lead to not truly independent training and test sets. For example, when the split between the training and test set is not carried out sequentially.

- Avoid oversampling leakage. This can occur in situations of imbalanced datasets where there is a need to perform minority oversampling, e.g. using SMOTE algorithm. If this oversampling is performed before splitting the datasets into training and test portion, information leakage could happen.

- Leakage due to aggregation. This type of leakage can happen during the pre-processing state if the training and test dataset are grouped and normalised, which in turn leads to the leakage of aggregated statistics from the train to the test dataset.

b) Data availability and prediction horizons

It is crucial to ensure that the data used for the training of the machine learning models are available during their execution. One of the challenges of the development of these types of models in the field of aviation is that datasets tend to contain historical realised data, but it is very difficult to know which data were available at a given moment. This is particularly relevant for traffic and network related data. Both planned and realised data evolve over time (e.g. new flight plan submissions, flights cancellations, flights which have not submitted their flight plan yet, ATFM regulations which are issued at a given time, ATFM regulations which are cancelled). It is relatively easy to access datasets which contain the final flight planned trajectory and the realised flight, but it is

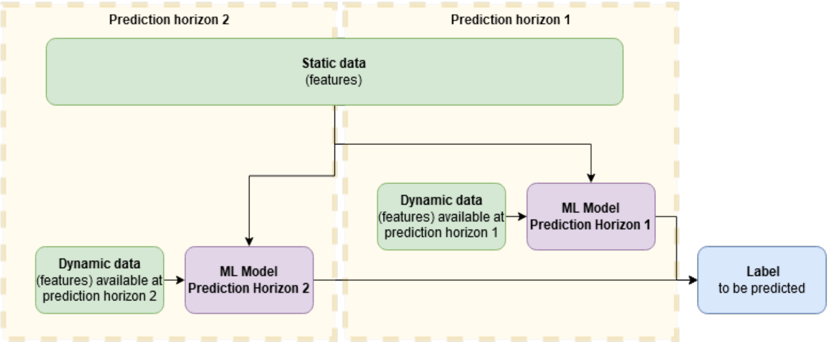

Figure 2. Prediction horizons with static and dynamic data models traininng

difficult to know at a given moment in time which flight information was available for a particular flight (and for the remaining flights in the network).

In Dispatcher3 prediction horizons are defined to identify when the machine learning models will be executed with respect to the flights. As shown in Figure 2, the same label can be predicted with the information available at different horizons using distinct models. For example, the probability of a flight being affected by ATFM regulation (label to be predicted which can be obtained from historical data) could be predicted X hours prior to the schedule of the flight (prediction horizon 1) or the day prior operations (prediction horizon 2). Historical datasets will allow us to label if the flight was regulated or not, i.e., the label to predict (the regulation of the flight) is the same for both models, but the data available to compute the features from which to predict the labels might vary at these horizons.

It is useful to differentiate between *static* and *dynamic* data (and features). Static data (and features) do not vary as a function of the prediction horizon (as depicted in Figure 2). Examples of these data are origin and destination, aircraft type or time of the day when the flight is scheduled. Dynamic data might be different at different horizons. For example, the weather forecast at arrival might be updated over time, or the expected demand at arrival airport might be different if the flight plans from other flights are available or not. This distinction allows us to estimate the importance of the dynamic features on the overall performance of the algorithms and therefore the relevance of the prediction horizon for a particular problem.

Some assumptions over the dynamic data might be required (e.g. which traffic demand is assumed to be available at a given horizon). For this reason, analysing the impact of these dynamic features on the predictions is very important. Moreover, we could train the model only at a given horizon but use the available information at a different one to generate the prediction. In the example of the probability of being regulated due to ATFM, the model could be trained using for example the information on demand assuming that all flight plans are available, but then on execution the most up to date information would be used instead. This might lead, naturally, to some underperformance of the models as the input data used to compute the features for the prediction is not the same as the data used for the computation of the features used to train the model. A difficult aspect is the quantification of this degradation.

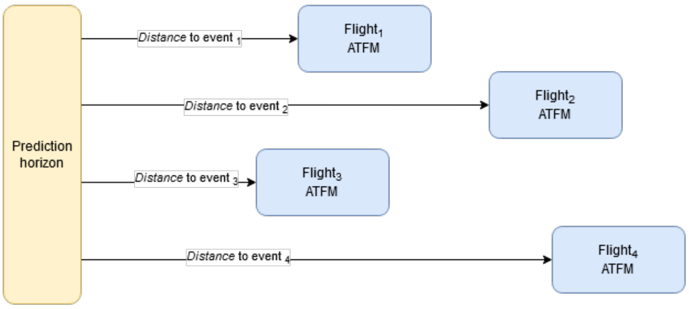

Figure 3. Prediction horizon at a given time (e.g. at 9h)

c) Data alignment over the prediction horizons

Different approaches can be used to define the prediction horizons, and these have an impact on the quality of the datasets used to train the models. Each approach has some benefits and drawbacks, and different problems might benefit from different considerations. Once a prediction horizon is defined the data available at this horizon should be defined (either by gathering it from historical records or by defining some assumptions on the data to be considered being available at a given period).

i. Preiction horizon fixed at a given time

The first simple approach is to define a given time (as shown in Figure 3). For example day prior operations or at 9h00 on the day of operations. The main idea is to gather the data available at that given time and use this to compute the features required to estimate a given target variable. In some cases, the definition of this horizon might have some ambiguity (e.g. when exactly is the horizon at day prior operation? which datasets are considered to be available?). Some of these ambiguities can be mitigated by ensuring that datasets used are available within the defined period.

