

Article

Probabilistic Prediction of Separation Buffer to Compensate for the Closing Effect on Final Approach [†]

Stanley Förster * , Michael Schultz  and Hartmut Fricke 

Chair of Air Transport Technology and Logistics, Institute of Logistics and Aviation, TU Dresden, 01062 Dresden, Germany; michael.schultz@tu-dresden.de (M.S.); hartmut.fricke@tu-dresden.de (H.F.)

* Correspondence: stanley.foerster@tu-dresden.de

[†] This paper is an extended version of our paper published in Proceedings of the 9th International Conference for Research in Air Transportation (ICRAT), 2020.

Abstract: The air traffic is mainly divided into en-route flight segments, arrival and departure segments inside the terminal maneuvering area, and ground operations at the airport. To support utilizing available capacity more efficiently, in our contribution we focus on the prediction of arrival procedures, in particular, the time-to-fly from the turn onto the final approach course to the threshold. The predictions are then used to determine advice for the controller regarding time-to-lose or time-to-gain for optimizing the separation within a sequence of aircraft. Most prediction methods developed so far provide only a point estimate for the time-to-fly. Complementary, we see the need to further account for the uncertain nature of aircraft movement based on a probabilistic prediction approach. This becomes very important in cases where the air traffic system is operated at its limits to prevent safety-critical incidents, e.g., separation infringements due to very tight separation. Our approach is based on the Quantile Regression Forest technique that can provide a measure of uncertainty of the prediction not only in form of a prediction interval but also by generating a probability distribution over the dependent variable. While the data preparation, model training, and tuning steps are identical to classic Random Forest methods, in the prediction phase, Quantile Regression Forests provide a quantile function to express the uncertainty of the prediction. After developing the model, we further investigate the interpretation of the results and provide a way for deriving advice to the controller from it. With this contribution, there is now a tool available that allows a more sophisticated prediction of time-to-fly, depending on the specific needs of the use case and which helps to separate arriving aircraft more efficiently.

Keywords: probabilistic prediction; final approach; time-to-fly; random forest; quantile regression



Citation: Förster, S.; Schultz, M.; Fricke, H. Probabilistic Prediction of Separation Buffer to Compensate for the Closing Effect on Final Approach. *Aerospace* **2021**, *8*, 29. <https://doi.org/10.3390/aerospace8020029>

Academic Editor: Felipe A. C. Viana

Received: 11 December 2020

Accepted: 21 January 2021

Published: 26 January 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Airport performance is mainly driven by the efficient use of limited resources along the aircraft trajectories. In this context, the airport capacity is defined as the maximum amount of aircraft movements in a given period of time and provides a defined quality of service (delay). One of the most limited components of the airport system is the runway (system). Air traffic controllers have to make sure that the runway utilization is maximized dependent on the traffic mix, environmental constraints (e.g., weather or noise), operational modes (arrivals and departures) ensuring safe operations by considering separation standards (see [1–4]). Capacity efficient operations are becoming more important since around 1.5 million flights (accounting for 8% of the demand) will not be accommodated in 2040 [5]. Thus, the capacity gap is equivalent to eight fully-used runways, spread over 17 different states in the European Civil Aviation Conference (ECAC) region, where 21 airports lack capacity. The higher amount of air traffic movements will also lead to increased traffic complexity, especially in high-density areas like the terminal maneuvering area (TMA).

The complexity in this area further increases with future, more diverse separation standards, like re-categorisation of ICAO wake turbulence separation (RECAT-EU) [6],

time-based separation [7], or pairwise separation [8], which all lead to enhanced combinatorial complexity due to an increased number of different aircraft classes to consider in separation management.

Additionally, independent of the separation scheme applied, the closing effect plays a dominant role in the development of inter-aircraft distance along with the final approach. After passing the final approach fix (FAF), aircraft follow individual speed profiles while decelerating to landing speed. Typically, the leading aircraft, which is much closer to the threshold, is slower than the succeeding aircraft that catches up consequently. Hence, aircraft need to be separated much wider at the beginning of the final approach leg by adding some buffer on top of the required separation, which is consumed by the closing effect. Ideally, the buffer is fully depleted when the preceding aircraft crosses the threshold and the distance to the succeeding aircraft equals the required minimum separation. However, since speed profiles are not known exactly in advance, especially due to uncertainty induced by weather effects, controllers usually apply a buffer that is larger than required by the closing effect. Therefore, air traffic controllers require enhanced support tools that help to overcome those challenges by providing hints on how to separate aircraft most efficiently. Having available an aid on the radar screen that visualizes the specific separation that is to be applied to a certain pair of aircraft, controllers do not need to remember a large matrix of combinations in the first place. Additionally, the support tool can be enabled to consider the compression effect and suggests an appropriate buffer. This however requires knowledge about the aircraft's speed profiles along the extended runway centerline and upstream to anticipate the compression effect so that the support tool's advisories won't lead to loss of separation when followed by the controller.

Naturally, the air transportation system is both a competitive and collaborative environment, where stakeholders have to optimize their economic benefits considering various restrictions. By giving airport stakeholders access to data from different sources, airports will be able to make more accurate predictions about their operational progress in the next period. In this context, data-driven predictions and machine learning techniques could enhance airport processes to show hidden interdependencies in the complex airport system. The speed and extent, with which data is shared, have massively increased over the last years aiming at a resilient air transportation system [9]. The obligation to equip large aircraft (i.e., maximum take-off mass (MTOM) > 5.7 t and max. indicated air speed (IAS) > 250 kn) with a Automatic Dependent Surveillance-Broadcast (ADS-B) transponder started in Europe in June 2020 [10,11] and in the US almost all controlled air spaces are reserved only for appropriately equipped aircraft since January 2020 [12]. It is expected that current surveillance systems will be extended with ADS-B and future ground stations will be fully based on this technology, which is significantly cheaper to install and to operate. Due to the simple requirements on the receivers, ADS-B has contributed to the development of online services that display the current air traffic in real-time with worldwide receiver networks (depending on the local coverage), such as The OpenSky Network (opensky-network.org) [13], Flightradar24 (flightradar24.com), or FlightAware (flightaware.com). This technology also offers a solution for monitoring remote areas and flights over the oceans with space-based ADS-B [14]. Furthermore, the equipment of ground vehicles at aerodromes should enable more comprehensive monitoring of the traffic situation on the corresponding movement areas at the apron [15].

1.1. Status Quo

For modeling the aircraft's speed profile, we follow the idea of data-driven methods over physics-based models. While the latter is always a simplified representation of the real, complex system neglecting contributing factors that are hard to model in full detail, data obtained from measuring the behavior of the system implicitly comprises inherent effects influencing the system. Even though data-driven methods have experienced great attention recently and are expected to provide much better results, especially with the

availability of big amounts of data in mind, there are now other challenges to be solved when employing those methods.

One effect is the so-called data set shift or distribution shift [16], which describes the effect that training and test data set come from different distributions, which might render a model useless when it is trained on one distribution and tested or even employed in production on another one. This usually occurs when the data sets are acquired at different times or locations. For instance, a model trained on data acquired during summer may not be applied during the winter season without thorough testing or adaption.

Another issue that is receiving increasing attention in the field of machine learning is the lack of understanding about the inner workings of the models being developed [17]. This is also highly related to trust and certification issues in safety-critical systems like air transportation. Most of the models in the field are usually considered black-box models, i.e., it is not straightforward to grasp what relationships in the data a model identified during the learning process and how those influence the prediction.

In the context of aviation, several methods from the field of artificial intelligence are used to cluster aircraft trajectories [18–21], detect anomalies [22–25] and predict aircraft trajectories [26–28], development of dynamic airspace designs [29,30], analyse runway and apron operations [24,31,32], determine airport performance including the impact of local weather events [33], and for aircraft turnaround operations [34]. More initiatives to leverage ADS-B open data to improve the state of the art are already commonplace, esp. in the field of aircraft modeling [35,36]. An ADS-B based milestone concept is already developed [37] by the authors to enable the data-driven performance management even for small/medium-sized airports (see Figure 1). Therefore, the incoming ADS-B messages and the data contained therein are processed accordingly and used to monitor arrival flows, runway occupancy, aircraft ground trajectories, taxiing times, and airport capacity utilization. This first step in monitoring airport operations offers operators improved situational awareness and needs now to be enhanced with forecasting capabilities.



Figure 1. ADS-B-based milestone approach for data-driven operational awareness and framework for upcoming prediction tasks using Zurich Airport (LSZH) as an example [37]. Milestones are for instance actual in-block time (AIBT), aircraft ready time (ARDT), actual off-block time (AOBT), or actual take-off time (ATOT).

Trajectory prediction is a very prominent field that provides a lot of use cases for machine learning in aviation. While historically being dominated by model-based approaches, e.g., [38,39], data-based methods are gaining growing attention. One criterion to distinguish the prediction methods is whether they are intended to predict the time of arrival [40,41] or a full trajectory, spatially and temporally [42–44]. The accuracy of the predictions is quantified solely on the comparison between prediction and ground truth during the model testing phase. There is no measure of uncertainty for a single prediction, like the output of a prediction interval would provide.

