Learning to Rank Flight Routes for Improved Air Traffic Demand Predictions

Ramon Dalmau, Pablo Gascó, Hamid Kadour, Éric Allard & Gilles Gawinowski
European Green Sky Directorate
EUROCONTROL
Brétigny-Sur-Orge, France

***Abstract*—Effective air traffic flow and capacity management (ATFCM) relies on accurate predictions of both traffic demand and capacity. This paper focuses on the former because it presents the greatest uncertainty. Traffic demand predictions are typically based on flight plans submitted by airspace users to the Network Manager. However, to optimise their flights with the most up-to-date information, many users delay submitting their flight plans until just a few hours before departure. This delay leads to the implementation of ATFCM measures with incomplete traffic information. Currently, missing flight plans are estimated using the PREDICT system, which performs its task effectively. However, it operates based on straightforward rules and does not account for factors such as air traffic flow management regulations, convective weather activity, or the business strategies of airspace users. Prior efforts to improve PREDICT have utilised complex, city-pair-specific data-driven models that encountered practical constraints due to insufficient training data. Moreover, these models were only capable of predicting the most likely flight plans from those observed in the past, without ensuring their validity in the current environment. The main goal of this paper is to presents an alternative methodology, which consists of modelling the decision-making processes of flight dispatchers when submitting flight plans, leveraging historical data and learning-to-rank techniques. Preliminary results are presented, along with a discussion of key challenges encountered and lessons learned, offering insights for future research directions.**

***Keywords*—Flight plan predictions; machine learning; ranker**

I. Introduction

Air traffic flow and capacity management (ATFCM) is a crucial element in ensuring the safe and efficient utilisation of airspace and airport resources. It involves monitoring and balancing traffic demand with the available capacity of airports and airspace. When predicted traffic demand exceeds the declared capacity, measures such as delaying departures through ATFCM regulations are implemented to prevent congestion.

The ability to accurately predict traffic demand – how many aircraft will be using a particular airspace or airport at any given time – is fundamental to effective ATFCM. This prediction largely depends on the flight plans submitted by airspace users to the Network Manager (NM), which provide essential data on expected routes (including waypoints, airways and flight levels), departure and arrival times, and aircraft types.

However, a significant challenge arises due to the timing mismatch between when ATFCM measures need to be implemented and when these flight plans are actually submitted. To be effective, ATFCM measures must be planned and executed well in advance, often hours or even a day before peak traffic periods. Conversely, flight plans are typically submitted much closer to the time of departure, sometimes just a few hours before the flight. This timing gap could lead to ATFCM measures that are based on incomplete traffic demand information. This mismatch could result in either overestimating or underestimating actual demand. Overestimation may lead to unnecessary ATFCM measures, causing avoidable delays and inefficiencies. Conversely, underestimation can lead to last-minute congestion, resulting in reactive measures that are often less effective and more disruptive for airspace users.

Addressing this timing mismatch is essential for enhancing the accuracy of traffic demand predictions and, by extension, the overall efficiency and effectiveness of ATFCM measures. This paper addresses this challenge by proposing a method that utilises historical data and learning-to-rank methods to estimate traffic demand based on typical patterns, even before flight plans are submitted. Although predicting traffic demand before the submission of flight plans based on historical data is not new, the current approaches have well-known limitations.

For instance, the NM currently predicts traffic demand as early as D-6 using the PREDICT tool, which relies on predefined rules to forecast flight plans, such as using the same flight plan filed by the airline for the same city-pair seven days prior. While generally accurate, these rules may overlook factors such as weather, the response of airspace users to ATFCM regulations, and the business strategies employed by flight dispatchers using their planning tools.

Sophisticated models based on machine learning, as found in the literature, also come with shortcomings. For example, many models are designed to operate on a single city-pair, requiring the training of a separate model for each city-pair. This approach is impractical from a machine learning operations perspective. Additionally, the data available to train each individual model is often insufficient. Furthermore, these models typically generate a flight plan directly from a catalogue of previously observed flight plans or “clusters”, without ensuring that the selected plan is valid[1]. This raises concerns about what happens if none of the historical flight plans are valid or if the network structure has changed.

[1]Here, the term ”valid” refers to Integrated Initial Flight Plan Processing System (IFPS)-compliant. The IFPS system validates each flight plan against the relevant Aeronautical Information Regulation and Control (AIRAC) data and route availability document (RAD) restrictions and if valid distributes the flight plan to the relevant actors, including concerned air traffic service units.

In this paper, we propose to monitor all flight plan submissions and changes. Each time a flight plan is submitted or altered, we generate several valid flight plan proposals that were available to the airspace user at that exact moment, as determined by the NM. We then extract key performance indicators (KPIs) for each proposal, such as fuel consumption, flight duration, distance, route charges, and ATFCM delay. The flight plan chosen by the airspace user is labelled as the preferred option, while the generated proposals are labelled as less preferred. A machine learning model is then trained to rank these flight plans based on their KPIs, taking into account the specific context (e.g., airspace user, city-pair, aircraft type) and other general features that allow for the establishment of a single model applicable to several city-pairs.