Once the prediction horizon is defined all data available can be gathered to compute the features. For example, flight plans,

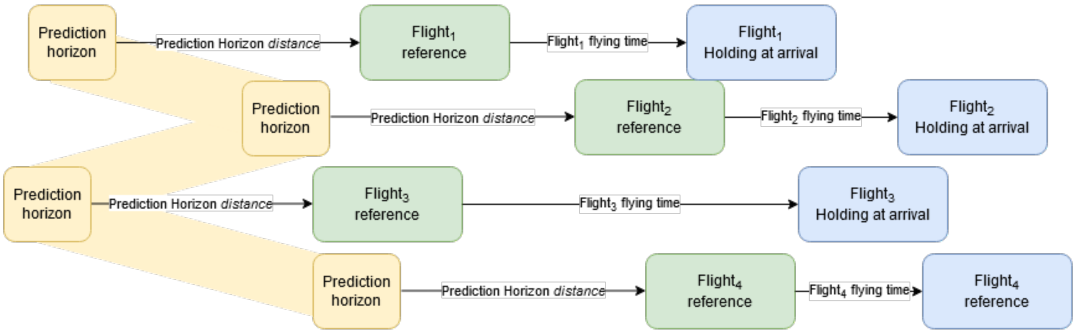

Figure 4. Prediction horizon with respect to flight reference (e.g. SOBT)

weather reports, or information on airspace configurations, available at that given moment. Some assumptions might be required regarding the data available. For example, if the horizon is set at 9h00 and the demand at the expected arrival time at a given airport is estimated as a feature for a flight, some of the flights arriving at the airport defining this demand might already be flying, others might have already flight plans submitted, while for others their schedule might only be available.

Figure 3 presents an example of a prediction horizon defined at a given time to predict the probability of ATFM. Note that this approach is valid also to predict aggregated data, for example, expected holding at arrival at a given time aggregating all flights in that given time.

Two drawbacks can be identified: first, for every event that is to be predicted (e.g. probability of ATFM in the example of Figure 3) the temporal distance between the prediction horizon and the event might be different for different observations. This could be an issue as the same feature (e.g. expected congestion at arrival for a given flight or expected weather) might be computed with different degrees of accuracy. Adding features which characterise this distance might improve the performance of the models.

Second, if the model is executed outside the trained moment (e.g. executing the predictions with information available at 11h for a model trained at 9h) there will be a shift on dynamic features from the execution horizon to the trained horizon. This discrepancy between the dataset used for training and the dataset used for execution might produce some reduction in performance which might be difficult to quantify. Different models could be trained at different prediction horizons and then different approaches could be used: using the model which is closest to the execution time, interpolating the results of the closest models, etc.

ii. Prediction horizon defined at a given distance with respect to flight reference

As the models are trained to do a prediction for a given flight, e.g. predicting the expected holding at arrival for a given flight, the prediction horizon can be defined with respect to a reference linked to the flight. For example, 3 hours prior SOBT. This has the significant advantage that the models are operationally easy to utilise. This simple definition indicates that the model will be valid to do predictions x hours prior SOBT.

However, this can lead to some issues relating to a misalignment of the different observations used to train and execute the models due to the distance between the reference point (x hours prior SOBT) and the phenomena to be predicted. As in the example of Figure 4 where the variable to be predicted (expected holding at arrival) depends on the conditions of the airspace/airport at arrival of those flights. This means that due to the different planned flight duration, flights which will experience similar congestion (and therefore probability of holding in this example) will be scheduled at different times (shorter flights with a shorter distance to the event). Therefore, the prediction horizon will be misaligned for different observations and if for example current weather observed at the arrival airport is used as a feature, for some flights that weather will be many hours before the others.

This is not a problem on itself and with enough data and adding information which, somehow, encapsulates this distance between the prediction horizon and the factors affecting the event to be predicted, the potential impact of this misalignment can be mitigated.

However, one needs to be cautious when datasets are imbalanced, e.g. if the number of short-haul fights is much larger than long-haul flights, and it would be interesting to analyse the performance of the models as a function of this distance between prediction horizon and event to be predicted.

Note that as in the previous case, it could be possible to filter the samples to keep the ones which have an aligned target event. For example, by creating a model with a prediction horizon X hours prior the SOBT of the flights for flights arriving to their destination in a given arrival window, so that conditions at arrival are similar. The utility of this might be small and allowing the model to infer if these features are relevant might be a more suitable approach.

