The Journal of Supercomputing (2024) 80:17212–17246
https://doi.org/10.1007/s11227-024-06060-6

A deep learning-based approach for predicting in-flight estimated time of arrival

Jorge Silvestre[1] · Miguel A. Martínez-Prieto[1] · Anibal Bregon[1] · Pedro C. Álvarez-Esteban[2]

Accepted: 9 March 2024 / Published online: 24 April 2024

Abstract

Predictability is key for efficient and safe air traffic management. In particular, accurately estimating time of arrival for current passenger flights may help terminal controllers to plan ahead and optimize airport operations in terms of safety and resource allocation. While traditional physics-based simulations are still widely used, they are complex to model and often fail to include many factors affecting the progress of a flight. In this paper, we propose a deep learning approach based on LSTM that leverages the 4D trajectory of the flight and weather data at the destination airport, to accurately predict estimated time of arrival. We evaluate our model on flights arriving at Adolfo Suárez-Madrid Barajas airport (Spain), in the first three quarters of 2022, achieving a mean absolute error of 2.65 min over the entire flight and reporting competitive short- and long-term predictions at different spatial and temporal horizons.

Keywords LSTM networks · Air traffic management · Estimated time of arrival

Jorge Silvestre, Miguel A. Martínez-Prieto, Anibal Bregon and Pedro C. Álvarez-Esteban contributed equally to this work.

✉ Jorge Silvestre
jorge.silvestre@uva.es

Miguel A. Martínez-Prieto
miguelamp@uva.es

Anibal Bregon
anibal.bregon@uva.es

Pedro C. Álvarez-Esteban
pedrocesar.alvarez@uva.es

[1] Department of Computer Science, University of Valladolid, Plaza de la Universidad, 1, Segovia 40005, Spain

[2] Department of Statistics and Operational Research, University of Valladolid, Paseo de Belén, 7, Valladolid 47011, Spain

1 Introduction

Intelligent Transportation Systems (ITS) play a key role in making transport services more predictable, improving their safety and reducing their costs and emissions. These systems use data collected from a wide range of in-vehicle sensors and other context-specific sources (e.g. scheduled services, movement flows, traffic congestion, weather, etc.) to monitor, analyse and improve transport operations.

The role of ITS is particularly interesting in the case of air traffic management (ATM), a sector that has almost recovered pre-pandemic traffic levels after the severe impact of the air traffic restrictions caused by COVID-19 [1]. This return to normality also brings back old problems, like *flight delays*, one of the common issues in this sector. A recent report of EUROCONTROL [2] shows that more than 35% of flights in Europe arrived at least 15 min. late with respect to their scheduled arrival time, what has a huge impact in terms of costs, emissions, and passenger satisfaction. This report also identifies reactionary delays (those caused by previous flights) as the main cause of flight delays, thus stressing the importance of having accurate predictions of the time of arrival to plan ahead and minimize cascading effects. Additionally, other sources of delays include airline or airport operations and weather conditions, among others. These situations increase the complexity of ATM operations, especially around airports, where air traffic controllers must monitor and handle incoming and outgoing flights, while making efficient use of the available resources and ensuring safety. To do this effectively, controllers must have a good understanding of the current situation and, when possible, predict ahead the future conditions of the airspace.

Flight plans have been one of the main information sources for air traffic control. Any flight taking place in European airspace must file its flight plan, indicating the intended flight path (in the form of *waypoints*, i.e. relevant geographic locations to describe the flight path), as well as the scheduled departure and arrival times, and other relevant information for planning purposes [3]. Flight plans are communicated up to one week before the start of the flight, but may be modified or amended at any time, enabling tactical decisions to be made in terms of resource allocation, estimation and meeting of the needs of all stakeholders (airlines, authorities and customers) and security assurance. In practice, flight plans provide only a rough description of the expected flight path based on forecast conditions that may change during the flight. However, in-flight updates are not frequent and are mainly due to significant changes in the flight plan, such as long delays at any point in the flight or diversions due to bad weather. This still leaves a high level of uncertainty for pilots and air traffic controllers, who often have to make decisions in the event of sudden changes. This has been reported as a potential factor of risk, given the workload and the pressure the controllers are under [4].

Surveillance systems are also used to assist air traffic controllers, by providing them with the position of the aircraft throughout the flight. ADS-B (Automatic Dependent Surveillance-Broadcast) [5] has progressively replaced secondary

radars for this purpose, taking advantage of the aircraft's capabilities to determine its position as well as other important flight parameters (altitude, speed, bearing, etc.), which are continuously emitted by the vehicle. ADS-B equipment is mandatory for aircraft operating commercial flights in the world's major airspaces and plays a key role in introducing the concept of *Trajectory-Based Operations* (TBO) [6] in intelligent ATM systems. TBO go beyond decision making based on flight plans thanks to the notion of *4D trajectory*, which integrates time into the 3D (latitude, longitude, and altitude) flight path [7]. Trajectories are thus described in terms of position and time and are agreed upon by all involved stakeholders to allow for better allocation of airspace and airport resources. Therefore, 4D trajectories enable flight delays to be considered as deviations from the expected trajectory, in the same way as changes in horizontal positions or flight levels, contributing to understand these deviations and improve the predictability of ATM operations.

This paper explores how 4D trajectories can be used to improve predictions of the *estimated time of arrival (ETA)*. ETA is a major factor for ATM operations, because it determines when a flight will arrive at the destination airport, allowing for efficient resource allocation in the transit airspace and at the airport. In the context of this work, ETA is defined as the estimated time until a flying aircraft lands at the destination airport, that is, until touchdown. Therefore, our study does not include taxiing times at the origin and destination airports. Most current research predicts ETAs in the Terminal Manoeuvring Area (TMA) [8–13], as this is where some of the most critical ATM operations take place, but these predictions can be valuable at any point along the flight path to cover as much airspace as possible. On the other hand, some studies [14] take an individual approach, looking at a single route or segmenting the traffic according to different criteria. However, as flights approach the airport, the traffic coming from the different routes becomes more homogeneous when performing the approach manoeuvres defined for that airport. The behaviour of the aircraft may also be similar during the flight, especially during the cruise phase. In other words, learning from multiple routes should help to identify and model increasingly rich flight patterns, rather than focusing on single routes that require, on the other hand, training and maintenance of specific models, with the additional costs that this entails.

Our approach treats flight trajectories as time series (including the four dimensions mentioned above and some other features from surveillance, flight plan and weather data) and trains a deep learning model to make ETA predictions for incoming flights to a given airport. In particular, we designed an architecture based on Long-Short Term Memory (LSTM) neural networks [15] to leverage flight dependencies in the long and short terms of the flight that influence on the accurate estimation of its time of arrival. This architecture combines data relative to surveillance, flight plans and weather conditions at the destination airport, to characterize the flight state and predict an accurate estimated time of arrival based on the actual conditions in which the flight is taking place. We apply a global approach, unlike other proposals that define a single model for each pair of origin and destination airports, to leverage the similarities between trajectories departing from different airports.