Most studies that assign uncertainties to predictions can be found in the context of arrival and flow management for planning and scheduling inbound aircraft. Usually, those applications comprise much longer time horizons (from 20–30 min up to several hours before arrival) and are used for building up sequences rather than fine-grained separation management. In the context of predicting arrival times, efforts have been made to also predict the probability to which the actual arrival time may deviate from the estimate [45,46], as well as making this information available in the decision finding process of controllers [47]. Furthermore, methods have been developed that aim to infer the accuracy of the prediction by using a generative model that creates multiple possible trajectories and infers a probability distribution from this set [48,49].

Research so far is mostly of methodological nature providing little to no application or use case for the prediction models. However, there is some work available that focuses on the very specific application of trajectory and time-to-fly prediction in the context of separation management during the approach. In [50] the authors seek to predict optimal separation for the following aircraft when the leader is at 4DME (Distance Measuring Equipment, which corresponds to 4 NM ahead of the threshold), considering aircraft speed profiles, wind conditions, and the resulting closing effect. The speed profile and time-to-fly values are formulated as functions of aircraft type and wind band. Finally, the authors derive fixed values for additional separation buffer to apply, depending on wind speed and RECAT-EU wake turbulence category, which reflects average behavior. The follow-up study evaluates different machine learning techniques to improve the prediction of speed profiles and time-to-fly [51]. The authors evaluated six regression techniques, as well as four neural network techniques. However, predictions are again made per aircraft-type and wind-band combination and not for individual aircraft. The authors of [52] use a similar data set and concept for predicting mean speed profile and time-to-fly per cluster of aircraft-type and wind-band. Contemplating previous studies, authors assume that a normal distribution of the actual values and in addition to the mean, track the standard deviation, as well. The following studies, where predicting those values for the final approach segment is presented, serve as an enabler for Eurocontrol's Leading Optimised Runway Delivery (LORD) Tool [53], which is a controller assistance tool that provides an additional indicator on the radar screen that represents the proposed position for an aircraft behind its predecessor so that when the leading aircraft passes the threshold, the distance between aircraft is as close as possible at the minimum required separation.

Our work contributes to the field of machine learning in trajectory prediction describing an approach that combines time-to-fly prediction and uncertainty estimation by allowing us to quantify the prediction interval and even give a full probability function over the remaining time-to-fly. Furthermore, we additionally provide a specific use case making use of the presented model in a real ATC environment.

1.2. Focus and Structure of The Document

In the presented paper, we focus on the area around the final approach, including the upstream part of the turn onto instrument landing system (ILS) (instrument landing system, provide horizontal and vertical flight guidance) course. We develop a methodology to predict time-to-fly not only as a point estimate as in the work presented before but also considering the uncertainty inherent to predictions of the future. This will enable the assessment of possible impacts on safety and capacity, which is subsequently employed to find a trade-off for optimal separation. Controllers required their support tools to propose only actions that will not compromise safety in the first place. Therefore, a prediction method is required that allows deriving information about the uncertainty of the prediction, the prediction interval.

In the following Section 2, we first present the data our analysis is based on, including a description of the data preparation steps. Section 3 describes the method for probabilistic prediction of time-to-fly and provides an overview of the model assessment measures. Afterward, in Section 4 the model implementation, concrete training procedure, as well

as predictions and assessment, is provided. Complementary, we briefly present the use case of determining the optimal initial separation buffer required to compensate for the compression effect along with the final approach. Finally, the paper ends with a discussion and outlook in Section 5.

2. Data Preparation

2.1. Data Description

The flight track data set comprised 6 months of arrivals and departures to/from a major European airport serving over 400,000 flights per year. All flights were recorded within a 40 NM radius around the airport reference point (cf. Arrival Sequencing and Metering Area (ASMA) [54]) and are represented by a metadata record as well as a list of trajectory points. A flight's metadata contains information about

- Flight identifier
- Date and time of arrival or departure (i.e., touch-down/take-off)
- Origin and destination airport
- Runway identifier
- Call sign
- Aircraft type

For each flight, the trajectory was represented by a series of records with a 4-second update interval. Each record contained information about the aircraft's position (easting, northing, altitude) and velocity (ground speed).

Furthermore, meteorological aerodrome report (METAR) data was acquired to provide weather information for the full recording period covered by our data set [55]. Each METAR record was composed of a timestamp and the actual report. A report was issued every half an hour at xx:20 and xx:50, respectively. From each report, we extracted information about wind (speed and direction), pressure, and temperature (cf. [56]).

2.2. Preprocessing

2.2.1. Data Selection

In the preprocessing phase, we prepared the data for further analysis and the use in model training and testing phase. This comprised data set selection, merging of different data sets into a common data frame, as well as data imputation, filtering, and some additional pre-calculations.

The data used for this study were clustered by aircraft type and landing runway. The proposed methodology was outlined using an exemplary cluster that contained about 5000 arrivals performed within six months. This way, the model training process was more tailored towards specific characteristics and the predictions were more streamlined, instead of having to deal with more diverse data, which is suspected to have a decreasing effect on model accuracy.

For weather data, no pre-selection was made, the full 6 month period of nearly 9000 records was included in the base data set. All records that were not required, for instance, due to night flying restrictions and the resulting absence of traffic data during that time, were automatically abandoned in the following data merging step.

Finally, we included airport-specific information, especially regarding the landing runway. The most important feature here was the location of the threshold, against which distance and time-to-fly were measured.

2.2.2. Data Merging

After all, data were obtained and included in the base data set, the flight tracks were merged with airport data. This related every point of the trajectory with data about the location of the threshold to allow computation of distance to go. Subsequently, for each record of the trajectory, atmospheric data from the METAR data set was attached. Since METAR reports were issued every half an hour, every point of the trajectories that was recorded in between those times is related to the last, most recently available weather

data record. We included both, the METAR report time, as well as the track recording time in the feature set. This allowed the model to conclude the time difference during the training phase. A schematic view of all the data sets and how they are merged is depicted in Figure 2.

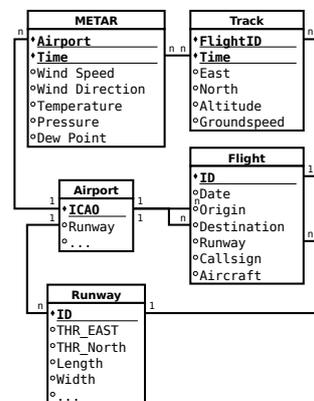


Figure 2. Schematic view on data sets and their relationship.

2.2.3. Imputation

During the data imputation phase, records with missing data were identified and, if possible, values were inferred. Flight tracks were always fully recorded, therefore, no missing data were found in the data set. Contrarily, several effects led to incomplete records in the METAR data set.

First, a METAR record may have been missing. This was the case for 15 out of the expected 8928 records, which corresponded to 0.17%. Another 10 records contained the string 'NIL', which indicates that no METAR record was issued for this time. Furthermore, even if available, not every METAR report contained information about all the weather features that were required for the analysis. For instance, an update may concern runway parameters exclusively (e.g., visibility) but not wind parameters or temperature.

For all those cases, data imputation was required to fill missing values in the data set. In the first step, we inserted 'NIL' reports where a METAR report was completely missing. This led to a fully populated data set. However, this did not provide additional information about weather properties like wind or temperature. Therefore, in a second step, all records with missing information about those properties were selected. This comprised 32 records where temperature, dew point, and pressure were missing, as well as 25 records where wind speed was missing. A numerical value for the direction of the wind was missing in 770 records. This is not only because of missing METAR reports, but there existed a special value 'VRB' for slow winds (up to 9 knots in this case) from variable directions. In any case, the longest consecutive sequence of missing weather data was of length 7, which equals a 4 h period.

For filling up the missing weather data, we used a linear interpolation approach. While this can be done straight forward for features like temperature, dew point, and pressure, wind speed and direction need special handling. Since wind direction is a cyclical feature, a simple linear interpolation was not possible. Instead, the wind was treated as a vector that reflected speed (length) and direction, and which is decomposed into its Cartesian components, usually named u and v . Those components are then interpolated independently of each other and the results are transformed back into two values for speed and direction.

2.2.4. Filtering

To reflect the scope of the time-to-fly prediction that was described above, the flight tracks were truncated accordingly. Therefore, a rectangular region around the final approach path was defined that covered the area where aircraft were usually cleared to

intercept ILS. This comprised the extended runway centerline within the 40 NM radius of data coverage and extends orthogonally to this until just before the downwind leg to not include that part of the track. Figure 3 depicts this area schematically.

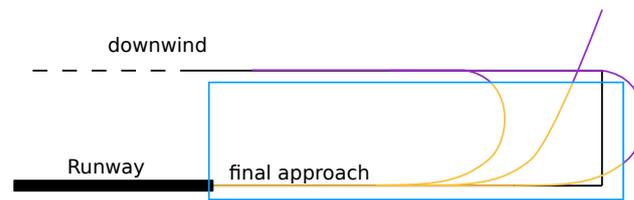


Figure 3. Area of interest, blue rectangle depicts the region to which flight tracks are truncated. The orange part of the flight tracks are part of the data set, the purple parts are discarded. (not to scale).