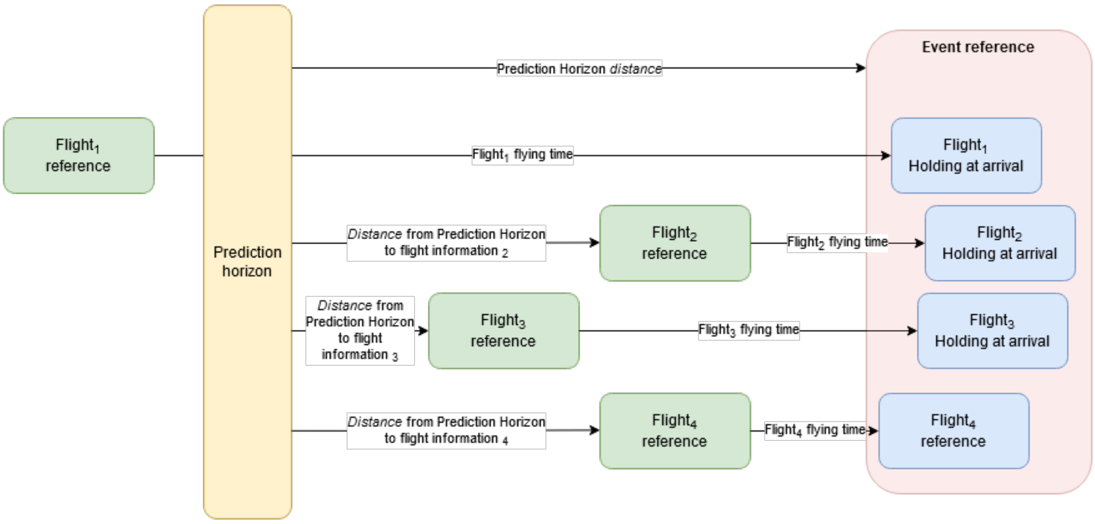
Prediction horizon
Prediction Horizon *distance*
Event reference
Flight$_1$ reference
Flight$_1$ flying time
Flight$_1$ Holding at arrival
Distance from Prediction Horizon to flight information $_2$
Flight$_2$ reference
Flight$_2$ flying time
Flight$_2$ Holding at arrival
Distance from Prediction Horizon to flight information $_3$
Flight$_3$ reference
Flight$_3$ flying time
Flight$_3$ Holding at arrival
Distance from Prediction Horizon to flight information $_4$
Flight$_4$ reference
Flight$_4$ flying time
Flight$_4$ reference

Figure 5. Prediction horizon with respect to event (e.g. time when holding at arrival is produced)

iii. Prediction horizon defined at a given distance with respect to event

A final possible approach is to define the prediction horizon with respect to the event which is targeted to be predicted. For example, as shown in Figure 5, estimating the expected holding at arrival x hours prior the arrival. This approach might be more suited for models which are not flight-centred but event-centred. For example, it could be considered that the probability of experiencing a holding does not depend on the characteristics of a given flight but more on the conditions of the arrival airport at the time of arrival of the flight. In that case, one could define the problem as estimating the expected holding time for flights arriving at the airport at a given time-window (e.g. arriving around 9h) with the information available x hours prior that time.

Note that in this case, depending on the horizon, some flights might already be flying, for others might have a flight plan already submitted, while others might still be only scheduled. This might render the prediction for a specific flight difficult and be more suited to estimate for example the situation of the infrastructure, e.g. expected average holding/delay at arrival at a specific time given the situation of the ATM network x hours before.

d) Data verification

Assumptions tend to be incorporated into the models already in the stage of data acquisition, transformation, and ingestion (typically performed by data engineers). However, those processes often are not documented well and ETL or ELT pipelines. This can become very problematic when a model wants to be validated by the users because doubts or unclear model results trace back all the way to the initial stages of data collection.

In Dispatcher3, for these reasons we intend to document as much as possible the processes of data collection and the description of raw data sources, as well as transformations performed on the datasets to yield model-ready data.

2) Machine learning model selection and tuning

a) Hyper-parameter tuning.

Choosing the right set of hyper-parameters for a model for a particular application is one of the key choices in machine learning model development. This represents a very computationally expensive problem, and it varies greatly not only from one model type to another, but also from one application to another. An experienced data scientist might ad-hoc choose an appropriate set of hyper-parameters for an application and a model and perform fine-tuning from that set as a starting point, achieving satisfying performance.