A comprehensive evaluation is conducted to analyse the performance of our proposal on the basis of the different parameters that characterise it and to compare its results with a selected baseline, which includes prominent comparable solutions in the state of the art. Our approach is evaluated at the *Adolfo Suárez-Madrid Barajas* airport, using incoming flights (from 40 different airports) during the first three quarters of 2022, reporting a mean absolute error (MAE) of 2.65 min and a root-mean-squared error (RMSE) of 4.30 min over the entire flight, for all of the routes considered in this study. These results demonstrate that LSTM is a viable approach to ETA prediction in ATM and can surpass other techniques that are the state-of-the-art at this task, such as ensemble and boosting machine learning methods. This paper also demonstrates how a global model can outperform individual models that are specific for a single route. Our experiments show that including trajectories from different routes improves the robustness of the model for each of those routes, with generalized improvements along the whole route.

In summary, this paper makes three main contributions:

- A novel approach to estimating the time of arrival at any point along the flight path, based on surveillance data and taking into account the weather conditions at the destination airport.
- An effective LSTM-based architecture that leverages the similarities between different routes arriving at the destination airport to provide more accurate results than specialized, state-of-the-art individual models.
- A case study of European international flights arriving at Madrid Barajas-Adolfo Suárez (Spain), an airport that has not yet been studied in the literature, despite its high traffic volume.

The rest of the paper is organized as follows. Section 2 gives a broad picture of the estimated time of arrival problem, and Sect. 3 provides the basic background to understand our approach. Section 4 describes the selected data that we use to make ETA predictions using the LSTM-based architecture presented in Sect. 5. Section 6 describes our case study and the process of generating the dataset used in our experiments, which are carefully presented and analysed in Sect. 7. Finally, Sect. 8 presents our main conclusions and devises our lines of future work.

2 Related work

Accurate prediction of the *estimated time of arrival (ETA)* is crucial to reduce costs and environmental impact of flights, and, at the same time, to improve its safety, capacity and efficiency [14]. Moreover, an inaccurate prediction of the ETA of a flight can have a cascading effect which, in turn, may have an impact on the arrivals of other flights that will have to wait until the necessary resources are available for landing. However, due to all the nondeterministic events that can occur during a flight, providing accurate predictions of the ETA is a challenging task.

ETA prediction for commercial flights was originally addressed using deterministic methods [16], based on aircraft performance and physics simulations. The basic

idea of these methods is to compute a reference flight trajectory and then calculate the time needed to fly it. Although effective, these methods are complex and expensive to develop, and their predictions are highly dependent on the considered simulation conditions (weather, traffic congestion, etc.), so predictions will be inaccurate if these conditions do not hold during the flight.

On the other hand, data-driven approaches have gained importance in recent years, due to the increased availability of air traffic-related data and hardware resources capable of running computationally intensive machine learning algorithms. These methods leverage historic data to learn hidden patterns and are better suited to adapt to unseen or rare circumstances, providing more accurate predictions in the uncertain conditions that govern ATM operations. Thus, data-driven approaches generalize better than fine-tuned deterministic ones and are therefore currently the most appropriate choice to address the problem at hand.

Most of the existing data-driven approaches [8–13, 17] build a prediction model for a particular arrival airport, enabling ETA predictions for all incoming flights, regardless of their departure airport. All of them report figures for short-term ETA predictions, when the aircraft is close to the destination airport in terms of distance (mostly between 25 and 100 nautical miles, or NM, away from the airport) or time (between 5 and 60 min before landing), but do no report numbers at the early stages of the flight. In [18], a single model is proposed to predict all flights within a given area, but it is suggested that individual models for each destination airport (or groups of related airports) would be more effective. Ayhan et al. [14] follow this approach and build optimized models for each particular route (i.e. a pair of departure and arrival airports), enabling long-term accurate ETA predictions, at the price of a complex deployment that involves maintaining multiple models (one per route) at the destination airport.

A second aspect to consider is the data used to construct the prediction models. This decision must take into account the many factors that can affect the operation of a flight. *Surveillance information* (mainly ADS-B) is present in most of the proposals, allowing to describe enriched 4D trajectories, where time and situation (latitude, longitude and altitude) are enhanced with other valuable features, such as speed or heading, to characterize the aircraft movement over time. *Weather data* (wind direction and speed, visibility, etc.) is also present in most of the solutions, but they differ in whether they consider this information only at the arrival airport [8, 11, 12, 17, 18], also at the departure airport [19] or throughout the flight [14]. *Flight plan data*, reporting origin and destination airports, the (scheduled and actual) off-block and takeoff times (if the flight has already started), the scheduled arrival time or the expected total duration of the flight, among other features, are also commonly used [10, 11, 13, 14, 17–19]. Finally, *seasonality information* [8, 12, 17, 19] (e.g. the day of the week, the month or the time of the day in which the flight will occur), *congestion information* [12, 14, 18, 19] (traffic density metrics at the airports or airspace levels) or *information about resource management* [8, 11] (e.g. configuration of runways) have proven to be useful for ETA prediction.

The main difference between the existing state-of-the-art solutions lies in the method they used to predict ETAs. Various machine learning methods have been evaluated for such purpose, highlighting bagging ensemble models, such as random

forests (RF) [20] or Extra-Trees (ET) [21], and boosting methods, such as Gradient Boosting Machines (GBM) [22] or Adaptive Boosting (AB) [23]. More recently, Feed-Forward Neural Networks (FFNN) [24] and other deep learning models, such as Long Short-Term Memory (LSTM) networks [15], have been used with varying degrees of success.

Glina et al. [8] propose a RF-based model (called Quantile Regression Forest), which provide short-term predictions (between 3 and 60 NM) with RMSE values between 0.33 and 1.25 min, for flights arriving at Dallas/Fort Worth International Airport (ICAO code, *KDFW*). Kern et al. [18] also use random forests to predict ETAs for domestic routes in the USA and report a MAE reduction of 42.7%, compared to the ETA prediction provided by the Federal Aviation Administration (FAA) system. Kim [17] uses linear and median regression and a nonparametric additive model to predict ETAs for incoming domestic flights at the Denver International Airport *(KDEN)*. In this case, the models were trained on 2010 data and then predictions were made for flights in 2011, reporting a mean absolute deviation of 8.63 min and concluding that departure delays are the most important factor for improving predictions. Dhief et al. [11] compare RF, ET and GBM models, at the Changi Airport *(WSSS)*, concluding that ET performs better than the other methods, and reporting a RMSE of 1.92 min at 100 NM from the destination airport.

Subsequent publications demonstrate the superiority of GBM over bagging methods, at different time horizons. In [13], a GBM-based model is used to predict ETAs for incoming flights to the Malpensa-Milan Airport *(LIMC)*, reporting RMSE values of 175 s and 304 s, at 20 and 60 min from the arrival airport, respectively. Chen et al. [12] compare GBM, RF and FFNN predictions at different distances from the Zurich airport *(LSZH)*, concluding that GBM performed slightly better than RF and reporting RMSE values of 3.16 and 4.75 min, at 45 and 250 NM of distance, respectively. An ensemble "stacked" model is proposed in [10], reporting slightly better figures than GBM (and other methods) for ETA prediction at the entry point of Terminal Manoeuvring Area (TMA) of the Beijing Capital International Airport *(ZBAA)*. Achenbach et al. [19] also propose an ensemble model, which combines GBM and linear regression. This ensemble performs better than its constituent models, reporting effective predictions at long term (RMSE of 5.9 min at departure), for flights flown by the A320 European fleet arriving at two airports. Ayhan et al. [14] make an exhaustive comparison of machine learning methods for ETA predictions on 10 major flight routes in Spain, including LSTM for the first time. In this case, GBM and AB report the best numbers, within 4 min of RMSE on average, regardless of the flight length. It is worth noting that LSTM reports unstable numbers in this evaluation, providing the most accurate prediction for a given route and reporting twice as much error as AB on another. Recently, Ma et al. [9] proposed a spatio-temporal neural network model that reports comparable numbers to a LSTM-based approach, for incoming flights to the ZBAA airport.