II. Background

The following subsections provide foundational background on flight planning, pre-tactical flight plan predictions, and ranking algorithms, respectively.

A. Flight planning

Flights departing from, arriving in, or overflying any country within the NM's area of operations are required to submit a flight plan. According to the International Civil Aviation Organisation (ICAO) [1], flight plans should be submitted at least 3 hours before the estimated off-block time (EOBT). Indeed, the NM's guidance document *All Together Now 2024* [2] highlights the importance of filing as early as possible, ideally no later than 4 hours before EOBT, to ensure efficient operations.

When airspace users determine a flight plan, they must balance several KPIs. While the most direct route (i.e., shortest distance) is often preferred, considerations such as weather, military activity, ATFCM regulations, and route charges may lead to alternative choices. For instance, avoiding regulated airspace might reduce delays but increase the distance. Similarly, to overfly cheaper airspace, airspace users may choose to fly longer distances. Ultimately, airspace users seek to maximise revenue by selecting flight routes that best align with their business objectives, which remain undisclosed.

In order to address these complicated trade-offs, many airlines use advanced flight planning software to generate optimal flight routes automatically, like Lido (Lufthansa Systems), Jeppesen JetPlanner or SITA Flight Folder, among others. These systems analyse various routes, predict fuel usage, estimate costs, and ensure compliance with all relevant airspace restrictions. Despite the reliance on automated systems, human oversight remains crucial in flight planning. Pilots, dispatchers, and other airline personnel review the automatically generated plans and may make changes based on real-time information, operational considerations, or specific airline policies.

B. Pre-tactical flight route predictions

In the NM system, two essential datasets – the forecast dataset and the operational dataset – play a critical role.

The forecast dataset is developed and refined during the pre-tactical phase, starting six days before the day of operations (D-6) and continuing until the day before (D-1).

This dataset contains only flights forecasted by PREDICT, which are constructed using a variety of data sources, including wind predictions, North Atlantic (NAT) traffic forecasts, airport slots, airline schedules, and traffic patterns from similar days in the past, typically from one week earlier[2]. The purpose of this dataset is to provide a detailed and accurate projection of traffic demand, which guides the preparation of regulations and other tactical updates. These plans are maintained within the forecast dataset until they are transferred to the operational dataset on D-1, around 16:00 UTC. Even after this transfer, the forecast dataset remains accessible until the end of the day of operations (D), though it no longer evolves after the handover.

The operational dataset, on the other hand, becomes active from D-1 and continues to be used throughout the day of operations. It is the primary dataset for real-time traffic management. Flight routes can be submitted by airspace users several days in advance, and these are integrated into the operational dataset approximately 24 hours before the EOBT. Unlike the forecast dataset, which is static after D-1, the operational dataset is dynamic, continuously updating as live flight plans are filed and adjusted. A key distinction between these two datasets lies in their purposes: the forecast dataset is exclusively for predictive modelling and planning, containing only anticipated flights without official flight plan identifiers, while the operational dataset includes all filed flight plans.

C. Ranking Algorithms

The goal of ranking algorithms is to order items – such as documents, products, web pages, or routes – according to their relevance to a query. They are widely used in applications like search engines and recommendation systems, where the effectiveness of the ranking directly impacts user satisfaction.

The performance of a ranking algorithm is evaluated using metrics that quantify how well the algorithm orders items, particularly how effectively it places the most relevant items at the top of the list.During the training, a loss function measures the difference between the predicted ranking generated by the algorithm and the ideal ranking. The primary objective during training is to minimise this difference, thereby improving the quality of the rankings. Pairwise and groupwise methods are two common approaches to constructing these loss functions, each focusing on different aspects of the ranking order.

In the pairwise approach, the loss function focuses on the relative ordering of pairs of items. The key idea is to ensure that for any two items, if one item is more relevant than the other, it should be ranked higher. Thus, this approach evaluates the ranking by comparing pairs of items, aiming to minimise the number of incorrectly ordered pairs. In contrast, the groupwise approach evaluates the ranking of an entire list of items as a whole, rather than just focusing on individual pairs. The goal is to optimise the overall order, considering all items together. This method often uses metrics that account for the position of each item in the list, giving more weight to the correct ranking of items at the top.

[2]For special events like holidays or strikes, the NM pre-tactical team may choose a different reference day to create a more accurate forecast.