As soon as an aircraft entered this region, a prediction of its time-to-fly was possible. Flights that left the area after entering it, for instance, due to overshooting the localizer (LOC) intercept, were considered from the first entry on, provided they land eventually.

Additional filter criteria were altitude and track to exclude flights that crossed the area much higher or in a direction that was not appropriate for the approach. Furthermore, missed approaches, i.e., flights that did not reach the runway threshold, were removed.

2.2.5. Feature Transformation

For random forest methods, there was no normalization or standardization of the data required since RF models are based on decision trees, which are invariant to linear transformations because they use indicators like information gain, Gini index, and variance reduction, which are not affected by scaling of the variables. This stays true for quantile regression forest (QRF), even though it is a regression method because there is no explicit regression coefficient to measure the relationship between features and the response variable.

Finally, the last step of the data preparation phase was the decomposition of the cyclical features representing wind direction and aircraft track into their Cartesian components. This step was required, since, for cyclical features, the relation between distant values could be much closer than indicated numerically, like for instance tracks 005 and 355. To account for this, we split the aircraft speed vector and wind speed vector each into two features, one for the sine and one for the cosine part. This corresponds to the handling of wind direction during the data imputation phase, as described above.

2.3. Exploratory Data Analysis

After the preprocessing phase, there were 4886 flights available contributing to a total of 452,037 data points. Figure 4 visualizes the tracks that were included in the data set.

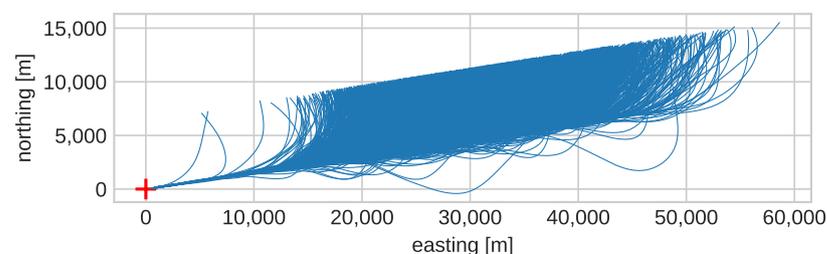


Figure 4. The flight tracks that are part of the data set that is used for this study. The red cross depicts the runway threshold.

For all recorded flight tracks, the time-to-fly could be determined for every point along the flight path. Figure 5 depicts for all tracks of the data set the time-to-fly along the flight path. Furthermore, Figure 6 visualizes the time-to-fly distribution using a histogram over

all the values in the data set. It highlights the values that were expected at first contact with an aircraft entering the active area. It can be seen that those were not common distributions. The mean time-to-fly over all the data points was 206.58 seconds, which is depicted by a green line in the diagram.

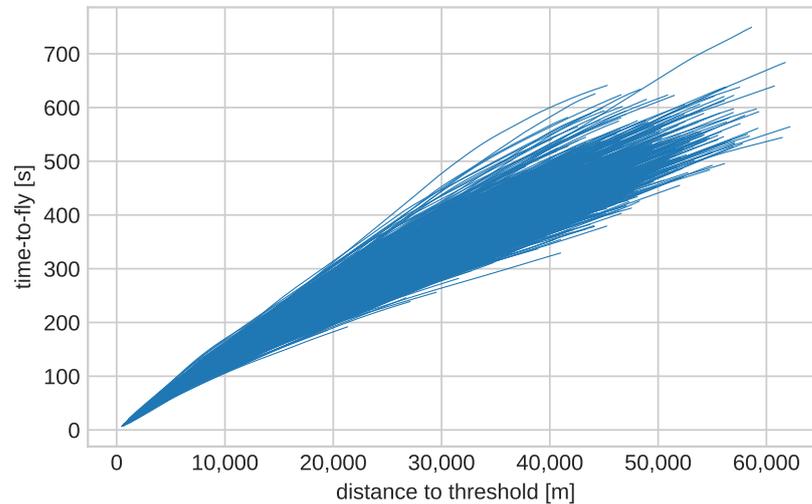


Figure 5. Time-to-fly above the distance to threshold.

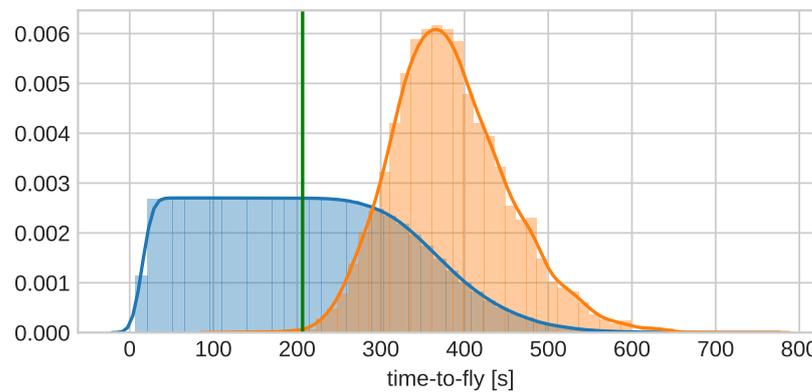


Figure 6. Distribution of time-to-fly values. The blue histogram comprises the full data set. The orange one depicts the time-to-fly at first entry of an aircraft. Solid curves depict the kernel density estimation. The green line depicts the mean over all time-to-fly values.

3. Quantile Regression Forests

3.1. Quantile Regression

Predictions made by machine learning (ML) algorithms are never absolute but rather probabilistic. Despite this fact, most of the ML methods usually provide a point estimate, which reflects only one property of the underlying probability distribution, depending on what the models are trained for, usually the conditional mean $\mathbb{E}(Y|X = x)$. However, many applications would benefit from more distinct information about the underlying distribution. Sometimes it is crucial to obtain a prediction interval, rather than a single value. For example, in air traffic control it is not only required to know what the state of an aircraft (i.e., its position, orientation, and speed) may be on average when performing predictions but to know the possible distribution of aircraft state to be able to detect developing safety-critical situations. The field of collision risk modeling makes extensive use of probability distributions over the position of aircraft [57,58]. Employing quantile regression methods allow for determining additional supports of the probability distribution by predicting several quantiles [59].

For a conditional random variable $Y|X = x$ and its cumulative distribution function (CDF) $F(y|X = x)$, the τ th quantile (with $0 \leq \tau \leq 1$) of the random variable is determined using Equation (2). When employing quantile regression techniques, the quantiles are the model's output and the probability function is what needs to be derived from these values. Figure 7 visualizes an exemplary quantile function as constructed from the model's prediction output for 100 quantiles with $\tau \in \{0.01, 0.02, 0.03, \dots, 1.00\}$.

$$Q_x(\tau) = F^{-1}(\tau) \quad (1)$$

$$= \inf\{\mu : F(\mu|X = x) \geq \tau\} \quad (2)$$

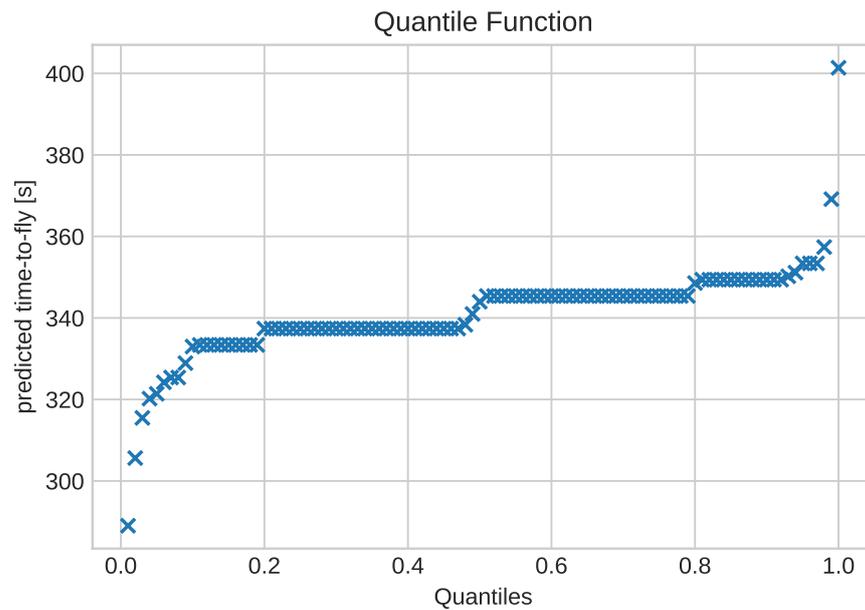


Figure 7. A quantile function constructed from the responses at 100 collocation points ($\tau \in \{0.01, 0.02, 0.03, \dots, 1.00\}$) as given by the QRF model.