Table 1 summarizes the main features of the reviewed approaches: the *scope* of their models and the machine learning *method* they used (*RF*: random forests, *LR*: linear regression, *GBM*: gradient-boosting machines, *AB*: adaptive boosting, *ET*: extra trees, and *LSTM*: long short-term memory neural networks); the *data* used to build these models are then displayed (*Su*: surveillance, *W*: weather, *FP*; flight plans, *Se*:

Table 1 Summary of data-driven approaches for ETA prediction

Approach	Scope	Method	Data						Prediction
			Su	W	FP	Se	C	R	
[8] (2012)	Arrivals at KDFW	RF	✓	✓		✓		✓	3–60 NM before landing
[18] (2015)	US territory	RF	✓	✓	✓		✓		During the flight
[17] (2016)	Arrivals at KDEN	LR		✓	✓	✓		✓	Before departure
[19] (2018)	Arrivals at two airports	Ensemble	✓	✓	✓	✓	✓		Before departure
[14] (2018)	Individual route	AB/GBM	✓	✓	✓		✓		Before departure
[13] (2019)	Arrivals at LIMC	GBM	✓	✓	✓				20–60 min before landing
[12] (2020)	Arrivals at LSZH	GBM	✓	✓		✓	✓		45–250 NM before landing
[11] (2020)	Arrivals at WSSS	ET	✓	✓	✓			✓	100 NM before landing
[10] (2020)	Arrivals at ZBAA	Ensemble	✓			✓			Entry point of the TMA (20–25 NM before landing)
[9] (2022)	Arrivals at ZBAA	LSTM	✓						0–4 min after passing the entry point of the TMA
Our proposal	Arrivals at LEMD	LSTM	✓	✓	✓	✓			15–150 min/25–250 NM before landing

seasonality, *C*: congestion and *R*: resources); and the flight points where the ETA is predicted. It is worth noting that the last row also describes *our proposal* in the same terms, for comparison purposes.

3 Background

Estimating time of arrival based on 4D-trajectory data can be intuitively approached as a sequence modelling problem. Each trajectory is described by multiple sequences of values, where each value depends on the previous values in the sequence. Moreover, these data points have time information associated with them, which allows us to interpret them as a time series problem. As such, the analysis of these data using deep learning can be tackled using different types of Recurrent Neural Networks (RNN). In this section, we succinctly describe how RNN work, focusing later on the Long Short-Term Memory (LSTM) architecture, which is used in this paper.

3.1 Recurrent neural networks

In recent years, methods based on Recurrent Neural Networks (RNN) have shown good performance in time series modelling tasks, provided that they are able to capture temporal dependencies in sequential data [25]. In contrast with traditional feed-forward neural networks, in which each layer passes information only to the next layer, recurrent layers present cycles within them, that is, they have links between the neurons in the layer. This fact enables RNN to have "memory" from the elements that have already been processed. A simple recurrent layer contains a single recurrent neuron that expects a sequence of elements as input. Recurrent neurons contain a *cell state* that changes after processing each input element and is used to process the next input element. The layer iterates on every element in the sequence: at each step, cell state is propagated from the previous iteration and used to process the next element in the sequence. Thus, the layer has "memory" of the elements that were processed before in the input sequence. When all elements in the sequence have been processed, the final output is passed to the next layer.

This structure causes RNNs to form very deep structures that increase the risk of vanishing gradient problem [25], particularly when analysing sequences with long-distance dependencies. Vanishing gradient problem consists on the error becoming too small or zero during the back-propagation step in model training. If errors are zero, the parameters of the model are not updated (their value does not change); this translates into a part of the model is not "learning" correctly, or taking a lot of time to learn long-term dependencies in long sequences. This problem can be handled using different techniques, but Long Short-Term Memory networks in particular have attracted much attention in recent years.

3.2 Long short-term memory

Long Short-Term Memory (LSTM) [15] is a gated recurrent network architecture that ensures error propagation even in deep recurrent layers, allowing the model to have "long-term" memory without the loss of "short-term" memory shown by traditional RNN. Inside a LSTM cell, three multiplicative units are defined that act as gates with different purposes: forget gate, input gate and output gate (see Fig. 1). The forget gate determines the extent to which the output of the previous iteration is used to process the next input element. The input gate controls how much information from the input element will contribute to the hidden state. This gated unit protects the hidden state from perturbations and irrelevant elements in the input sequence. The output gate outputs the most relevant parts of the hidden state, once it has been updated. This helps to filter the information to be passed on to the next iteration, avoiding the propagation of irrelevant information from the current hidden state. When the last element in the sequence has been processed, this output is passed to the next layer in the neural network. During this process, hidden state and output are updated separately, which helps to ensure long-term memory.

LSTM networks have already proven to be effective in dealing with air traffic data in different problems in the ATM field. In [26], LSTMs were combined with convolutional neural networks to predict the aircraft type using ADS-B data. Different aircraft types present particular patterns in their displacement, which can be identified in sequences of surveillance data. LSTM can analyse the progression of the latitude and longitude values, among other features, to classify the flight according to the aircraft type. Beyond RTA prediction, other regression problems have also been tackled with the use of LSTM. Shi et al. [27] proposed an architecture for trajectory prediction based on ADS-B data. Their results showed that the last reported positions of an aircraft, among other features of interest for this task, allow accurate prediction of its future positions.

Gated Recurrent Units, or GRU [28], reduced the number of gates in LSTMs by merging forget and input gates into a single "update gate". This simplification

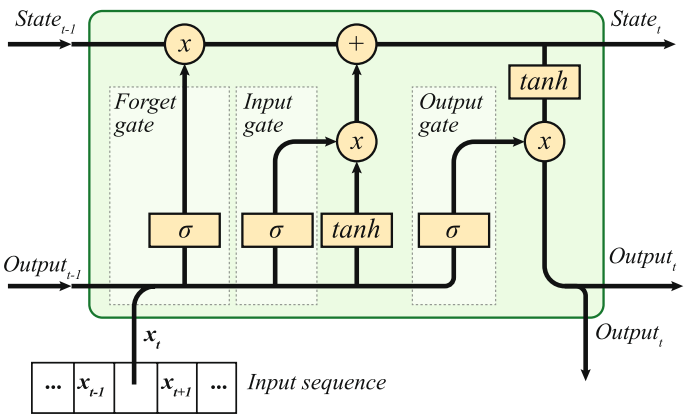

Fig. 1 LSTM unit internal structure

often leads to small improvements in training times since the architecture has fewer parameters to train. However, the performance difference between GRU and LSTM units depends mostly on the problem.