However, in most cases, the data scientists do not know what a good starting set of hyper-parameters would be, and performing a full search over a whole grid of possible combinations of parameters is very time-consuming. Therefore, and in line with the experience of the machine learning community, the approach proven as most effective is a combination of a random search of hyper-parameters within a manually assigned range (relying on the heuristics set by the empirical observations for each model). This yields a computationally not overly expensive method that usually gives very good results.

b) Choice of the ML model

It is often difficult to select the appropriate machine learning model to use for a specific application. Indeed, the only thing one can rely on is empirical research on the performance of the models and often that can vary significantly from one field to another. Most advocated approach recently is to keep it simple approach: data scientists are encouraged to try simpler models first, establish a baseline model that can be used for the performance assessment and run an exhaustive comparison of models throughout development iterations before advocating for specific methodologies.

c) Data-centric vs. model-centric approach

Often times in the presentations and papers on novel machine learning techniques, more complicated architectures are being advocated due to alleged superior performance. However, in most cases the performance improvement comes from better formulations of the problems, better data pre-processing, feature engineering, or simply working on data quality itself (rather than trying to find that perfect model).

3) Uncertainty modelling considerations

The characterisation of this individual uncertainty (or error) on each prediction is paramount in many fields. For example, uncertainty can rapidly grow when applying the outcome of these models in dynamic and unstable systems, or when their outcome is combined with other models. This is the case for many applications in the field of Air Traffic Management when integrating the prediction models into airlines and air traffic control support decision tools. Supervised machine learning models can generate the relationship between input (features) and target variables from a training dataset. Once a model has been trained, some error is expected between predicted and actual realisations of the target variable. This error accounts for both aleatory uncertainty in the phenomena being modelled and epistemic uncertainty in the capability of the model to represent the relationship between features and target variables. The homoscedasticity of the error on the predictions by the model cannot be always assumed for several reasons: the training set could be more or less dispersed on different regions of the feature space; the underlying processes and relationships being modelled could present aleatory uncertainty; and the machine learning model might have limitations which could produce more accurate predictions on different regions of the feature space. For this reason, averaged statistics and the distribution of the error on the predictions for the entire validation set cannot generally be used as an estimation of the uncertainty of a single prediction. The local uncertainty of the model could be different than the average dispersion of the error and even present some skewness.

Different approaches have been suggested in the literature to overcome these limitations and to estimate the uncertainty and reliability of the individual predictions, such as: sensitivity analysis on the models [3], delta method based on nonlinear regression [4], Bayesian method [4], bootstrap method developing several neural network models with subsets of training set [5], local neighbourhood prediction interval using clustering techniques [6], mean-variance estimation method [7], gaussian processes [8], [9] or quantile regression, which estimates multiple quantiles simultaneously [10],[11],[12],[13].

Most of these methods provide either an estimation of the variance of the error or an interval of reliability but are not able to describe the distribution of possible values. In Dispatcher3, we propose the use of a probabilistic classifier to characterise the distribution of the error of a prediction relying on the estimation of this error on the training set, obtaining the discrete distribution of the possible expected values of the prediction [14]. This will be done particularly for ML models which need to be integrated into other higher-level models as the modelling of the propagation of uncertainty becomes paramount on those cases; or when translating indicators, e.g. delay into cost due to the non-linearities of cost of delay [15]. Having only the expected value without its distribution will probably lead to underestimation of cost.

When this integration is not required and only the qualitative indication of the uncertainty is to be transmitted to the end user, the problem can be translated into a discrete probabilistic classification problem with definition which are operationally relevant. This is the case for holdings at arrival, where instead of estimating the expected exact amount of holding, the probability of experiencing different qualitative amount of holdings (minor, high, severe) are used.

4) Prediction of non-observed in historical dataset actions

Supervised machine learning models are trained to predict labels as a function of features. These are computed based on historical datasets. This means that it might be difficult to predict the impact of situations which are not available in the training datasets. Depending on how the models generalise this issue might be more or less acute.

Airlines and ATM operations are being monitored throughout the day and actions are performed to mitigate negative situations. For example, an aircraft might be swapped with another, a flight cancelled to prevent the propagation of delay, or flights might resubmit a flight plan to avoid a congested region. If these actions are not recorded, then it is difficult to generate a predictive system which allows the user to advance when these negative situations which require their intervention might be needed.

One of the ways of mitigating this is the integration of the outcome of ML models into higher-level models which can simulate the *do nothing* approach. For example, by propagating estimated delay through the day so that the impact on reactionary delay of not intervening on the planning can be estimated [14]. This enables the possibility to identify situations which if action is not performed undesirable outcomes are obtained. Directly using machine learning models could be difficult as in the historical datasets the actions of the duty managers would already be recorded.

5) Models explainability

One of the most discussed topics in the industry powered by AI is the explainability of such advanced AI models. When it comes to more complex architectures that are being used in various applications, the models itself is a black box and it can be difficult to disentangle how the model arrives at a certain conclusion (forecast). Indeed, most ML models do not provide insight into their inference process. However, and that is especially prominent in aviation, as we wish to integrate AI powered systems into decision-making processes and allow them to assist human workers in their operations, the need for explainability rises. To have a solid human-machine interface that allows for partial levels of automation and human users to interact with intelligent systems in a meaningful and perhaps collaborative way, model explainability is a key component.