III. Literature Review and Contribution

Improving pre-tactical traffic demand predictions has been a longstanding focus of research, with various machine learning models proposed over the past decade. For instance, [3] introduced machine learning models aimed at predicting flight plan choices during the pre-tactical phase. Their approach involved clustering historical flight routes for each city-pair and then predicting the most likely cluster using an individual model. Although effective, this method required separate models for each city-pair, leading to scalability challenges. The authors employed multinomial regression and decision tree models, framing the problem as a multi-class classification task. This modelling choice limited the ability to generalise because classification tasks require a fixed number of outputs.

Building on this foundation, [4], [5] developed machine learning models to predict pre-tactical flight plans and requested flight levels, demonstrating improved accuracy over the PREDICT system. However, scalability challenges persisted. Although they tested a broader range of models, the core approach remained the same: framing the city-pair route choice problem as a multi-class classification task.

Further extending this research, [6] introduced a model that predicts airline route preferences by considering factors such as fuel consumption, route charges, flight duration, and other KPIs, enabling adaptation to new, unobserved routes. This approach improved generalisation. However, the authors continued to rely on multi-class classification models, leaving their claims on generalisation across different city-pairs unverified, as the models' outputs remained fixed in number and meaning.

In our work, we propose a methodology to address several key challenges that have posed difficulties for previous methods in enhancing flight plan predictions during the pre-tactical phase. A major challenge has been the impracticality of building separate models for each city-pair, which complicates machine learning operations and limits the data available for training each model. To overcome this, we propose re-framing the route choice problem as a ranking problem, enabling the development of a universal model that can be applied across all city-pairs. The rationale, as supported by [6], is that the KPIs used in decision-making for flight planning, such as fuel usage, are generally consistent across different city-pairs.

Another significant challenge is that many existing models predict a specific flight plan without ensuring its validity on the day of operations, considering dynamic factors like restricted airspace, and other operational constraints. This can result in scenarios where none of the flight routes represented by the classifier's outputs are valid. To address this, our ranking models shifts the focus from predicting a specific flight plan to learning the score (or airspace user preference) assigned to a flight plan based on generic, city-pair-independent KPIs. This approach aligns more closely with the decision-making process of flight dispatchers, who evaluate these KPIs when selecting a flight plan. By treating flight plans as inputs rather than outputs, we ensure that all flight routes fed into our model are valid, as they are sourced from the NM.

Previous models have encountered significant challenges when attempting to incorporate wind. Creating a detailed wind map requires sophisticated approaches, often necessitating the use of complex neural network architectures. From an operational perspective, however, the primary concern for flight dispatchers is not the wind itself, but rather its impact on flight attributes such as flight duration and fuel consumption. Instead of directly modelling the wind across a flight route, it is more efficient to focus on the KPIs that inherently reflect the wind's influence. Thus, we argue that it is unnecessary to explicitly include wind data as a feature in the model. Instead, the model should directly capture the relationship between the flight route and these KPIs. By doing so, we can effectively account for the impact of wind without the need for the complex and computationally demanding task of wind data integration.

Finally, an essential aspect of training a machine learning model is ensuring that the inputs adequately explain the outputs. In the context of flight planning, this means that the inputs must accurately reflect the information evaluated by flight dispatchers and their decision-support tools when submitting a flight plan. This consideration has often been overlooked in previous research. To address this, we propose closely monitoring flight plan submissions and changes, using the information available at the time of submission. This approach increases the likelihood of modelling the same decision-making process as the flight dispatcher, thereby capturing the cause-and-effect relationship in the ranking model.

IV. Dataset

The dataset used to train the ranker was constructed using NM business-to-business (B2B) services, specifically utilising the publish/subscribe (P/S) and request/reply (R/R) systems. In the P/S system, clients subscribe to specific topics and receive notifications whenever new information is published. Conversely, the R/R service allows a client to make a request to a server, which then responds with the requested information.

For this study, we subscribed to the initial flight plan (IFP) and change (ICH) messages, ensuring immediate notification whenever a flight plan was submitted to NM or modified. From these messages, we extracted the flight *keys* – including the aerodromes of departure and destination, callsign, and EOBT – which uniquely identify each flight. Additionally, we retrieved the corresponding original ICAO route, detailed in field 15 of the flight plan. This field includes a sequence of waypoints and airways, along with altitude and speed instructions. For instance, a flight from Madrid to Paris might be filed with the route `N0480F360 DCT TERTO UN857 GOLDA UN858 SOPET UN872 BAMES`. In this example, `N0480F360` specifies a speed of 480 knots at flight level 360 (FL360); `DCT` indicates a direct route between waypoints; `UN857`, `UN858`, and `UN872` refer to airways; while `TERTO`, `GOLDA`, `SOPET`, and `BAMES` are the specific waypoints.

Then, we prepared a `RoutingAssistanceRequest`, which returns a list of valid routes for a given flight, along with computed KPIs such as fuel consumption, route charges, and flow-related what-if impacts (e.g., ATFCM delay).