4 Data description

This section describes the data used in our approach, according to the experiences reported in the related work. We organize the corresponding features in three main groups: surveillance, flight plans and weather, which are summarized in Table 2.

4.1 Surveillance data

Surveillance data are used to describe the flight status over time. Each data point has an associated *timestamp* (assigned by the ground receiver at the reception time)

Table 2 Description of the features extracted from data to train the proposed models

Feature	Description	Sample value
	Surveillance	
Latitude	Latitude of a position update	51.477402
Longitude	Longitude of a position update	−0.4745
Altitude	Altitude over the ground at a position update (feet)	225.0
Distance	Haversine distance to LEMD airport (miles)	773.208995
Speed	Horizontal speed with respect to the ground (knots)	141.0
Vertical rate	Speed of climb or descend (feet per second)	2689.0
Track	Angle between aircraft heading and north	267
Operator	IATA code of the operating airline	*BAW*
	Flight plans	
Departure airport	ICAO code of origin airport	*EGLL*
Day of week	Arrival day within a week	5
Time of day	Scheduled arrival period of the day	*evening*
Departure delay	Difference in minutes between planned and actual off-block time	0
	Weather	
Wind direction	Direction from where the wind blows	35.0
Wind speed	Wind speed (knots)	4
Max. temperature	Maximum expected temperature	15.0
Min. temperature	Minimum expected temperature	1.0
Cloud altitude	Altitude of lowest clouds, if any (miles)	30
Visibility	Horizontal visibility in case of fog (miles)	4
Sky status	Qualitative description of sky status	*CAVOK*
	Target variable	
RTA	Remaining time to arrival (seconds)	7048

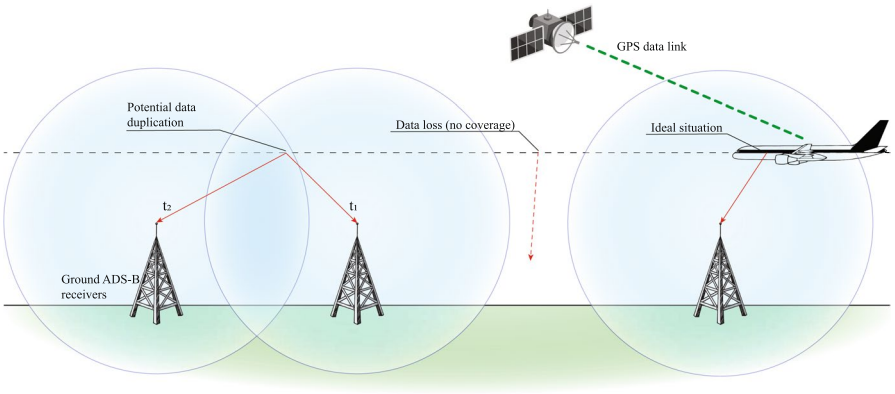

Fig. 2 ADS-B messages are broadcasted from the aircraft and captured by receivers on the ground. Overlapping of coverage areas of different receivers can result in data duplication, while lack of coverage can result in data losses

and contains information about the 3D-position of the aircraft (*longitude*, *latitude* and *altitude*), the *instant speed* (both horizontal and vertical) and the direction the aircraft is heading (*track*). Information about the *airline* that operates the flight is extracted from the callsign (ID code of the flight). Additionally, we calculate a *distance* feature, which is the Haversine distance of the aircraft with respect to LEMD airport.

We use surveillance data provided by OpenSky [29], an open, community-based network of receivers with great coverage of the European airspace. ADS-B messages are broadcasted from the aircraft and received by ground stations, as shown in Fig. 2. Ideally, the broadcasted message should be received by a specific ground station without incident, but two main problems can arise: (i) *data duplication*, when the same message is received by more than one ground station, and (ii) *data loss*, when ADS-B messages are not received due to lack of coverage in a particular region. OpenSky post-processes ADS-B messages to deal with some of these issues and convert them into *state vectors* [5], which preserve the most important surveillance information (identification, position and speed) of the aircraft, and assigns the corresponding flight callsign. However, there are still irregularities in the resulting data. To further reduce them and improve data quality, we perform additional data processing tasks, which are described in Sect. 6.1.

4.2 Flight plan data

Flight plans provide us with scheduling data such as the *expected times of departure and arrival*, the *actual times of departure and arrival* (which are calculated after the end of the flight) or the *departure airport*. We use this information to calculate the *delay of departure*, which is obtained as the difference between the scheduled time and the actual time of the takeoff, and two seasonality features, to exploit daily and weekly time patterns: (i) the *day of week*, in which the flight is

scheduled to end; and (ii) the *time of day* [19], that describes the hour range the aircraft is scheduled to arrive; based in our observations, we consider three periods: morning (7–13 h), evening (13–20 h) and night (20–7 h).

We use the EUROCONTROL Network Manager[1] as source of flight plans. On the one hand, it provides the *Flight Plans* feed, which publishes (i) plans for future flights, including pre-flight scheduling information (such as planned departure and arrival times), airline, origin and destination airports and estimated flight time, and (ii) modifications with respect to a previous version of a flight plan, for flights not yet departed. On the other hand, the *Flight Data* feed provides information during and after the flight has departed, such as the actual departure and arrival times, or any significant changes with respect to the flight plan. We identify the last version of the flight plan (which contains the most up-to-date information) to extract the features explained above.

4.3 Weather data

Weather data are used to characterize the expected weather conditions at the destination airport for the flight's estimated time of arrival. The most relevant features are those related to *wind condition* (direction and speed), because these factors determine the direction of approach the aircraft must take, and influence its speed and manoeuvres. Other selected features describe *temperatures*, *visibility* conditions and *sky conditions*.

In this case, we use forecast reports from weather stations located at the destination airport (TAF, or Terminal Aerodrome Forecast). These reports describe the weather conditions expected in the surroundings of the airport over a period of time (typically, for the next 24 h), which is accordingly subdivided into smaller time periods when changes in conditions are expected. Weather forecasts include data about wind (direction and speed), temperatures, precipitation, icing probability and visibility, many of which may influence landing and take-off operations.

4.4 Target variable

We define a *remaining time to arrival (RTA)* value, which is used as target variable for our model. RTA is the difference (in seconds) between the timestamp of each state vector and the actual landing time of the flight to which it belongs. We use surveillance information to obtain the landing time, even though flight plans provide an end time value, but it usually corresponds to the time at which the pilot was cleared to initiate the landing procedure, several minutes before actually landing.

[1] https://www.eurocontrol.int/network-operations.

5 Architecture

4D trajectories consist on long sequences (several hundreds or even thousands) of flight points, which hide long- and short-term temporal dependencies within the multiple time series they comprise. As stated in Sect. 3, LSTM-based networks are able to learn from long sequences of data, such as 4D trajectories, to capture both short-term and long-term dependencies between the elements in the sequence. This fact motivates our decision to build a LSTM-based neural network to predict the estimated time of arrival of a flight, as shown in Fig. 3. This architecture consists of a single LSTM layer, with a hidden state of dimension n, and a fully connected (FC) layer with one cell and linear activation, to transform the output of the LSTM layer (a vector of length n) into a scalar value, which is the predicted RTA value for the input sequence. There are some considerations for the input sequences to LSTM networks that need to be taken into account to make ETA predictions over 4D trajectories.