6) Data drift on exploitation

The main assumption of ML pipelines is that the data samples used for training are independent and the distributions of training, test and evaluation data sets identical. However, in real-world setting the statistical properties of the target variable the ML model is trying to estimate change over time, which is often referred to as 'data drift'. This is something that becomes a large problem when putting ML models into production.

V. Advice Generation

A. Overview and motivation

As previously presented, Dispatcher3 develops a set of individual machine learning models to predict specific indicators at given prediction horizons for flights. During the airline operations the duty manager, dispatcher and pilot need to access this information. The relevance (and validity) of the predictions will evolve over time. Moreover for the same flight different predictions relating to similar processes will be produced. These need to be integrated in a comprehensive interface/visualisation so that they can be used to improve the operational decisions and situational awareness.

Figure 6 presents the different individual ML models. These provide information at different prediction horizons: with tactical pre-departure predictions, such as expected holding at arrival, and pre-tactical information, such as probability of ATFM delay being issued.

Each of these individual models will produce a prediction which will be either a probability, an expected value, or a distribution of possible values. As depicted in Figure 7, the outcome of some of these models needs to be collected so that a comprehensive view is provided: tactical pre-departure information, and ATFM information. Some of these models can be combined in a more complex manner to produce further predictions: reactionary delay integrator. All these models will be underpinned by the flight operations plan and its visualisation.

In this flight operation plan, the different flight, and their rotations (planned and realised) will be visualised integrating the outcome of the different models.

With these considerations, the advice generator has three main objectives:

1. Collect all the different individual prediction models into an integrated system. This needs the identification of which models are suitable for each of the planned flights at a given moment in time.
2. Integrate the outcome of the different models into a single visualisation. This will require the consideration of uncertainty in the predictions and complementarity of individual ML models.
3. Use the individual machine learning models to generate a forecast of non-observed actions in the historical dataset, such as reactionary delay or probability of breaching a curfew.

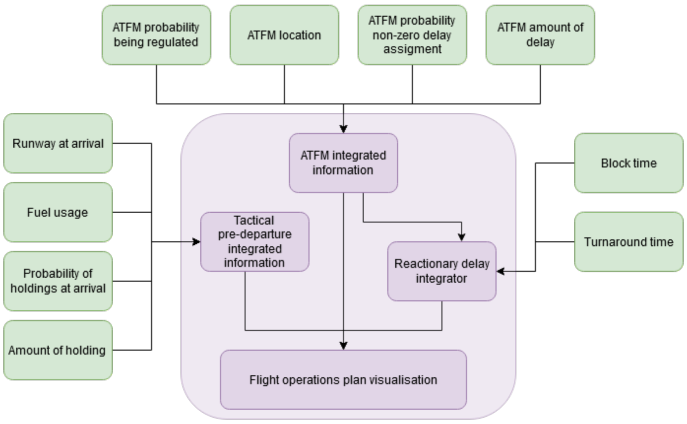
ATFM probability being regulated
ATFM location
ATFM probability non-zero delay assigment
ATFM amount of delay
Runway at arrival
Fuel usage
Probability of holdings at arrival
Amount of holding
ATFM integrated information
Tactical pre-departure integrated information
Reactionary delay integrator
Flight operations plan visualisation
Block time
Turnaround time

Figure 6. Advice generator integration of different ML models

B. Information representation

There are two aspects that should be considered when presenting the information: the level and how the information is presented.

1) Level of information presented

It is paramount to present the information with the right level of detail. As mentioned previously, the advice generator will rely on the visualisation of the flight plans and their rotations as depicted in Figure 7.

Figure 7 presents an example of the visualisation of the different rotations for a set of aircraft throughout the day. Four sets of flights are identified at a given moment in time:

- Already flown, which have landed prior to the current time.
- Flying, which are being currently operated.
- At pre-departure. These flights are close in time, their final flight plan has already been generated, and if pre-departure aspect will impact them such as ATFM regulations these will already be known. For these flights, pre-departure models will be available (e.g. probability holding at arrival).
- Planned, which are planned flights for which new flight plans might be generated. For some of these flights if they are affected by ATFM regulations might not be known yet, and therefore predictive models could be useful.