Deriving the CDF from the quantile function requires inverting the relationship presented above in Equation (1). Since the prediction can be performed for discrete quantiles only, the inverse of the quantile function, $Q(\tau)^{-1}$, depicts a right-continuous, non-derivable empirical cumulative distribution function (ECDF) $F(t)$, which is visualized in Figure 8. This distribution can then be used to perform further investigations that consider the uncertainty in the prediction of time-to-fly, like a trade-off analysis as presented later in Section 4.

Usually, when performing quantile regression, ML models need to be trained individually for all the quantiles that are required to provide the necessary supports of the underlying distribution, which results in a large set of models, each of which requires training time and memory. In contrast, a QRF model needs to be trained only once and can afterward provide predictions for arbitrary quantiles [60].

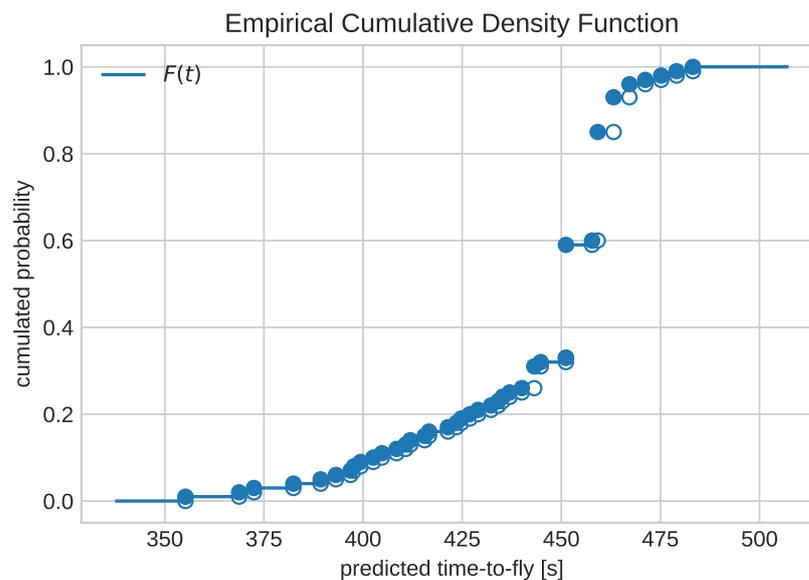


Figure 8. Plot of inverse quantile function, which results in a right-continuous empirical cumulative distribution function.

3.2. Random Forest

Random Forest is an ensemble method based on binary decision trees. It belongs to the category of supervised machine learning algorithms. A prediction of the target value is achieved based on the outcomes of multiple, distinct trees, hence it is called a forest. At each level of a tree, the decision, which branch to follow, is made based on the value of certain features. Depending on the objective of the prediction (classification or regression) the trees' outputs are regarded as a vote for a certain class or contribute to the calculation of a numerical value.

In general, a random forest model is trained via a randomized bootstrapping process [61]. This means the training data set is sampled n times, depending on the number of trees in the forest. Each sample contains exactly the same amount of observations as the original data set. The observations are chosen randomly from the base data set, which leads to the effect of duplicate and missing records. Therefore, this method is called sampling with replacement.

After sampling n individual sets of training data, the training process starts growing one tree per sample. A tree is grown by an iterative splitting of the data sample into two parts based on a decision criterion that gives the reduction in variance to determine the quality of a split [61].

The training process of a QRF model does not differ from training a standard random forest model except how the values in the leaves of each tree are handled. While in a classic random forest regression model, only the average of all observations that a leaf contains is stored, in QRF all the observations of the training phase are preserved. In the prediction phase, those values are necessary to calculate individual weights for constructing the quantile function as described in detail in [60].

3.3. Model Quality Assessment

The quality of the model has to be assessed during the training process, as well as afterward to allow for a comparison with similar approaches. We first provide a brief summary of common measures for models that provide point estimates, whereafter metrics are presented that explicitly account for quantile estimates and probabilistic predictions. During training phase, quality measures are typically termed *loss function* or *loss value*. However, in this section, we use the terms *loss*, *score*, and *measure* interchangeably.

3.3.1. Point Estimate Assessment

For quantification of model performance and accuracy, a wide variety of methods have been developed in the past [62]. All of those methods target classic, not quantile-enabled ML approaches that could provide point estimates. We select the three most commonly employed measures and briefly discuss them in the following. Equations (3)–(5) depicts the equations on how to calculate them, where \hat{y} denotes the predicted value and y the associated expected value of a sample from a data set with N samples in total.

$$RMSE = \sqrt{\frac{1}{N} \sum_{n=1}^N (y_n - \hat{y}_n)^2} \quad (3)$$

$$MAE = \frac{1}{N} \sum_{n=1}^N |y_n - \hat{y}_n| \quad (4)$$

$$MAPE = \frac{1}{N} \sum_{n=1}^N \frac{|y_n - \hat{y}_n|}{y_n} \quad (5)$$

The root mean squared error (RMSE) seems to be the most common measure, although it is hardly interpretable. Rather, it is more suited to compare different models or model configurations against each other regarding their prediction quality. In contrast, the mean absolute error (MAE) gives the average deviation of the prediction from the expectation in the same domain as the dependent variable, which makes it much more meaningful. However, the absolute error integrates a large variety of magnitude both of absolute time-to-fly values and prediction errors. For instance, predicting a time-to-fly of 550 s when the actual value is 500 s is much better than predicting 250 s when the actual value is 200 s. In this example, even though the absolute error is identical, the relative error is 10% in the former case and 25% in the latter. Therefore, the scale-independent mean absolute percentage error (MAPE) measure is employed to provide a third quantification for the quality of the model that constitutes the average relative prediction error.

3.3.2. Probabilistic Prediction Assessment

To provide probabilistic predictions instead of point estimates, it is appropriate to additionally calculate measures that are tailored to determine the accuracy of the prediction. Therefore, probability scoring as a measure for the accuracy of probabilistic predictions of the QRF model is additionally applied. Two common scoring functions that specifically target probabilistic prediction assessment are the *pinball loss (PL)* and the more comprehensive continuous ranked probability score (CRPS). Those measures allow for comparative prediction evaluation, i.e., to compare different models or model configurations against each other.

The pinball loss L_τ is an asymmetric, piecewise linear function, that acts as an accuracy measure for single quantile predictions [63]. The corresponding quantile score or quantile loss is calculated for a single quantile as depicted in Equation (6), where q_τ denotes the prediction for quantile τ and y the actually observed value. The overall model performance is determined by averaging the PL values over all the predicted quantiles $\tau \in [0, 1]$.

$$\begin{aligned} L_\tau(q_\tau, y) &= \mathbb{1}_{q_\tau < y} (y - q_\tau) \tau + \mathbb{1}_{y \leq q_\tau} (q_\tau - y) (1 - \tau) \\ &= (\tau - \mathbb{1}_{y \leq q_\tau}) (y - q_\tau) \end{aligned} \quad (6)$$

Since we are not interested in the score of individual quantiles but rather in the accuracy of the resulting distribution as a whole, we employ a more general score function, specifically, the continuous ranked probability score, which is the integral of the pinball score overall quantiles of the underlying distribution [64]. It generalizes the MAE for probabilistic predictions by measuring the distance between the predicted CDF $\hat{F}(x)$ and the CDF of the observed value, which itself is basically the Heaviside function (also known as *step function*, cf. Equation (8)), as depicted in Equation (7) [65]. The overall performance of the model is then calculated by averaging the CRPS results over all predictions from the

validation data set. Since it is defined in the same domain as the dependent variable, it is directly comparable to the MAE of a point estimate model.

$$CRPS(\hat{F}, y) = \int_{-\infty}^{\infty} (\hat{F}(x) - H(x - y))^2 dx \quad (7)$$

with

$$H(x) = \mathbb{1}_{x \geq 0} = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{else} \end{cases} \quad (8)$$

Quantile decomposition can be applied to transform the calculation of the CRPS in the form provided in Equation (9), which allows us to plugin the quantiles that are provided by the QRF model directly [66,67]. Changing the notation from Heaviside to indicator function and some additional repositioning of the operands, it can be seen from the resulting Equation (10) that the integrand is the pinball loss function from Equation (6) [68].

$$CRPS(\hat{F}, y) = 2 \int_0^1 (H(q_\tau - y) - \tau)(q_\tau - y) d\tau \quad (9)$$

$$= 2 \int_0^1 (\tau - \mathbb{1}_{y \leq q_\tau})(y - q_\tau) d\tau \quad (10)$$

4. Prediction of Time-To-Fly and Compression Buffer Compensation

4.1. Implementation

For implementing the prediction model we set up a toolchain comprising widely-used ML frameworks from the Python [69] and R [70] community, respectively. Data preparation and analysis steps were conducted using the `scikit-learn` library [71], which provides several methods that help to accomplish standard ML tasks, like splitting a data set into train and test data, strategies for hyper-parameter tuning, accuracy measures, etc. The R package `quantregForest` [72] from the author of the original paper on QRF [60] provided implementations for growing a forest and making predictions for arbitrary quantiles. It has been found to provide higher performance in terms of calculation speed and memory usage than the `scikit-garden` adaption written in Python [73].