Fixed-length sequences LSTM networks expect input sequences of fixed length, but 4D trajectories from different routes, or even trajectories within the same route, may have different lengths (depending on the travel distance and the amount of available surveillance state vectors), so trajectory data need to be transformed into a suitable form. We use a sliding window of length *lookback* (*lb*) to ensure fixed-length sequences: for a trajectory with p state vectors, $(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_p)$, we generate $p - lb$ sub-sequences of length lb (as illustrated in Fig. 4) and label each one with the RTA value corresponding to the last vector in that window. Thus, lb is a determinant parameter to our model: the longer the input sequence, the more clues the model has to make a prediction, but it also increases the complexity and time of the training process.

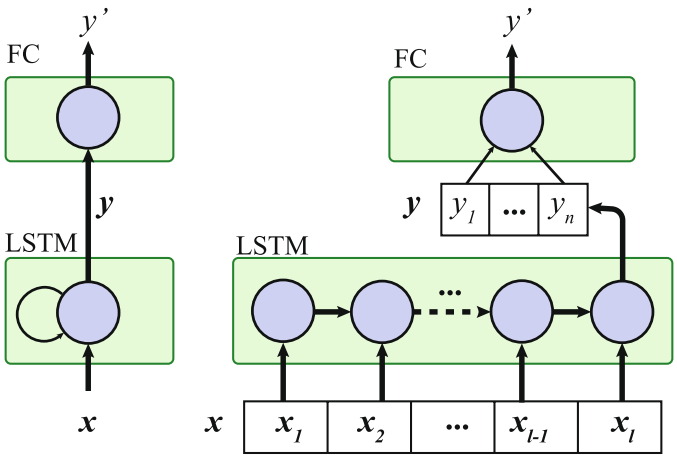

Fig. 3 LSTM model architecture. Unrolled form (right) explicitly represents each timestep in the input sequence

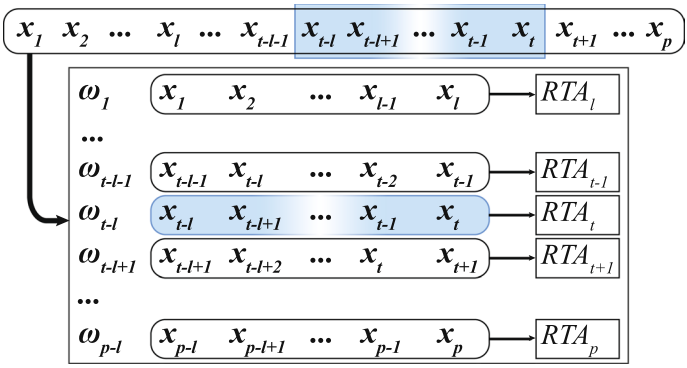

Fig. 4 Extraction of sub-sequences using a sliding window

Sequences with regular periodicity

LSTM networks are not explicitly designed to deal with incomplete or irregular time series data, so irregular patterns in the time dimension, due to missing elements or uneven element spacing, can affect the performance of the predictive model [30]. This is the case with ADS-B surveillance data, and that is due to two main reasons. On the one hand, it is not the aircraft that provides the timestamp when sending the ADS-B messages, but the ground receivers when receiving those messages; as a result, chronologically sorted surveillance data may not be in the same order as they were sent. On the other hand, surveillance coverage is limited in some (mainly maritime) regions, so that messages sent when the plane is flying above these areas are often lost. There are also other problems, such as ADS-B messages captured by multiple receivers, and therefore having different timestamps, or messages where the timing information is inconsistent for different reasons. Figure 2 illustrates these situations. Any ADS-B message broadcasted outside of the combined coverage area of the receiver network will be lost. On the contrary, if there are two or more receivers in range, each of the receivers will capture the message and set its timestamp as the time of reception. If the transmission times (t_1 and t_2 in the figure) are different, then the same ADS-B message is recorded twice with inconsistent time data.

All these situations are addressed to ensure that the input data are evenly distributed over time, with a regular time interval between adjacent elements. First, trajectories are *downsampled* [30] to ensure higher temporal uniformity. The resulting representation can be seen as a summarised trajectory, in which the generalisation of the discovered patterns is improved and potential noise data are removed. However, sampling might discard valuable information if applied too aggressively. Second, sub-sequences that contain gaps of more than a given time threshold are also removed, to minimize irregularities. In this paper, we set this threshold at a maximum of 3 min between adjacent state vectors.

6 Case study

In this work, we focus our case study on the *Adolfo Suárez-Madrid Barajas airport* (ICAO code, *LEMD*), the leading Spanish airport in terms of passenger traffic and the fifth in Europe in 2022. LEMD has four physical runways, arranged as two pairs of parallel runways, that can be used for either takeoff or landing operations, depending on the current runway configuration. LEMD uses two different configurations: *north* (north-facing runways are used for takeoffs and south-east-facing runways are used for landings) and *south* (vice-versa), which are chosen on the basis of weather conditions and available resources. A sample of 200 flights landing at LEMD for January 2022 is illustrated in Fig. 5, which shows prevailing the runway configuration at that moment. The *north* configuration was the most frequently used configuration in the first two months of 2022, but from March onwards, the distribution between configurations became more even due to the change in prevailing weather conditions. Due to this fact, the estimated time of arrival at Madrid-Barajas is even more uncertain. This is because incoming flights may need to execute different approximation manoeuvres depending on the airport's current configuration in order to land on the assigned runway. These manoeuvres may take several minutes and cannot be anticipated, since the landing runway is only assigned and communicated to the pilot when the aircraft is already close to the airport and therefore cannot be used as input to the proposed model.

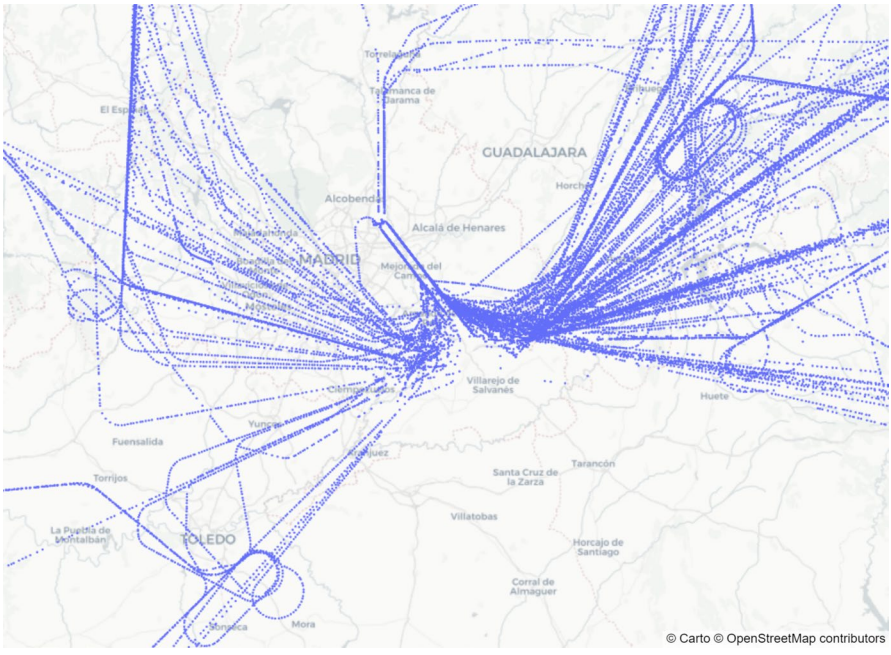

Fig. 5 Sample of 200 flights arriving at LEMD (Jan, 2022)

6.1 Dataset generation

Our study covers incoming flights to LEMD in the first nine months of 2022, i.e. we collected data from the above-mentioned sources from 1 January to 30 September 2022. It is worth noting that there is a significant imbalance in the number of flights from each departure airport to LEMD, which could bias the model in favour of more frequent routes. We choose the 40 most frequent routes to avoid it and limit them to a maximum of 70 trajectories per month. Figure 6 shows the 40 selected airports on the map and indicates, using colours, the number of available trajectories for each one during the study period.