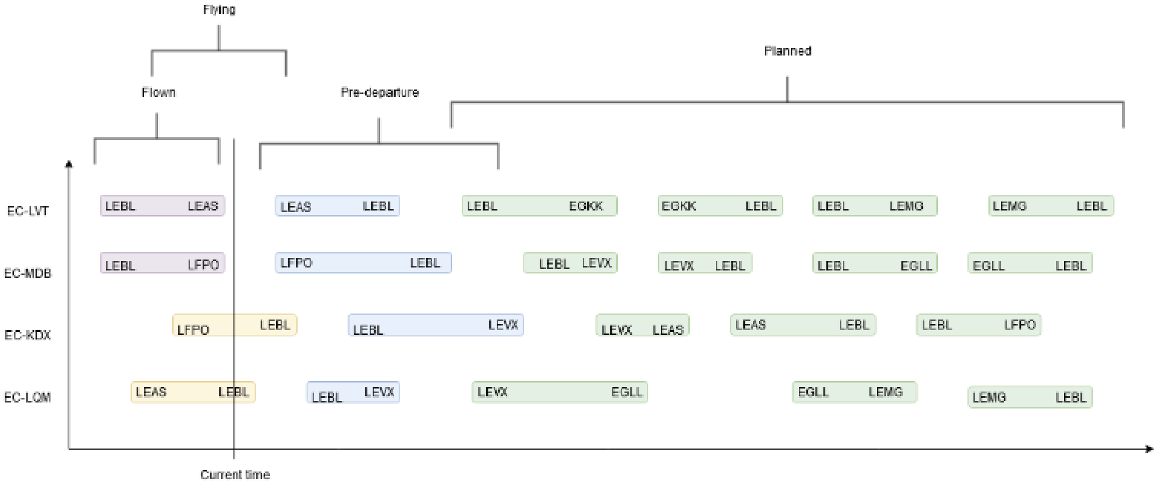

Figure 7. Visualisation of planned and exsecuted flihgts and rotations

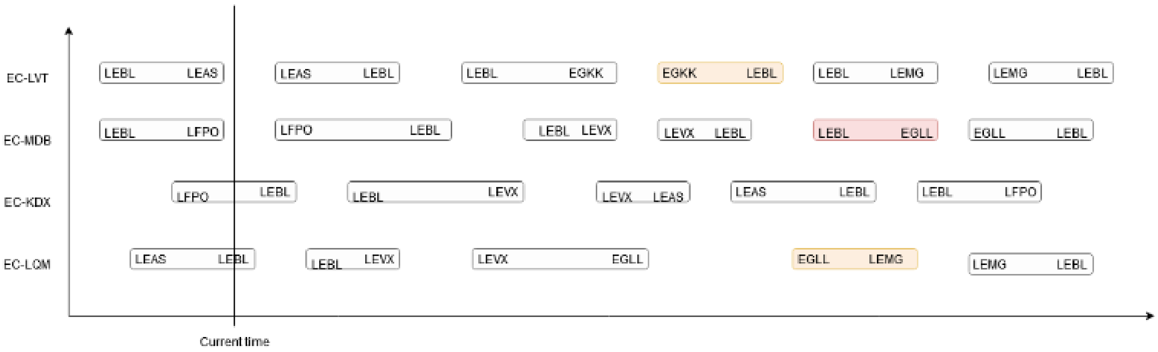

Figure 8. Visualisation of robability of being affected by ATFM regulation

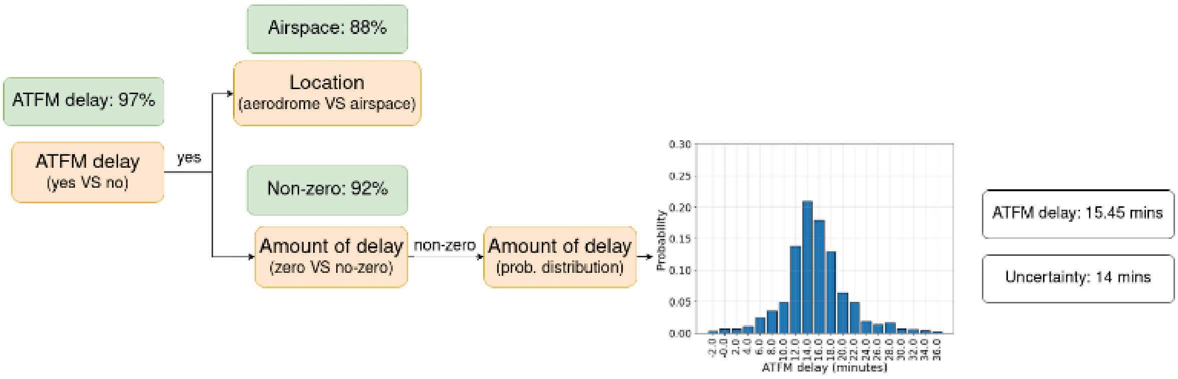

Figure 9. Visualisation of probability of being affected by ATFM regulation [16]

Note how the flights belonging to the different sets will change as time evolve. The visualisation will evolve as new flight plans and operations are generated. This high-level view will be useful for the duty manager to easily identify flights that are impacted by different operational aspects as predicted by the models. For example, Figure 8 presents how the probability of being impacted by ATFM regulation could be presented for the *planned* flights.

If a flight is selected then more detailed information could be provided, for example integrating the ATFM information available from the different models as depicted in Figure 9.