It is worth noting that this service takes into account the most recent weather forecasts and utilises accurate trajectory prediction tools to compute fuel consumption and flight duration for a given route and aircraft type. This service can be invoked in two ways: (1) to evaluate proposals for an existing flight, where only the flight keys are required (as the flight is already in the system), or (2) to evaluate proposals for a new flight, where a complete flight plan must be specified, including the flight keys, aircraft type, and route. For the creation of the training dataset, we utilised the former approach, as the receipt of a message indicates that a flight with the specified flight keys is already present in the system.

The service, in its basic form, requires the flight keys of the existing flight, the number of proposed routes to generate, and a reference to determine the flight levels: `ORIGINAL` (using the flight levels from the original ICAO route), `HIGHEST` (using the highest reached flight level), and `LONGEST` (using the longest flown flight level). The proposed routes can be generated from city-pair statistics, including the set of routes flown in the last 12 AIRAC cycles, flights currently in the system, and a path generator that dynamically creates proposed routes by exploring the network of air routes, free-route areas, and direct segments. In the current implementation, we configures the service to only propose historical routes from city-pair statistics. The number of proposed routes is another configurable parameter of the service, set to 10 by default. However, depending on the reference flight level, fewer than the requested number of proposals may be returned.

Algorithm 1 outlines the steps followed to create the dataset using the aforementioned P/S and R/R NM B2B services.

Algorithm 1 Training dataset creation using NM B2B services

Require: Maximum number of proposed per flight N_{max}

$X \leftarrow \{\}$
Subscribe to `IFP` and `ICH` messages
while a message is received **do**
    `flight` $\leftarrow$ flight keys of the message
    `originalRoute` $\leftarrow$ Route of the message
    `proposedRoutes` $\leftarrow \{\}$
    for `referenceRequestFlightLevels` in {`ORIGINAL`, `LONGEST`, `HIGHEST`} **do**
        $n \leftarrow N_{\text{max}} - |$`proposedRoutes`$|$
        `proposedRoutes` $\leftarrow$ `proposedRoutes` $\cup$ `RoutingAssistanceRequest(flight, referenceRequestFlightLevel,`n`)`
    end for
    $X \leftarrow X \cup \{($`originalRoute, proposedRoutes`$)\}$
end while

In this dataset, each observation corresponds to an IFP or ICH message and includes the original ICAO route along with up to N_{max} proposed ICAO routes. For each route, whether original or proposed, the `RoutingAssistanceRequest` provides the KPIs and flow impacts, which serve as features for the ranker discussed in the following section. The default parameter of 10 proposed routes was selected.

We started data collection on June 17th, 2024, targeting flights departing from or arriving at the top-50 busiest airports in Europe, covering up to 2.5K city pairs[3]. The results in this paper are based on data collected up until the 13th of September, 2024, comprising around 1M observations.

The advantages of using NM B2B services to generate the training dataset, compared to existing datasets based primarily on EUROCONTROL's Data Demand Repository (DDR), are significant. First, our approach leverages NM's consolidated trajectory prediction tools to extract KPIs directly from routes. By making requests immediately after receiving the messages, we maximise the likelihood of using the exact same weather forecasts seen by the emitter. This is crucial, as we anticipate that flight planning tools utilise comparable tools and weather data. Consequently, the KPIs available to flight planners or dispatchers just before issuing an IFP or CHG message should closely match those in our dataset. This accuracy extends to flow-impact metrics such as ATFCM delay and regulations.

Second, the alternatives generated by NM are valid at the exact moment of the event. Unlike previous approaches that might include alternatives that were not selected just because they were invalid at the time, we use NM-compliant routes.

Third, our model is independent of specific environment data (such as routes, waypoints, and sectors), relying solely on the KPIs of potential routes. This ensures that our model remains valid even if the environment changes – such as the introduction of new waypoints or airways. In such cases, we simply need to obtain updated route proposals from NM along with their corresponding KPIs to generate an accurate ranking.

Lastly, our approach allows for rapid integration into operational systems. Specifically, we only need access to the `RoutingAssistanceRequest` to obtain the available routes and corresponding KPIs and provide NM with the predicted ranking of valid routes for each flight.

V. Model

This section presents the ranking model proposed in this paper. Specifically, Section V-A outlines the input features utilised in the model, Section V-B details the output generated by the model, and Section V-C provides a comprehensive explanation of the implementation of the ranker. It is important to note that this project is ongoing, and the designs presented here are preliminary. Future work will explore additional input features, models (e.g., neural networks), and/or loss functions.

A. Input features (predictors)

Feature engineering was applied to the dataset presented in the preceding section to compute the input features (i.e., predictors) for each route used by the model to predict the corresponding score and generate the rankings of each observation (or message). The features are grouped into various sets and tagged according to their respective topics. A summary of the sets and associated features is provided in Table I.