The acquired raw data collection needs to be transformed to ensure high-quality 4D trajectories. This process is performed in three stages that are described as follows.

Trajectory reconstruction This first stage focuses on determining flight trajectories and enriching them with flight plan data. First, we search the Network Manager's Flight Plan feed for flights arriving at LEMD, from all of the 40 selected departure airports. Their identification (aircraft ICAO24 code and flight callsign) and time information (departure and arrival times) are then used to assign ADS-B vectors to individual flights and reconstruct the corresponding trajectory. Surveillance or flight plan data that cannot be joined are discarded at this time, and flights with less than 300 state vectors are also removed. Finally, each vector is enhanced with the latest

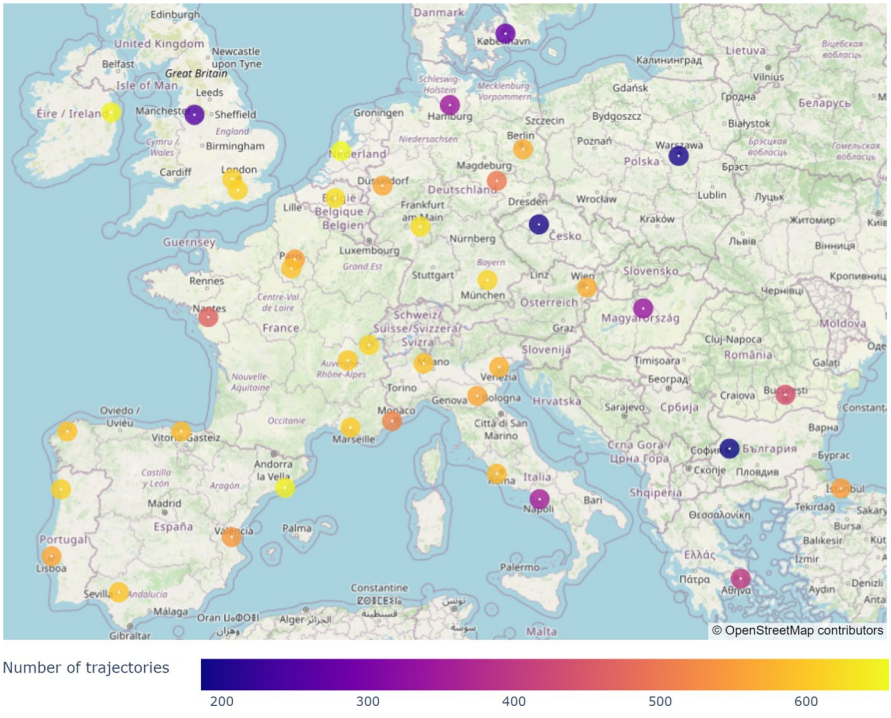

Fig. 6 Airports considered in the study. Marker colour indicates the number of trajectories in our dataset

available weather forecast (the most recent report published before the vector timestamp) valid for the scheduled arrival time.

Quality checking Once reconstructed, the trajectories are cleaned to remove the quality problems inherent in ADS-B: incorrect time information, duplicate data or incorrect field values (altitude, speed, GPS position, etc.). It includes various cleaning operations: elimination of vectors with unrealistic latitude, longitude, altitude or velocity values; sorting of the state vectors within the trajectory in case of misplaced vectors in the time sequence; and linear interpolation of timestamp and altitude values for reordered vectors. Then, RTA values are updated accordingly to the new timestamps (i.e. the difference between its timestamp and the landing time).

Trajectory selection Trajectories with multiple loops during a holding procedure are removed at this stage, as they present a different challenge, due to their complexity and unpredictability [31]. Holding procedures force an aircraft to wait in the air until they are given permission to land at the airport and are characterized by their looping trajectory pattern near the airport. As stated before, these procedures cannot be predicted before the aircraft enters in the TMA. Given the formulation of our problem, holdings create a time shift in the entire trajectory (holding manoeuvres can last several minutes), which leads to an inconsistent computation of the RTA for state vectors of similar nature. In total, 338 trajectories (1.61%) with multiple holding patterns were removed.

Finally, trajectories that exceed the monthly limit set for each departure airport are randomly discarded at this stage. The resulting dataset from the above process consists of 19,633,275 state vectors from 20,560 trajectories, describing flights from the 40 selected airports to LEMD, during the study period. The monthly distribution of trajectories among the airports is shown in Appendix A.

This Appendix presents the distribution of trajectories among the airports considered in the study. The data are divided in months, since the data were sampled to ensure a homogeneous time distribution and an equivalent representation of each route.

6.2 Adaptation to the model

Additional transformations must be applied to the resulting dataset to satisfy LSTM constraints for different experimental configurations. First, trajectories are downsampled to ensure a regular distribution over time of their state vectors. The state vectors of each trajectory are divided into buckets of *SP* seconds (sampling period) according to their timestamp: the first vector of each bucket (in chronological order) is kept, and the rest are discarded. This operation is depicted in the top half of Fig. 7 for the case SP = 15. We also remove all sub-sequences that contain a gap of more than 180 s, between adjacent vectors. Then, the categorical features are transformed into real values using label encoding (i.e. replacing each categorical value with an integer), and all features are normalized into [0,1] range according to Eq. 1, where v is the original value of the feature f, and $v^f_{\min}$ and $v^f_{\max}$ are the minimum and maximum values of the distribution for that feature.