After the preparatory steps of data acquisition, merging, cleaning, and transformation, the final data set was split into three parts before applying the QRF algorithm. The first part is the training subset, which was used to fit the model and which comprises 60% of the available data. Second, the validation data set comprised 20% of the data. It was used for model assessment during hyper-parameter tuning. The remaining 20% formed the test data set used after the fitting and tuning processes are finished to provide an assessment of the model's performance in terms of prediction quality and accuracy according to the measures presented above. Sampling training, validation, and test data from the same base data set ensured that the data comes from the same distribution.

4.2. Hyper-Parameter Tuning

A QRF model provides several parameters for configuration, hyper-parameters, which are shared with the underlying, general random forest algorithm. Those parameters are the number of trees to grow, the number of features to consider in a branch split (`mtry`), and the minimum number of samples that a leaf node should contain (`nodesize`). Following the size of the data set, the calculation time and memory requirements did not allow us to investigate every single variation of the different parameters. Therefore, we started with a fixed population size of 100 trees per forest to investigate the influence of the other two parameters. For regression trees, the default values are 5 for the `nodesize` parameter and 4 (more specifically: number of features, which is 12 in our data set, divided by 3) for the `mtry` parameter. Therefore, we started hyper-parameter search in the vicinity of those values. For each combination, we calculated the RMSE, MAE, and CRPS based on samples from the validation data set. The results are presented in Figure 9.

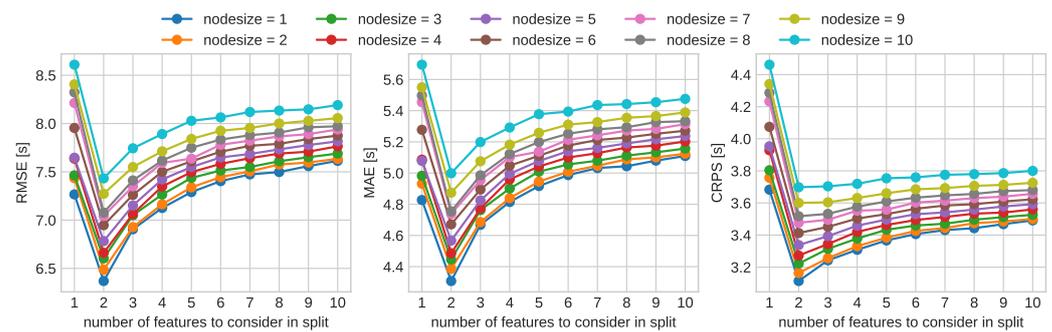


Figure 9. Results of three different accuracy measures for various model configurations consisting of a combination of parameters for the number of features to consider in a split, and for the minimum number of samples in the leave nodes (nodesize). In all three cases, the best performance is reached for the configuration with two features per split and a single data point per leave. In all cases, the number of trees per forest is 100. All calculations were conducted on the validation data set.

Subsequently, we selected the best combination of the feature-split and node-size parameters and performed additional calculations for the different number of trees to find the optimal size of the forest. The results are presented in Figure 10. As can be seen from the figure, the number of trees influenced the model's prediction quality positively with increasing population size, which corresponded to the general rule of thumb to use as many trees as possible, which is primarily limited by the memory available. However, the change in CRPS between a forest size of 100 to 900 trees was magnitudes lower than what could be achieved by tuning the node size and feature split parameters. The improvement was below 0.45% from the highest score at a population of 100 trees and the lowest (best) score for 800 trees.

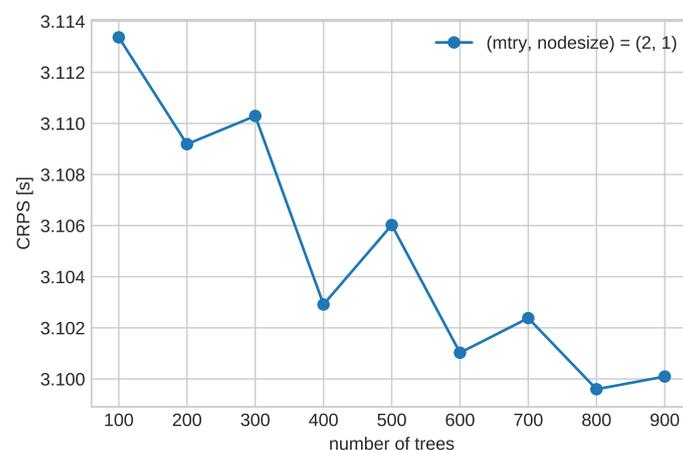


Figure 10. continuous ranked probability score based on the validation data set for different tree population sizes and a fixed setting of node-size (=1) and feature-split (=2) parameters.

4.3. Model Assessment

After training and tuning the model using the train and validation data subsets, we provided an assessment of the model based on predictions made from the test data set, which the model did never see before. The test data, therefore, provided an unbiased assessment of model performance and allows for comparison with other models. The models parameter were selected according to the results of the hyper-parameter tuning ($mtry = 2$, $nodesize = 1$, $ntree = 800$). Besides accuracy measures for probabilistic predictions, we additionally included measures for point estimate accuracy to allow for comparison with similar models that did not perform probabilistic predictions. The results are summarized in Table 1.

Table 1. Final model accuracy assessment based on predictions performed on the test set.

| RMSE | MAPE | MAE | CRPS |
|-------|-------|-------|-------|
| 6.3 s | 2.2 % | 4.3 s | 3.1 s |

4.4. Determining the Separation Buffer

The prediction of time-to-fly for two aircraft in a leader-follower relationship could now be used to calculate the probability of a certain separation at the threshold. Since we focused on time-based-separation for now, this corresponded to the difference between the two random variables X and Y representing the time-to-fly predictions of the leading and following aircraft, respectively. According to the formulation of cross-correlation, the difference $Z = Y - X$ between two random variables with probability density functions f_X and f_Y was calculated as depicted in Equation (11). This required assuming independence between variables X and Y , which is usually not the case for real aircraft in a leader-follower relationship. We made this assumption here since it is essentially the goal of the support tool to allow the controller to place aircraft far enough from each other so that every aircraft can follow its own speed regime without further intervention and thus be independent of each other.

$$f_X(z) \star f_Y(z) = \int_{-\infty}^{\infty} f_X(u) f_Y(z + u) du \quad (11)$$

To account for the fact that the presented QRF model did not provide a density prediction but a quantile prediction, whose inverse was a right-continuous and non-differentiable ECDF, we reformulated the equation using the Riemann–Stieltjes integral, which led to the following form of the cross-correlation:

$$F_X(z) \star F_Y(z) = \int_{-\infty}^{\infty} F_Y(z + u) dF_X(u) \quad (12)$$

With Equation (12), it was now possible to compute the probability of an infringement of minimum required separation S_T as $P(Z \leq S_T) = F_Z(S_T)$. The other way around, given an acceptable probability for loss of separation p_{LoS} , the time-to-gain or time-to-lose was calculated as the difference between S_T and the infimum of all possible separation values z that fulfill the probability constraint, as shown in Equation (13). A positive result Δt depicts a time-to-lose, a negative value depicts a time-to-gain.

$$\begin{aligned} \Delta t &= S_T - \inf\{z : P(Z \leq z) \geq p_{LoS}\} \\ &= S_T - \inf\{z : F(z) \geq p_{LoS}\} \end{aligned} \quad (13)$$

The last step for this use case was determining the acceptable probability for a loss of separation. This could be done either by explicitly giving a value for the probability derived from external analysis, expert opinion, etc. However, since the overall goal of the presented approach was to optimize capacity utilization, we needed to find a trade-off between two effects that led to decreased capacity utilization: a loss of separation (LoS) and a too wide spacing. Thus, it was required to balance the probabilities of both approaches to find the optimal initial separation of aircraft. Specifically, the probability for a LoS p_{LoS} should be equal to p_{LoC} , which depicts the probability to lose capacity due to too wide separation, as depicted in Equation (14). Here, S_T is the target separation at the threshold, which may usually be the minimum required wake vortex or radar separation. ΔS denotes the exceedance of the minimum required separation that is tolerable. When the separation exceeds $S_T + \Delta S$, capacity utilization is assumed to decrease by one movement (e.g., 1 arrival per hour). We introduced an additional factor γ , which allowed weighing the effects

of separation infringement and too wide separation events against each other, e.g., $\gamma = 2$ means that two arrivals lost due to too wide separation are equally acceptable as one LoS.

$$\begin{aligned}
 p_{LoS} &= p_{LoC} \\
 P_Z(Z \leq S_T - \Delta t) &= 1 - P_Z(Z \leq S_T + \gamma \Delta S - \Delta t) \\
 F_Z(S_T - \Delta t) &= 1 - F_Z(S_T + \gamma \Delta S - \Delta t)
 \end{aligned} \tag{14}$$

With this set of equations, it was now possible to determine the time-to-gain or time-to-lose either depending on an explicitly given probability for loss of separation. Alternatively, these time values could be calculated from a weighted trade-off between two different scenarios that could lead to degraded capacity utilization and if required, compute the LoS probability from that.