[3]NM B2B services requires dedicated certificates. Furthermore, the `RoutingAssistanceRequest` is a computationally expensive service, capped at 30 requests/min. One certificate may be adequate to cover hundreds of city pairs, but more certificates would be needed for the entire network.

TABLE I. INPUT FEATURES (PREDICTORS) OF EACH ROUTE, CATEGORISED BY TAG AND TYPE.

Tag	Name	Type
Flight attributes	Aircraft type (identifier) City-pair (identifier) Airline (identifier)	Categorical
KPIs of the route	Duration (min) Length (NM) Fuel consumption (kg) Route charges (Euros)	Numerical
Flow-related (ATFCM) impact	Delay (min) Number of regulations (#) Protected location of MPR (identifier) Type of location of MPR (identifier)	Numerical Categorical
Calendar	Hour Day of week Month. This feature was not used in the experiment.	Categorical
Convection risk from the most recent CBCF	Medium risk (%) High risk (%) Very high risk (%)	Numerical
Distance w.r.t the lateral and vertical profiles	Mean lateral and vertical embedding distance by: ■ City-pair and airline ■ City-pair, airline and aircraft type ■ Callsign	Numerical
Similarity w.r.t the set of waypoints	Mean precision and recall similarity by: ■ City-pair and airline ■ City-pair, airline and aircraft type ■ Callsign	Numerical

The first set of features is designed to provide context for the model, specifically including the airline, the city-pair, and the aircraft type. It is important to note that we also experimented with splitting the city-pair feature into separate fields for the departure and destination aerodromes. However, this approach degraded the quality of the model.

The second set of features encompasses KPIs related to the route, including duration, length, fuel consumption, and route charges. It is important to note that both duration and fuel consumption implicitly account for the most recent wind forecast available before the message was submitted. We expect this set of features to be highly relevant to the model.

The third set of features provides information about the flow-related ATFCM impact on the flight. This includes the ATFCM delay, the number of regulations affecting the flight, and details about the most penalising regulation (MPR), such as its protected location and type (e.g., airspace, aerodrome). The rationale for including these features is that some airlines may choose to avoid regulated airspace if the delay is high.

The fourth set of features includes calendar information such as the hour of the day, the day of the week, and the month of the year. This data is apparently important because airspace users may have varying preferences and operational patterns depending on the season or time of day. Due to the relatively small size of the dataset at the moment of writing this paper, however, the month feature was omitted.

The fifth set focuses on convective weather, specifically the Cross-Border Convection Forecast (CBCF) generated by EUMETNET (the European Meteorological Network). This forecast is a collaborative effort that supplies information about convective weather across European airspace to NM and participating air navigation service providers. During 2024, CBCFs were issued twice daily, at 7AM and 10PM, and included polygons that represent different levels of convection risk. These forecasts are valid for the following day, covering the period from 6AM to 9PM in 3-hour intervals. These features indicate the percentage of route affected by each risk level. For instance, a medium risk with value 0.9 indicates that 90% of the route is crossing medium risk polygons.

The sixth feature set represents the distance between a given route and all routes filed during the current and previous AIRAC. This set is still under development, and we are exploring the benefit of including complementary metrics, such as Fréchet distance. In the current implementation, each route's sequence of waypoints (i.e., the lateral profile) is transformed into an embedding – a fixed-size vector of numerical values that captures the route's shape. To foster generalisation, the lateral embedding is designed to be independent of spatial position, direction, and length.

The transformation process involves several steps: First, the route is rotated so that the direct path from the origin to the destination serves as the reference axis. Then, it is translated so that the origin airport aligns with the origin of the coordinate system. Finally, the route is scaled according to the great circle distance. After these transformations, the route is interpolated at n equidistant points, and the resulting x and y coordinates form the embedding. This process is illustrated in Fig. 1.

A similar yet simpler process is applied to the vertical profile (i.e., the sequence of flight levels). In this case, the altitude is interpolated at n equidistant points along the route.

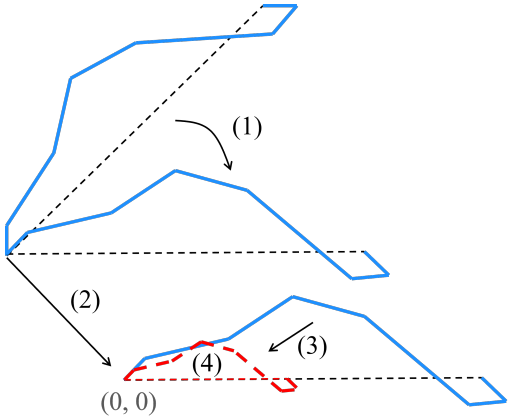

Figure 1. The steps to generate a 2D route embedding: (1) rotate, (2) translate, (3) scale, and (4) interpolate.