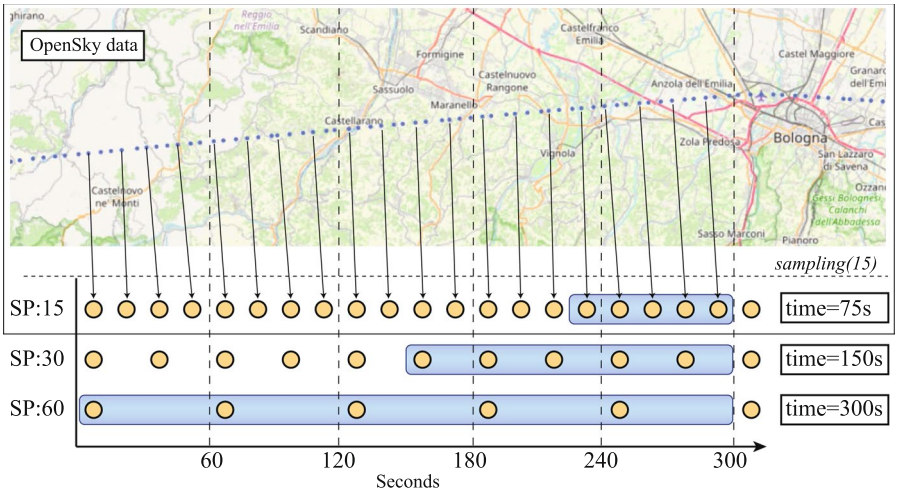

Fig. 7 Example of a downsampled trajectory where the periodicity is increased from $\tilde{5}$ s in the original OpenSky data to 15 s. The values of sampling period and lookback (in the figure, *lookback* = 5) determine the time the model has access to make a prediction

$$v' = \frac{v - v^f_{\text{min}}}{v^f_{\text{max}} - v^f_{\text{min}}} \tag{1}$$

Finally, trajectories are transformed into fixed-length sequences of *lb* elements (lookback), as illustrated in Fig. 4. For each trajectory, we generate all possible subsequences of *lb* neighbouring vectors (using a sliding window of equal length) and assign them the RTA value of the last vector in the window.

It is worth noting that the length of the "flight history" we use to make a prediction is determined by the sampling period and the lookback value. For example, for a sampling of 60 s and a lookback of 32 vectors, the model has access to the last 32 min of flight time to make a prediction. If the lookback is increased to 64 (keeping 60 s of sampling), then the "flight history" taken into account extends to the last 64 min of flight time. This is also true if we increase sampling period to 120 s, while keeping 32 vectors of lookback. However, these choices are not equivalent as the configuration with a lower lookback and higher sampling (32 and 120, respectively) provides less detailed data, because it describes the same time period with half as many state vectors. Figure 7 shows an example with *lookback* = 5. If we sample every 15 s (SP:15), a single window represents the previous 75 s of flight time. However, if the sampling period is 30 s, then each window amounts for 150 s of flight time. While the length of the window is the same in all cases (5 state vectors), the flight time and the level of detail in which the trajectory is described are different for each configuration.

7 Experiments

This section describes the experimental process we have performed to evaluate our proposal and compare it with the state of the art. First, we describe the experimental setup and methodology that we have followed to assess the performance and generalizability of our model. Then, we present our main findings and discuss the results and their contribution to the state of the art.

7.1 Experimental Setup

In the following, we provide a comprehensive description of the experimental setup used in our study. Note that all experiments are conducted on a 4-core Intel Core i5-1035G4 at 1.10 GHz with 16GB RAM. No GPU acceleration was used. The execution environment includes Python 3.9, TensorFlow 2.9.1 and Keras 2.11.0 for LSTM models, and Scikit-Learn 1.1.3 for GBM, AB and RF models.

Dataset The dataset produced by the generation process described in Sect. 6.1 is divided into the usual *train*, *validation* and *test* subsets (containing 72.25%, 12.75% and 15% of the trajectories, respectively). A randomized, stratified approach is applied by distributing trajectories in direct proportion to their monthly and route frequency. In this way, the trajectories are evenly distributed across the three subsets according to the distribution of the original data. Finally, data in each subset are adapted to the particular model configuration, according to the process described in Sect. 6.2.

LSTM model We evaluate two parameters that have a direct impact on the model: *lookback* and *units*. On the one hand, lookback determines the number of individual elements (state vectors) the model expects to process in order to make a prediction. The longer the sequence, the more information the model has to characterize the evolution of the flight. However, longer sequences require more computing power or model complexity to learn long-term, complex patterns from the data. We choose lookback values of 32 and 64, because our preliminary experiments reported poor performance for $l = 4, 8, 16$, and values larger than 64 were discarded as it was not possible to generate windows for some of the shorter routes considered. On the other hand, the number of units determines the dimensions of the internal representation that the model constructs from the input data. The higher this value, the more complex the model and the greater the risk of overfitting. After some preliminary testing, values of 10, 20, and 30 units were chosen.

We assign fixed values to the other hyperparameters of the model: we set the hyperbolic tangent as the activation function; the batch size to 128; the loss function to mean absolute error (MAE), and Adam [32] is used as optimizer. We also experimented with the ReLU activation function, but it caused unstable training processes due to exploding gradient problems. During training, early stopping was used as a regularization measure to avoid overfitting. Models were trained for 30 epochs and the version with the lowest validation loss was selected for evaluation. These configurations were applied to data with a sampling period of 30 and 60 s.

Baseline We consider a baseline that includes the most prominent approaches to ETA prediction in the state of the art: Gradient Boosting Machines (GBM), Random Forest (RF) and Adaptive Boosting (AB). Similar to [14], all models were configured by using the default hyperparameters from their implementation in the Scikit-learn package (version 1.1.3), but reducing the number of estimators from 100 to 50 in all models due to memory constraints. Preliminary tests showed that AB performed poorly compared to other models because the decision trees used as the default base estimator were too shallow. We decided to replace the base estimator of AB with the estimator implemented in RF, reporting better results. None of these models are designed to work with time series, so we provide individual state vectors as inputs instead of the constructed windows.

As an aside, we also tested using GRUs instead of LSTM units, but found no significant differences in model performance, so we excluded them from our study.

Metrics The models are evaluated using MAE (mean absolute error) and RMSE (root-mean-squared error) metrics. MAE is the mean of the absolute values of the differences between each objective RTA value, y_i, and the predicted value, $\hat{y}_i$, for every input i (Eq. 2). RMSE is the square root of the mean value of the squares of the differences between each objective value and the predicted value across all examples (Eq. 3). Due to its linear nature, MAE weights equally each example regardless of its error value. In RMSE errors are squared, so larger errors are weighted more than smaller errors and can be used as a metric of the variance of the error values. The values are provided in seconds to enable direct comparisons.

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{2}$$

$$\text{RMSE} = \sqrt{\frac{1}{\sum}_{i=1}^{n} (y_i - \hat{y}_i)^2} \tag{3}$$

In this paper, we use two types of metrics: (i) *global metrics* to indicate the mean error value across all sequences (examples) in the dataset, regardless of the point of the trajectory in which they are placed; and (ii) *at-time* and *at-distance* metrics, which are used to characterize the prediction error at particular points of the trajectory. These particular points can be selected by time (e.g. the error 60 min before landing) or by distance (e.g. the error at 100 NM from the destination airport).

Note that longer trajectories are subject to greater uncertainty, but "cutting" all trajectories at the same remaining time or at the same distance to the arrival airport allows for fair comparison, regardless of the route they describe. To calculate these metrics, we evaluate the model on the last available sequence at the selected point in the trajectory, provided that it is close enough to the cutting point. We define maximum thresholds of 300 s and 10 NM for the difference between the cutting point and the RTA value and distance of the last state vector in the sequence. Otherwise, the sequence is not taken into account in the evaluation, as it is not representative of the designated point in the trajectory.