4.5. Model Application

Finally, we present an exemplary use case in which we show how the developed model was applied to derive advisory for optimizing separation. For this, two aircraft (leader and follower) were randomly selected from the dataset. The aircraft's state values were then fed into the QRF model, resulting in a prediction of two quantile functions representing the time-to-fly of each aircraft. Those predictions are depicted in Figure 11, in which the quantile functions are already transformed into ECDF form.

Now, the difference between the time-to-fly distributions of the following and leading aircraft was calculated via cross-correlation as shown in Equation (12). The result represents the distribution of time-based separation at the threshold, which is illustrated by the blue curve in Figure 12. The red vertical line depicts the required separation of 60 s that has to be maintained. In this case, the probability for a separation infringement $p_{LoS} = F_Z(60)$ is 10%.

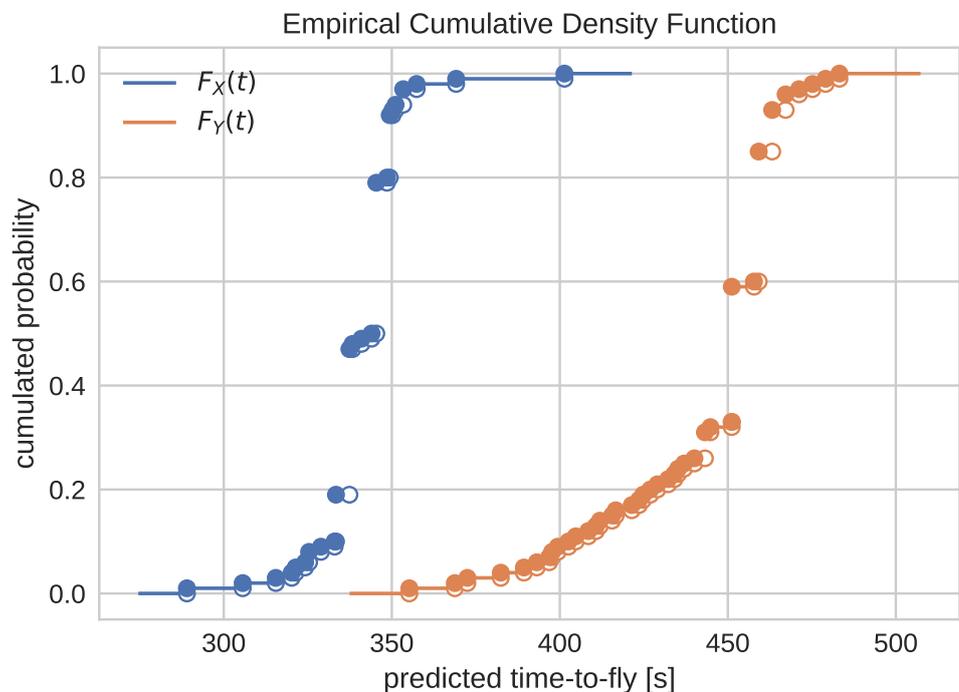


Figure 11. The right-continuous empirical cumulative distribution functions given by the QRF model as result of the time-to-fly predictions for two aircraft: leader ($F_X(t)$, blue) and follower ($F_Y(t)$, orange).

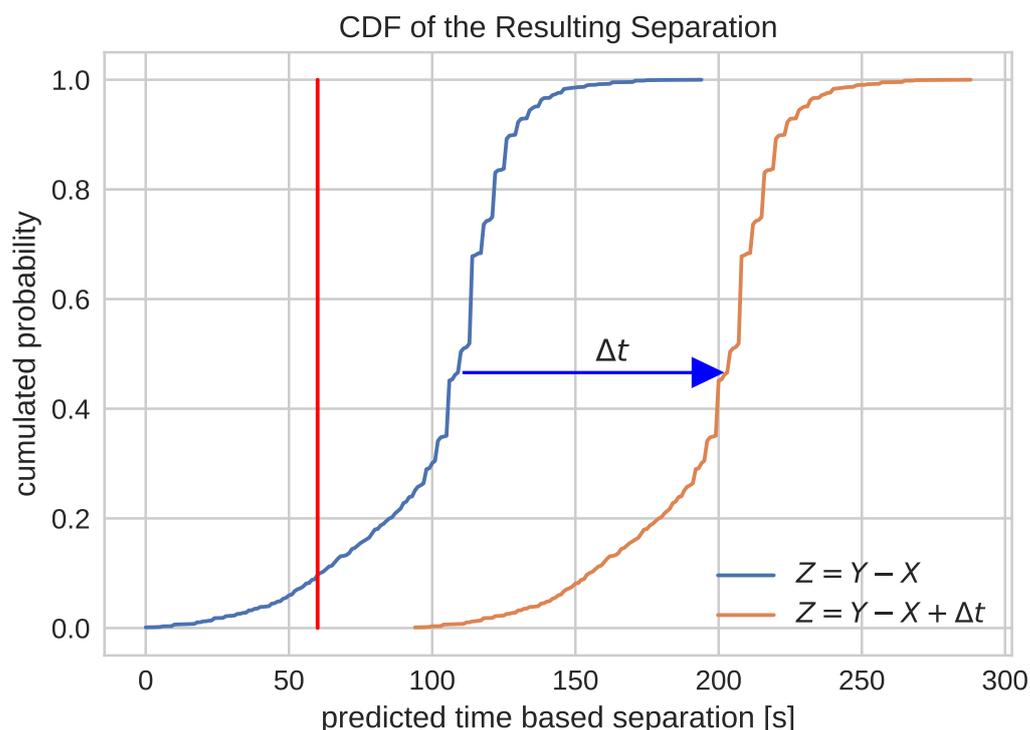


Figure 12. The predicted time-based separation between leader and follower (blue curve) will lead to a separation infringement (vertical red line) with probability of 0.1. The follower has to lose 84 seconds to reduce the probability below 10^{-4} (orange curve).

Subsequently, following the first of the two options presented above, we could now calculate a time-to-lose by setting the acceptable probability for loss of separation to $p_{LoS} = 10^{-4}$, i.e., only one in 10,000 approaches should lead to a separation infringement. This value may, for instance, be acquired by expert judgement. With Equation (13), the time-to-lose was determined to be 84 s for this example.

Following the second option, which finds Δt based on the trade-off between separation infringement and too wide separation, ΔS had to be determined. For this example, we simplified this task and took a homogeneous traffic mix (medium aircraft category only) assuming an average separation of 60 seconds. This means, an overall separation of $S_T + \Delta S = 120$ s or more was probable to lead to the loss of one arrival. Additional assuming that a loss of separation was twice as worse as a lost arrival due to too wide separation, we set $\gamma = 2$ and can use Equation (14) to calculate Δt , which is 34 s in this case (with $p_{LoS} \approx 1.8\%$).

This example case demonstrates how the developed methodology can be applied to obtain advice regarding time-to-lose or time-to-gain based on probabilistic predictions. It remains a strategic decision of the stakeholders, specifically the air navigation service provider (ANSP), which option (preselect p_{LoS} or trade-off between p_{LoS} and p_{LoC}) to choose for calculating Δt .

Since our data set does not contain information about the applied separation at the time of recording, we currently cannot evaluate the effect of the model when applied as described above. Specifically, we do not have the values for S_T that have been applied at the time of recording. The next step will therefore be to transfer the model into an ATC simulator and evaluate the application in human in the loop simulations.

5. Conclusions and Outlook

In this paper, we presented the development of a quantile regression forest model to predict the time-to-fly of aircraft from the turn onto the final approach course to the runway threshold. The quality of the model in terms of point estimate accuracy as well

as probability prediction has been investigated regarding several model error measures. Finally, we presented an approach on how to derive advice regarding time-to-gain/time-to-lose from the predictions depending on the trade-off between two scenarios leading to degraded capacity utilization.

Following the development of the initial model, the next step is to enable the model to be applicable for distance-based-separation as well. At the current stage, the proposed model considers the present state of the aircraft under consideration and recent weather only. It seems promising to also include (partially) the track history as a feature to improve the accuracy and precision of the prediction as well as extending the scope of features taking into consideration additional correlations between multiple aircraft in the sequence. However, it needs to be investigated whether the QRF approach can be employed for a variable length of time-series input.

Furthermore, the prediction of time-to-fly and its implication to the air traffic controller is just the first step towards a decision support tool. At the current state, the tool would only show a possible goal (time-to-gain/-lose) but the controller has to decide how to achieve the proposed system state. A future step is to enhance the tool's capabilities towards generating recommendations for ATC commands that help to reach the goal, using advanced ML techniques.

Author Contributions: Conceptualization, S.F. and M.S.; methodology, S.F.; software, S.F.; validation, S.F. and M.S.; formal analysis, S.F.; investigation, S.F.; resources, H.F.; data curation, S.F.; writing—original draft preparation, S.F. and M.S.; writing—review and editing, S.F. and M.S.; visualization, S.F.; supervision, M.S.; project administration, H.F. and M.S.; funding acquisition, H.F. All authors have read and agreed to the published version of the manuscript.