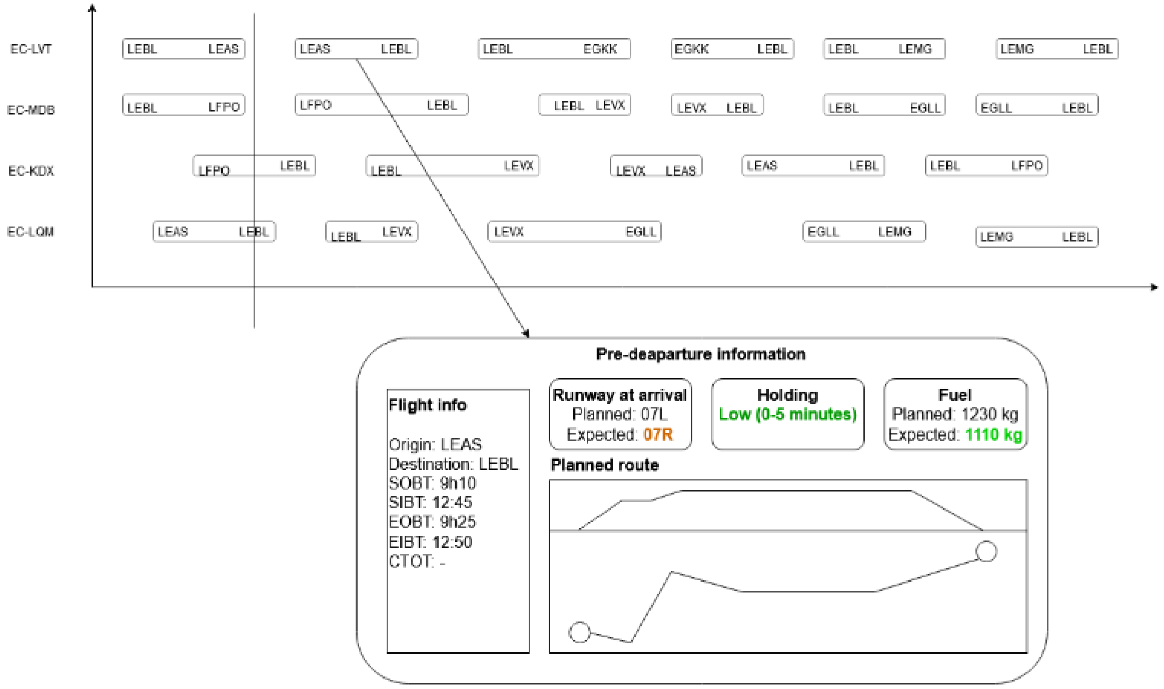

Figure 10. Pre-tactical information representatio

Similarly, the information for the tactical execution of the flight can be aggregated into a single visualisation which can be shared with the crew pre-departure (as shown in Figure 10. Note this is a representation on how information could be presented not how it might actually be implemented in the final release of Dispatcher3).

2) *Representation of data*

Besides providing the right information at the right level it is critical to consider how this information is presented, particularly with the consideration that the models will have uncertainty. As discussed previously the simplest approach to represent uncertainty of the models is to characterise their error, in Dispatcher3, for some models, we have also considered the possibility to directly predict a probability distribution of possible expected values. Finally, when a model is classifying between alternatives, besides the error of the model the probabilities of the classification could also be used as an approach to determine the certainty of the model on its predictions.

The advice generator (and the models) could consider different approaches:

- Provide the expected value without any further information. This could lead to actions by the end users which are not desirable as the quality of these predictions and uncertainties are not considered.

- The expected value could be accompanied by additional information encoded in the visualisation (e.g. using colours to indicate the certainty of the predictions).

- Use discretised predictions instead of expected values (e.g. high, medium, low categories instead of a value for the expected holding at arrival).

- Capture the uncertainty in a value and provide this along the expected value.

Dispatcher3 will model some of these and gather feedback from experts on the most suitable representation.

C. *Integration of ML models into advice generator*

Finally, as the advice generator is in charge of the visualisation of the planned flights and the outcome of the individual machine learning models, it will be the advice generator the component of Dispatcher3 which will execute the individual models with the data available in the system at a given moment.

VI. Conclusions and Future Work

Machine learning models are gaining traction in the aviation community to support flight operations with the objective of minimising operational costs and the environmental impact of aviation, i.e., improving the decision-making process within airlines.

These models tend to be *narrow*, i.e., tackle a specific indicator to be predicted within a given prediction horizon. The individual machine learning models could be integrated in either support systems, e.g. trajectory optimisers, or directly present the information to the end users. Moreover, in some cases, the outcome of the different models should be combined in a model-driven architecture to produce the relevant information, and not all models are valid for all flights in all operational situations. This strengthens the need for an architecture which incorporates an integrator of models as the one suggested in the advice generator in Dispatcher3.

The development of machine learning models requires the deployment of a data management infrastructure and well-defined data preparation pipelines. Several challenges to develop the individual models have been highlighted in this paper ranging from *common* issues on machine learning such as models explainability or data drift on exploitation to more relevant aspects to aviation such as the clear definition of prediction horizons and modelling of uncertainty.