In the current implementation, $n = 15$ was chosen to generate both lateral and vertical embeddings. Each distance metric is computed as the average Euclidean distance between the lateral or vertical embedding of the given route and those of last filed flight plans during the current and previous AIRACs, for the same city-pair by the same airline, airline-aircraft type combination, and callsign. As a result, this approach yields a total of six distance-based features: three based on lateral embeddings and three based on vertical embedding.

The final set of features is inspired by classification similarity metrics. The similarity between two routes – current and historical – is measured in terms of precision and recall:

$$\text{precision} = \frac{\text{\# of common waypoints}}{\text{\# of waypoints in the current route}},$$
$$\text{recall} = \frac{\text{\# of common waypoints}}{\text{\# of waypoints in the historical route}}.$$

For example, consider the set of waypoints of an historical route: `TERTO GOLDA SOPET BAMES`. If the set of waypoints of the current route were `TERTO TEDKA BAMES`, the precision and recall would be $2/3$ and $2/4$, respectively. Similarity metrics consists of the average precision and recall between the route and those of last filed flight plans during the current and previous AIRACs for the same city-pair, by the same airline, airline-aircraft type combination, and callsign. This approach generates a total of six similarity-based features: three for recall and three for precision.

B. Target (output)

We unfortunately lack explicit preference scores for the routes. Instead, we have only the information about which route was selected (the original) and which were proposed alternatives. We use the selection itself as a proxy for score. For each observation (i.e., message), the selected route is labelled as the preferred option (1), while all alternative proposed routes are labelled as less preferred (0). In other words, each observation contains one route with a score of 1, and the remaining routes are assigned a score of 0.

C. Implementation

In this study, we employed the CatBoost Ranker as our base model [7], chosen for several compelling reasons. First, CatBoost is particularly well-suited for datasets with high-cardinality categorical features, such as those found in our data, which includes city-pairs, aircraft types, and other categorical attributes. Its ability to handle such features without extensive pre-processing is a significant advantage.

Furthermore, CatBoost is inherently robust against overfitting, which is crucial for ensuring the generalisation capabilities of the model, especially when working with complex and dynamic data like flight routes. Furthermore, the relatively low number of major hyper-parameters that require tuning simplifies the model optimisation process.

Finally, CatBoost stands out as one of the most advanced tree-based models for ranking tasks, offering a wide range of loss functions that provide flexibility for various applications [8], [9]. In the current implementation, the model is trained to minimise the PairLogit loss, a widely used objective for ranking tasks. For each pair of items (i, j), the model predicts scores s_i and s_j, which are used to estimate the probability that item i should be ranked higher than item j. The probability is modelled using the logistic function:

$$P(i > j) = \frac{1}{1 + e^{-(s_i - s_j)}}.$$

The PairLogit loss function then minimises the negative log-likelihood of these pairwise comparisons, defined as:

$$\mathcal{L}_{\text{PairLogit}} = -\sum_{(i,j)} \log\left(P(i > j)\right) = \sum_{(i,j)} \log\left(1 + e^{-(s_i - s_j)}\right).$$

However, CatBoost is not without its limitations. As part of future work, we plan to compare our current CatBoost-based model with a neural network approach. Neural networks offer greater flexibility, particularly when it comes to defining more complex or customised loss functions. Additionally, a neural network model might enable a more seamless and transparent integration of embeddings and textual route features.

VI. Results

VII. Performance Evaluation

This section details the performance of the proposed ranker, which was trained on data collected from June 17th, 2024, to August 31st, 2024. The model's performance is assessed using a test set that spans the remaining days until September 13th, 2024. This test set, referred to as the *tactical test set*, includes observations similar to those used during training, where the user route is considered the ground truth and alternatives were generated with the `RoutingAssistanceRequest`.

Additionally, we compare the performance of our model with that of PREDICT during the pre-tactical phase using a dedicated test set. This test set was generated by initially calling PREDICT at midnight on D-1. For each predicted flight, the `RoutingAssistanceRequest` service was invoked to obtain 10 alternative proposed routes.

TABLE II. AVERAGE DISTANCE, SIMILARITY AND CLUSTER MATCH METRICS IN THE TACTICAL (T) AND PRE-TACTICAL (P) TEST SETS.