Funding: This work is part of the FUTURE project, which is sponsored by DFS Deutsche Flugsicherung GmbH and the German Federal Ministry for Economic Affairs and Energy in the course of the national aviation research program (Luftfahrtforschungsprogramm–LuFo). The authors would like to thank the Open Access Funding by the Publication Fund of the Technische Universität Dresden.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No new data were created in this study. Restrictions apply to the availability of the input data for this study.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

ADS-B Automatic Dependent Surveillance-Broadcast

AIBT actual in-block time

ANSP air navigation service provider

AOBT actual off-block time

ARDT aircraft ready time

ASMA Arrival Sequencing and Metering Area

ATOT actual take-off time

CDF cumulative distribution function

CRPS continuous ranked probability score

ECAC European Civil Aviation Conference

ECDF empirical cumulative distribution function

FAF final approach fix

IAS indicated air speed

ILS instrument landing system

LOC localizer

| | |
|-----------------|--|
| LORD | Leading Optimised Runway Delivery |
| LoS | loss of separation |
| MAE | mean absolute error |
| MAPE | mean absolute percentage error |
| METAR | meteorological aerodrome report |
| ML | machine learning |
| MTOM | maximum take-off mass |
| PDF | probability density function |
| PL | pinball loss |
| QRF | quantile regression forest |
| RECAT-EU | re-categorisation of ICAO wake turbulence separation |
| RMSE | root mean squared error |
| TAS | true air speed |
| TMA | terminal maneuvering area |

References

1. ICAO. *Procedures for Air Navigation Services—Air Traffic Management (PANS-ATM, Doc 4444)*; ICAO: Montreal, QC, Canada, 2007.
2. ICAO. *Air Traffic Services Planning Manual (Doc 9426)*; ICAO: Montreal, QC, Canada, 1984.
3. FAA. *JO 7110.65Y-Air Traffic Control*; FAA: Washington, DC, USA, 2019.
4. NATS. *Aeronautical Information Circular P 092/2017*; Technical Report; UK Aeronautical Information Services; NATS: Whiteley, Fareham, UK, 2017.
5. Eurocontrol. *European Aviation in 2040—Challenges of Growth Summary Report*; Eurocontrol: Brussels, Belgium, 2018.
6. Rooseleer, F.; Treve, V. *RECAT-EU-European Wake Turbulence Categorisation and Separation Minima on Approach and Departure*; Technical Report; Eurocontrol: Brussels, Belgium, 2015.
7. Eurocontrol. *Specification for Time Based Separation (TBS) for Final Approach*; Technical Report; Eurocontrol: Brussels, Belgium, 2016.
8. Holzäpfel, F.; Gerz, T.; Frech, M.; Tafferner, A.; Köpp, F.; Smalikhov, I.; Rahm, S.; Hahn, K.U.; Schwarz, C. The Wake Vortex Prediction and Monitoring System WSVBS Part I: Design. *Air Traffic Control Q.* **2009**, *17*, 301–322. [[CrossRef](#)]
9. Airports Council International Europe. *SESAR and the Digital Transformation of Europe's Airports*; 2018. Available online: <https://www.sesarju.eu/sites/default/files/documents/reports/SESARandthedigitaltransformationofeuropairports.pdf> (accessed on 25 January 2021).
10. European Commission. *Commission Implementing Regulation (EU) No 1207/2011*; European Commission: Brussels, Belgium, 2011.
11. European Commission. *Commission Implementing Regulation (EU) No 2017/386*; European Commission: Brussels, Belgium, 2017.
12. FAA. *14 CFR, FAR Section 91.225 “Automatic Dependent Surveillance-Broadcast (ADS-B) Out Equipment and Use”*; FAA: Washington, DC, USA, 2015.
13. Schäfer, M.; Strohmeier, M.; Lenders, V.; Martinovic, I.; Wilhelm, M. Bringing up OpenSky: A large-scale ADS-B sensor network for research. In *Proceedings of the 13th International Symposium on Information Processing in Sensor Networks*, Berlin, Germany, 15–17 April 2014; pp. 83–94.
14. Delovski, T.; Werner, K.; Rawlik, T.; Behrens, J.; Bredemeyer, J.; Wendel, R. ADS-B over Satellite The world's first ADS-B receiver in Space. In *Proceedings of the Small Satellites Systems and Services Symposium*, Majorca, Spain, 26–30 May 2014.
15. SESAR Deployment Manager. *ADS-B and Other Means of Surveillance Implementation Status*; SESAR Joint Undertaking: Brussels, Belgium, 2018.
16. Moreno-Torres, J.G.; Raeder, T.; Alaiz-Rodríguez, R.; Chawla, N.V.; Herrera, F. A unifying view on dataset shift in classification. *Pattern Recognit.* **2012**, *45*, 521–530. [[CrossRef](#)]
17. Murdoch, W.J.; Singh, C.; Kumbier, K.; Abbasi-Asl, R.; Yu, B. Definitions, methods, and applications in interpretable machine learning. *Proc. Natl. Acad. Sci. USA* **2019**, *116*, 22071–22080. [[CrossRef](#)]
18. Basora, L.; Courchelle, V.; Bedouet, J.; Dubot, T. Occupancy Peak Estimation from Sector Geometry and Traffic Flow Data. In *Proceedings of the 8th SESAR Innovation Days*, Salzburg, Austria, 3–6 December 2018.
19. Gariel, M.; Srivastava, A.N.; Feron, E. Trajectory clustering and an application to airspace monitoring. *IEEE Trans. Intell. Transp. Syst.* **2011**, *12*, 1511–1524. [[CrossRef](#)]
20. Basora, L.; Morio, J.; Mailhot, C. A Trajectory Clustering Framework to Analyse Air Traffic Flows. In *Proceedings of the 7th SESAR Innovation Days*, Belgrade, Serbia, 28–30 November 2017.
21. Olive, X.; Morio, J. Trajectory clustering of air traffic flows around airports. *Aerosp. Sci. Technol.* **2019**, *84*, 776–781. [[CrossRef](#)]
22. Olive, X.; Grignard, J.; Dubot, T.; Saint-Lot, J. Detecting Controllers' Actions in Past Mode S Data by Autoencoder-Based Anomaly Detection. In *Proceedings of the 8th SESAR Innovation Days*, Salzburg, Austria, 3–6 December 2018.