In aviation, historical datasets tend to contain a snapshot of the final planned operations. It is therefore difficult to know at a given moment in time the information which is available as flight plans tend to evolve over time. For pre-departure models (3H prior SOBT) this might not be necessarily an issue as the final flight plan is defined, but when larger lookahead times are seek this could be a complex task. We have found that a good approach is the division of features between static, which do not evolve over time, such as origin and destination airports, and dynamic features, which might depend on the information at the prediction horizon, such as weather data. This allows modellers to quantify the benefit of using the dynamic data and the potential error introduced by modelling using data which might slightly differ on operations. The paper presents different approaches towards the definition of the prediction horizons.

Uncertainty is paramount when dealing with flight operations. Airlines tend to be conservative with respect to their operations and the non-linearities of cost of delay means that prediction of just expected values are fully suitable. We suggest the use of probabilistic distribution as outcome of the models to capture these uncertainties. The discretisation of the target variable (e.g. holding at arrival) into operationally sound categories (e.g. mild, medium, severe) is also a suitable approach.

The estimation of non-observed data (e.g. reactionary delay if no action is performed) is also crucial to obtain information for the end user which are relevant to them, i.e., if no action is taken these flights might suffer this extra delay. The need of using models which incorporate the outcome of the different machine learning predictions is of high importance in this field.

Finally, the work on Dispatcher3 has highlighted the need to integrate the outcome of different machine learning models into a single interface with sometimes contradictory predictions.

Future work will focus on the finalisation of the different individual models and their integration in the prototype, and further exploring how the information should be presented to the end users.

Acknowledgment

This work is performed as part of Dispatcher3 innovation action which has received funding from the Clean Sky 2 Joint Undertaking (JU) under grant agreements No 886461. The Topic Manager is Thales AVS France SAS. The JU receives support from the European Union's Horizon 2020 research and innovation programme and the Clean Sky 2 JU members other than the Union. The opinions expressed herein reflect the authors' views only. Under no circumstances shall the Clean Sky 2 Joint Undertaking be responsible for any use that may be made of the information contained herein.

References

[1] Prats, X., de la Torre, D. and Delgado, L., In-Flight Cost Index Optimisation Upon Weather Forecast Updates, 41st Digital Avionics Systems Conference, Portsmout, VA, USA, 2022

[2] Delgado, L., de la Torre, D., Kuljanin, J. and Prats, X., Considering TMA holding uncertainty into in-flight trajectory optimisation, International Workshop on ATM/CNS, Tokyo, Japan, 2022

[3] Bosnić, Z., Kononenko, I., 2008. Estimation of individual prediction reliability using the local sensitivity analysis. Applied Intelligence 29, 187–203

[4] Khosravi, A., Nahavandi, S., Creighton, D., Atiya, A.F., 2011. Comprehensive review of neural network-based prediction intervals and new advances. IEEE Transactions on Neural Networks 22(9), 1341–1356

[5] Heskes, T., 1996. Practical confidence and prediction intervals. In: Proceedings of the 9th International Conference on Neural Information Processing Systems. NIPS'96, pp. 176–182. The MIT Press, Cambridge, MA, USA

[6] Shrestha, D.L., Solomatine, D.P., 2006. Machine learning approaches for estimation of prediction interval for the model output. Neural Networks 19(2), 225–235

[7] Nix, D.A., Weigend, A.S., 1994. Estimating the mean and variance of the target probability distribution. In: Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94), vol. 1, pp. 55–601

[8] Riis, C., Antunes, F., Gurtner, G., Camara Pereira, F., Delgado, L., Azevedo, C., 2021. Active learning metamodels for ATM simulation modeling. In: Proceedings of the 11 th SESAR Innovation Days, pp. 1–8

[9] Graas, R., Zun, J., Hoekstra, J., 2021. Quantifying accuracy and uncertainty in data-driven trajectory predictions with Gaussian process regression. In: Proceedings of the 11 th SESAR Innovation Days, pp. 1–8

[10] Amalia, F., Suhartono, S., Rahayu, S., Suhermi, N., 2018. Quantile regression neural network for forecasting inflow and outflow in yogyakarta. Journal of Physics: Conference Series 1028, 012232

[11] Koenker, R., Hallock, K.F., 2001. Quantile regression. Journal of economicperspectives 15(4), 143–156

[12] Meinshausen, N., 2006. Quantile regression forests. J. Mach. Learn. Res. 7, 983–999

[13] Moon, S.J., Jeon, J.-J., Lee, J.S.H., Kim, Y., 2021. Learning multiple quantiles with neural networks. Journal of Computational and Graphical Statistics 0(0), 1–11

[14] De Falco, P. and Delgado, L. 2021, Prediction of reactionary delay and cost using machine learning in proceedings of the Airline group of the International Federation of Operational Research Society (AGIFORS)

[15] Cook, A. and Tanner, G., 2015. European Airline Delay Cost Reference Values, Updated and Extended Values (Version 4.1), University of Westminster

[16] Mas-Pujol, S., De Falco, P., Salamí, E. and Delgado, L. Pre-tactical prediction of ATFM delay for individual flights, 41st Digital Avionics Systems Conference, Portsmout, VA, USA, 2022