Metric	Lateral embedding distance		Vertical embedding distance		Precision		Recall		Lateral cluster match		Vertical cluster match	
Test set	T	P	T	P	T	P	T	P	T	P	T	P
Fastest	0.13	0.06	14.30	13.31	0.52	0.53	0.50	0.51	0.65	0.69	0.85	0.31
Shortest	0.09	0.06	8.37	11.44	0.58	0.53	0.57	0.50	0.71	0.68	0.89	0.32
Lowest fuel	0.09	0.07	19.03	14.03	0.57	0.51	0.55	0.48	0.69	0.65	0.80	0.30
Lowest charges	0.11	0.11	5.37	11.20	0.58	0.47	0.56	0.45	0.71	0.60	0.92	0.31
PREDICT	-	0.05	-	7.57	-	0.58	-	0.55	-	0.76	-	0.32
Highest Rank	0.04	0.04	6.06	10.33	0.70	0.59	0.68	0.56	0.87	0.77	0.93	0.32

For each observation in the pre-tactical test set, the model is tasked with ranking up to 11 routes: the route predicted by PREDICT and up to 10 alternative routes. The last filed flight plan is used as the ground truth for the PREDICT test set. It is worth noting that this evaluation may not be entirely fair and exhaustive, as PREDICT is designed to forecast the first filed flight plan rather than the last. Consequently, the results obtained from the PREDICT test set should be interpreted with caution. Extensions of this work will involve an evaluation using the first filed flight plans as the ground truth.

In the pre-tactical test set, it is possible that none of the up to 11 candidate routes may perfectly match the ground truth, laterally and/or vertically. To address this, we applied the DBSCAN algorithm to independently cluster the vertical and lateral embeddings of the routes for each observation. We then assess whether a predicted route belongs to the same lateral or vertical cluster as the ground truth. Additionally, we will report the average lateral and vertical embedding distances between the predicted route and the ground truth, as well as the average precision and recall based on the waypoint sets.

In addition to comparing with PREDICT, we have defined several dummy baselines for comparison purposes. These baselines select the route with the lowest value for each KPI as the predicted route, providing a reference point for evaluation.

Since the dataset covers only three months, the experiment presented herein focuses on the top 250 most frequently used city-pairs for both training and evaluation, regardless of their performance quality. This approach ensures that the model encounters a diverse range of scenarios for each city-pair during training, which is crucial for achieving generalisation.

A. Aggregated performance metrics

Table II provides a comparative analysis of various route prediction models, including the proposed ranker, PREDICT, and several baseline models, evaluated across multiple metrics in both tactical and pre-tactical test sets. It is worth noting that the number of flights in these two test sets may differ, which makes a direct comparison difficult if not impossible.

The baseline models based on individual KPIs consistently underperform compared to both PREDICT and the ranker. These baseline models generally exhibit higher lateral and vertical embedding distances, lower precision and recall scores, and poorer cluster matching rates, reflecting their limitations in accurately predicting the users' preferred routes.

The proposed ranker demonstrates modest but clear improvements over PREDICT in several important metrics. Specifically, it achieves lower lateral embedding distances and higher precision and recall values, indicating that it more accurately predicts routes based on waypoint overlap. However, one area where the ranker falls short is in the vertical embedding distance, where it shows a significantly higher average distance than PREDICT. Despite this, the improvements in lateral performance suggest that the ranker provides meaningful enhancements in some aspects of route prediction, though the gains are incremental at this stage.

A notable takeaway from Table II is the pronounced difficulty all models face in predicting the vertical profile of routes, particularly in the pre-tactical test set. Both the ranker and PREDICT struggle with vertical distance metrics, which suggests that the vertical component of route prediction is a persistent challenge that requires further attention. Note, however, that by vertical profile we are not referring solely to the precise sequence of flight levels, but also to the specific locations where step climbs are performed, which naturally adds an additional layer of complexity to the prediction task.

B. Model explainability

Figure 2 displays the Shapley values in the observations of the test set. In short, the Shapley values represent the average marginal contribution of each feature in the model across all possible combinations of features. This means that the Shapley value for a given feature is the average difference in the model's prediction when that feature is included versus when it is not, considering all possible subsets of the other features. This provides a measure of the importance of each feature in the model's predictions. For more information on Shapley values, please see [10] and the references therein.

In this kind of figure, the y-axis indicates the name of the features, in order of mean absolute Shapley value from the top to the bottom. Each dot in the x-axis shows the Shapley value of the associated feature on the prediction for one observation, and the colour indicates the magnitude of that feature.

Figure 2 confirms that the model has successfully learned patterns that align with expectations, reflecting an understanding of route prediction factors that would resonate with human judgement. Notably, city-pair and fuel consumption emerge as the most important features, with higher fuel consumption consistently leading to lower predicted scores. Several route similarity and distance metrics also rank high in importance, where greater similarity boosts prediction scores, while higher distance metrics lead to lower scores, as expected.