23. Das, S.; Matthews, B.L.; Srivastava, A.N.; Oza, N.C. Multiple kernel learning for heterogeneous anomaly detection: algorithm and aviation safety case study. In Proceedings of the 16th International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, 24–28 July 2010; pp. 47–56.
24. Olive, X.; Bieber, P. Quantitative Assessments of Runway Excursion Precursors using Mode S data. In Proceedings of the 8th International Conference for Research in Air Transportation, Barcelona, Spain, 26–29 June 2018.
25. Basora, L.; Olive, X.; Dubot, T. Recent Advances in Anomaly Detection Methods Applied to Aviation. *Aerospace* **2019**, *6*, 117. [[CrossRef](#)]
26. Lv, Y.; Duan, Y.; Kang, W.; Li, Z.; Wang, F. Traffic flow prediction with big data: A deep learning approach. *IEEE Trans. Intell. Transp. Syst.* **2015**, *16*, 865–873. [[CrossRef](#)]
27. Di Ciccio, C.; Van der Aa, H.; Cabanillas, C.; Mendling, J.; Prescher, J. Detecting flight trajectory anomalies and predicting diversions in freight transportation. *Decis. Support Syst.* **2016**, *88*, 1–17. [[CrossRef](#)]
28. Liu, Y.; Hansen, M.; Lovell, D.J.; Ball, M.O. Predicting Aircraft Trajectory Choice—A Nominal Route Approach. In Proceedings of the 8th International Conference for Research in Air Transportation, Barcelona, Spain, 26–29 June 2018.
29. Gerdes, I.; Temme, A.; Schultz, M. Dynamic airspace sectorisation for flight-centric operations. *Transp. Res. Part C Emerg. Technol.* **2018**, *95*, 460–480. [[CrossRef](#)]
30. Gerdes, I.; Temme, A.; Schultz, M. From free-route air traffic to an adapted dynamic main-flow system. *Transp. Res. Part C Emerg. Technol.* **2020**, *115*, 102633. [[CrossRef](#)]
31. Herrema, F.; Curran, R.; Hartjes, S.; Ellejmi, M.; Bancroft, S.; Schultz, M. A machine learning model to predict runway exit at Vienna airport. *Transp. Res. Part E Logist. Transp. Rev.* **2019**, *131*, 329–342. [[CrossRef](#)]
32. Schultz, M.; Rosenow, J.; Olive, X. A-CDM Lite: Situation awareness and decision making for small airports based on ADS-B data. In Proceedings of the 9th SESAR Innovation Days, Athens, Greece, 2–6 December 2019.
33. Reitmann, S.; Alam, S.; Schultz, M. Advanced Quantification of Weather Impact on Air Traffic Management-Intelligent Weather Categorization with Machine Learning. In Proceedings of the 13th USA/Europe ATM R&D Seminar, Vienna, Austria, 17–21 June 2019.
34. Schultz, M.; Reitmann, S. Machine learning approach to predict aircraft boarding. *Transp. Res. Part C: Emerg. Technol.* **2019**, *98*, 391–408. [[CrossRef](#)]
35. Sun, J.; Ellerbroek, J.; Hoekstra, J.M. Aircraft initial mass estimation using Bayesian inference method. *Transp. Res. Part C Emerg. Technol.* **2018**, *90*. [[CrossRef](#)]
36. Sun, J. Open Aircraft Performance Modeling Based on an Analysis of Aircraft Surveillance Data. Ph.D. Thesis, Delft University of Technology, Delft, The Netherlands, 2019.
37. Schultz, M.; Olive, X.; Rosenow, J.; Fricke, H.; Alam, S. Analysis of airport ground operations based on ADS-B data. In Proceedings of the 1st conference on Artificial Intelligence and Data Analytics for Air Transportation, Singapore, 3–4 February 2020.
38. Chatterji, G. Short-term trajectory prediction methods. In Proceedings of the Guidance, Navigation, and Control Conference and Exhibit, Portland, OR, USA, 9–11 August 1999. [[CrossRef](#)]
39. Radišić, T.; Krajček, K.; Kovačić, I. 4D Trajectory Prediction for Arrival and Approach Phases of Flight. International Navigation Conference 2010. 2010. Available online: https://bib.irb.hr/datoteka/858355.Radii_Krajek_Kovai_-_4D_Trajectory_Prediction_for_Arrival_and_Approach_Phases_of_Flight.pdf (accessed on 25 January 2021).
40. Paek, H.; Lee, K.; Vela, A.E. En-route Arrival Time Prediction using Gaussian Mixture Model. In Proceedings of the 9th International Conference on Research and Air Transportation (ICRAT), Virtual Event, 15 September 2020.
41. Wang, Z.; Liang, M.; Delahaye, D. Automated data-driven prediction on aircraft Estimated Time of Arrival. *J. Air Transp. Manag.* **2020**, *88*, 101840. [[CrossRef](#)]
42. Ayhan, S.; Samet, H. Aircraft Trajectory Prediction Made Easy with Predictive Analytics. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; Association for Computing Machinery: New York, NY, USA, 2016; KDD '16, p. 21–30. [[CrossRef](#)]
43. Liu, Y.; Hansen, M. Predicting Aircraft Trajectories: A Deep Generative Convolutional Recurrent Neural Networks Approach. *arXiv* **2018**, arXiv:1812.11670.
44. Zeng, W.; Quan, Z.; Zhao, Z.; Xie, C.; Lu, X. A Deep Learning Approach for Aircraft Trajectory Prediction in Terminal Airspace. *IEEE Access* **2020**, *8*, 151250–151266. [[CrossRef](#)]
45. Tielrooij, M.; Borst, C.; van Paassen, M.M.; Mulder, M. Predicting Arrival Time Uncertainty from Actual Flight Information. In Proceedings of the 11th USA/Europe Air Traffic Management Research and Development Seminar (ATM2015), Lisbon, Portugal, 23–26 June 2015.
46. Takeichi, N.; Yamada, T.; Senoguchi, A.; Koga, T. Development of a Flight Time Uncertainty Model for Four-Dimensional Trajectory Management. *J. Air Transp.* **2020**, *28*, 134–143. [[CrossRef](#)]
47. Tielrooij, M.; Borst, C.; Mulder, M. Supporting Arrival Management Decisions by Visualising Uncertainty. In Proceedings of the 3rd SESAR Innovation Days, Stockholm, Sweden, 26–28 November 2013.
48. Barratt, S.T.; Kochenderfer, M.J.; Boyd, S.P. Learning Probabilistic Trajectory Models of Aircraft in Terminal Airspace From Position Data. *IEEE Trans. Intell. Transp. Syst.* **2019**, *20*, 3536–3545. [[CrossRef](#)]

49. Rocha Murça, M.C.; de Oliveira, M. A Data-Driven Probabilistic Trajectory Model for Predicting and Simulating Terminal Airspace Operations. In Proceedings of the 2020 AIAA/IEEE 39th Digital Avionics Systems Conference (DASC), Virtual Event, 11–15 October 2020; pp. 1–7. [CrossRef]
50. Herrema, F.; Treve, V.; Curran, R. Typical additional spacing-buffer to apply at 4DME for delivering separation minima. In Proceedings of the 34th Digital Avionics Systems Conference (DASC), Prague, Czech Republic, 13–17 September 2015. [CrossRef]
51. Herrema, F.; Curran, R.; Treve, V.; Visser, H.G. Evaluation of feasible machine learning techniques for predicting the time to fly and aircraft speed profile on final approach. In Proceedings of the 7th International Conference on Research and Air Transportation (ICRAT), Philadelphia, PA, USA, 20–24 June 2016.
52. van Baren, G.; Treve, V.; Herrema, F. Predicting time to fly on final approach for optimized delivery of separation. In Proceedings of the Integrated Communications Navigation and Surveillance (ICNS), Herndon, VA, USA, 19–21 April 2016. [CrossRef]
53. Cappellazzo, V.; Treve, V.; De Visscher, I.; Chalou-Morgan, C. Design Principles for a Separation Support Tool Allowing Optimized Runway Delivery. In Proceedings of the Aviation, Technology, Integration, and Operations Conference, Atlanta, GA, USA, 25–29 June 2018. [CrossRef]
54. Cappelleras, L. *Additional ASMA Time Performance Indicator Document*; Eurocontrol Performance Review Unit: Brussels, Belgium, 2015.
55. Valor, G.B.; López, D.J.M.G. OGIMET—Professional Information about Meteorological Conditions in the World. Available online: <http://www.ogimet.com> (accessed on 23 January 2020).
56. Schultz, M.; Lorenz, S.; Schmitz, R.; Delgado, L. Weather impact on airport performance. *Aerospace* **2018**, *5*, 109. [CrossRef]
57. Förster, S.; Fricke, H.; Vogel, M. Using Agent-Based Modeling to Determine Collision Risk in Complex TMA Environments. In Proceedings of the 8th International Conference on Research in Air Transportation (ICRAT), Barcelona, Spain, 26–29 June 2018.
58. Förster, S.; Fricke, H.; Rabiller, B.; Hickling, B.; Favennec, B.; Zeghal, K. Analysis of safety performances for parallel approach operations with performance based navigation. In Proceedings of the 19th USA/Europe Air Traffic Management Research and Development Seminar (ATM Seminar), Vienna, Austria, 17–21 June 2019.
59. Takeuchi, I.; Le, Q.V.; Sears, T.D.; Smola, A.J. Nonparametric quantile estimation. *J. Mach. Learn. Res.* **2006**, *7*, 1231–1264.
60. Meinshausen, N. Quantile regression forests. *J. Mach. Learn. Res.* **2006**, *7*, 983–999.
61. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
62. Hyndman, R.J.; Koehler, A.B. Another look at measures of forecast accuracy. *Int. J. Forecast.* **2006**, *22*, 679–688. [CrossRef]
63. Koenker, R.; Bassett, G. Regression Quantiles. *Econometrica* **1978**, *46*, 33–50. [CrossRef]
64. Matheson, J.E.; Winkler, R.L. Scoring Rules for Continuous Probability Distributions. *Manag. Sci.* **1976**, *22*, 1087–1096. [CrossRef]
65. Hersbach, H. Decomposition of the Continuous Ranked Probability Score for Ensemble Prediction Systems. *Weather Forecast.* **2000**, *15*, 559–570. [CrossRef]
66. Laio, F.; Tamea, S. Verification tools for probabilistic forecasts of continuous hydrological variables. *Hydrol. Earth Syst. Sci.* **2007**, *11*, 1267–1277. [CrossRef]
67. Jordan, A.; Krüger, F.; Lerch, S. Evaluating probabilistic forecasts with scoringRules. *arXiv* **2017**, arXiv:stat.CO/1709.04743.
68. Ben Taieb, S.; Huser, R.; Hyndman, R.J.; Genton, M.G. Forecasting Uncertainty in Electricity Smart Meter Data by Boosting Additive Quantile Regression. *IEEE Trans. Smart Grid* **2016**, *7*, 2448–2455. [CrossRef]
69. Python Software Foundation. *Python Language Reference*; Version 3.8; Python Software Foundation: Wilmington, DE, USA, 2020.
70. R Core Team. *R: A Language and Environment for Statistical Computing*; R Foundation for Statistical Computing: Vienna, Austria, 2020.
71. Pedregosa, F.; others. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
72. Meinshausen, N. *quantregForest: Quantile Regression Forests*; R Package Version 1.3-7; 2017. Available online: <https://mobile.jmlr.org/papers/volume7/meinshausen06a/meinshausen06a.pdf> (accessed on 25 January 2021).
73. Scikit-Garden. Available online: <https://scikit-garden.github.io> (accessed on 23 January 2020).