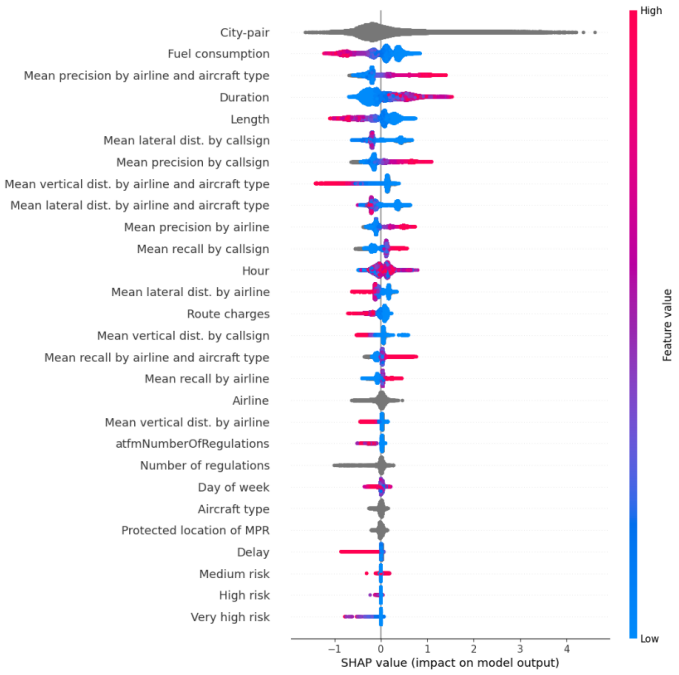

Figure 2. Shapley values in the routes of the tactical test set.

ATFCM and weather-related features appear towards the bottom of the ranking. This, however, does not imply that they are unimportant; rather, many flights are not impacted by convective areas or ATFCM regulations. Since this figure shows only the average Shapely value, their actual relevance may be underrepresented. A more detailed analysis focused on flights affected by ATFCM regulations or adverse weather might reveal an increase in the importance of these factors.

VIII. Conclusion

This paper addresses a gap in the literature on route prediction by identifying limitations in existing methods – namely model maintenance, generalisation and validity of the proposed routes – and proposing a learning-to-rank approach. While the primary objective of this approach is to enhance pre-tactical flight predictions, it has potential applications in the tactical phase as well. For instance, when ATFCM regulations are activated, the ranker could be asked to rank the current and alternative (valid) routes considering ATFCM delays in order to identify possible route changes before they happen.

While the primary focus of this paper is on methodological advancements, we also present preliminary results from initial experiments. Although these results do not yet demonstrate substantial improvements over PREDICT, they highlight the feasibility and solidity of our approach. We believe that with further development in model implementation, input feature refinement, and the use of a larger dataset, there is significant potential for improvement. Instead of detailing the extensive list of possible enhancements in this section, we have discussed various ideas throughout the paper. This approach provides a clear perspective and direct links on how the model will be improved in the near future.

Acknowledgement

This project has received funding from the SESAR 3 Joint Undertaking (JU) under grant agreement No 101114715. The JU receives support from the European Union's Horizon Europe research and innovation programme and the SESAR 3 JU members other than the Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or SESAR 3 Joint Undertaking. Neither the European Union nor the granting authority can be held responsible for them.

References

[1] International Civil Aviation Organization, "Eur regional supplementary procedures (supps) (doc 7030)," 2018.
[2] Network Manager, "All Together Now 2024," 2024.
[3] R. Marcos, O. García-Cantú, and R. Herranz, "A machine learning approach to air traffic route choice modelling," 2018.
[4] M. Mateos, I. Martín, P. García, R. Herranz, O. G. Cantú-Ros, and X. Prats, "Full-scale pre-tactical route prediction: Machine learning to increase pre-tactical demand forecast accuracy," in *Proceedings of the 9th International Conference on Research in Air Transportation (ICRAT)*, (Online), 2020.
[5] M. Mateos, I. Martín, R. Herranz, O. G. Cantú-Ros, and X. Prats, "Predicting requested flight levels with machine learning," in *Proceedings of the 10th SESAR Innovation Days*, (Online), 2020.
[6] M. Mateos, I. Martín, R. Alcolea, R. Herranz, O. G. Cantú-Ros, and X. Prats, "Unveiling airline preferences for pre-tactical route forecast through machine learning," in *Proceedings of the 11th SESAR Innovation Days*, (Online), 2021.
[7] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "Catboost: unbiased boosting with categorical features," *arXiv preprint arXiv:1706.09516*, 2018.
[8] A. Gulin, I. Kuralenok, and D. Pavlov, "Winning the transfer learning track of yahoo!'s learning to rank challenge with yetirank," in *JMLR: Workshop and Conference Proceedings*, (Haifa, Israel), pp. 63–76, 2011.
[9] I. Lyzhin, A. Ustimenko, A. Gulin, and L. Prokhorenkova, "Which tricks are important for learning to rank?," 2023.
[10] S. M. Lundberg, G. Erion, H. Chen, *et al.*, "From local explanations to global understanding with explainable AI for trees," *Nature machine intelligence*, vol. 2, pp. 56–67, 2020.