Table 3 Global metrics

Model	SP	LB	Time	Units	MAE (s)	RMSE (s)
LSTM				10	224.27	338.52
	30 s	32	16 m	20	220.06	333.95
				30	233.31	341.55
				10	162.90	260.85
	30 s	64	32 m	20	**159.24**	**257.82**
				30	161.04	261.52
				10	218.77	321.59
	60 s	16	16 m	20	218.79	332.58
				30	211.56	320.56
				10	164.18	262.22
	60 s	32	32 m	20	163.60	263.56
				30	161.17	259.91
GBM	30 s	–	–	–	220.45	323.73
AB	30 s	–	–	–	222.41	458.28
RF	30 s	–	–	–	256.92	552.01

7.2 Results and analysis

Table 3 shows global metric results for different model configurations. To make it easier to refer to individual configurations, we will use the notation (SP:X,LB:Y,U:Z) to indicate the model with sampling period of X seconds, lookback of Y vectors, and Z units, respectively. Note that the third column (*Time*) refers to the length (in minutes) of the input sequences, and it is obtained by multiplying the value of SP and LB. The results for the baseline methods are also reported at the bottom of the table using data sampled at 30 s, as in our most prominent configuration.

We first analyse the impact of downsampling by comparing same time values and different samplings. In this case, they report similar numbers for sequences describing either 16 or 32 min of flight time, but SP:30 outperforms SP:60 for longer sequences, reporting a reduction in MAE between 0.13 and 4.36 s. It is also shown that the number of units has little effect on the reported error values, but it is worth noting that more units imply higher training costs. Finally, the lookback value has the greatest impact on the result. By doubling the number of state vectors of the input sequences, the model has access to twice the flight time, which helps to better characterize its current state and provide a more accurate prediction. (SP:60,LB:32,U:30) reduces MAE in 50.39 s compared to (SP:60,LB:16,U:30), and this difference is consistent for 10 and 20 unit models. The gap is even larger for SP:30, where (SP:30,LB:64,U:20) outperforms (SP:30,LB:32,U:20) in 60.82 s in terms of MAE. The smallest MAE (159.24 s) is reported by the configuration (SP:30,LB:64,U:20), although (SP:60,LB:32,U:30) also reports competitive error values, increasing MAE by ≈2 s (161.17 s). In all cases, the RMSE values for each configuration confirm all of these observations.

We promote the configuration (SP:30,LB:64,U:20) for comparison purposes with the baseline models, which are trained on data with the same sample period. As

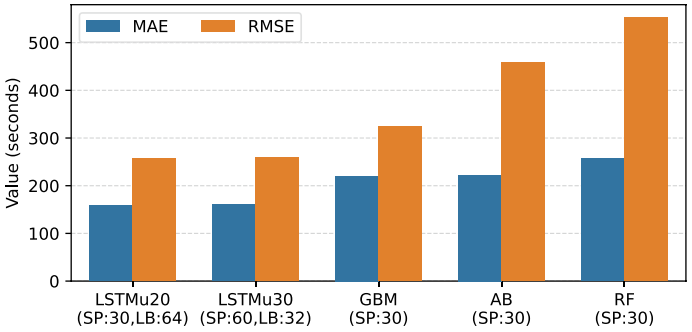

Fig. 8 MAE and RMSE values of baseline models and best LSTM configuration

Table 4 Metrics at-time for LSTM models (time = 32 min). Lowest values (in seconds) for each metric are bolded

MAE at-time	15 min	30 min	60 min	90 min	120 min	150 min
SP:30, LB:64, U:20	**93.80**	**143.50**	**173.92**	**205.12**	264.36	345.03
SP:60, LB:32, U:30	95.84	145.15	175.83	209.12	**255.65**	**326.83**
GBM (SP:30)	153.33	189.21	209.46	253.62	298.01	348.24
AB (SP:30)	113.67	198.19	207.74	251.24	358.89	385.19
RF (SP:30)	191.30	225.64	243.60	320.15	385.22	395.16
RMSE at-time	15 min	30 min	60 min	90 min	120 min	150 min
SP:30, LB:64, U:20	**173.65**	**224.92**	**258.35**	299.46	378.32	477.81
SP:60, LB:32, U:30	175.76	229.11	259.39	**296.10**	**356.93**	473.09
GBM (SP:30)	255.73	299.60	304.55	342.98	398.16	**472.45**
AB (SP:30)	379.49	429.63	308.95	489.61	533.39	521.00
RF (SP:30)	693.32	467.57	543.27	638.97	534.18	525.14

shown in Fig. 8, GBM and AB provide similar results, with a MAE of 220.45 and 222.41 s, respectively. However, GBM achieves a significantly lower RMSE value than AB, indicating that GBM error values have a lower variance and thus its predictions are more consistent. RF performs poorly in both metrics, confirming that boosting approaches are superior to ensemble approaches, as noted in the related work. The LSTM model is superior to all the baselines, improving their best result in terms of MAE by 61.23 s. The study of the RMSE yields the same conclusions, with LSTM providing a result 65.09 s better than the best baseline method, GBM, which highlights the improved stability of LSTM over GBM.

To complete this analysis, Tables 4 and 5 show the values for the at-time and at-distance metrics for our two selected configurations. Accordingly with the global metrics, (SP:30,LB:64,U:20) outperforms (SP:60,LB:32,U:30) on all at-time metrics up to 90 min before landing, although the differences are small, between 2 and 4 s. However, SP:60 model yields better results at 120 and 150 min, with a MAE 8.71

Table 5 Metrics at-distance for LSTM models (time=32 min). Lowest values (in seconds) for each metric are bolded

MAE at-distance	25NM	45NM	60NM	100NM	125NM	250NM
SP:30, LB:64, U:20	**55.76**	**81.77**	**102.85**	**132.61**	137.54	**149.19**
SP:60, LB:32, U:30	58.69	83.72	106.45	133.16	**136.46**	149.62
GBM (SP:30)	103.94	149.69	183.82	204.01	211.04	205.24
AB (SP:30)	69.78	105.11	141.97	187.92	211.32	211.66
RF (SP:30)	127.34	173.51	215.69	248.06	248.70	218.16
RMSE at-distance	25NM	45NM	60NM	100NM	125NM	250NM
SP:30, LB:64, U:20	**98.34**	**138.25**	**167.18**	**215.28**	218.77	**233.83**
SP:60, LB:32, U:30	99.72	140.45	172.21	217.29	**215.09**	234.52
GBM (SP:30)	231.68	282.72	326.47	344.97	358.16	361.29
AB (SP:30)	372.43	396.10	424.19	451.72	474.65	469.91
RF (SP:30)	568.44	780.65	765.67	544.76	515.27	431.92

and 18.20 s lower, respectively. These evaluation metrics measure the performance when most of the remaining flights are still at cruise level, although they exclude most short-range routes (e.g. those corresponding to national flights in Spain). SP:30 has a bigger density of data around the destination airport than SP:60, which may be biasing the model towards the area surrounding the airport in spite of farther parts of the routes. Also, sampling reduces the amount of noise in the data and thus may help the model to generalize better at higher rates when there are not sudden changes in the trajectory. In consequence, SP:30 models might be best fitted to sequences near the airport, where the more detailed representation of the trajectory may benefit the model, while SP:60 performs better on sequences that are farther away from the airport, where there is less variability in the trajectory. This situation holds true for the at-distance metrics, although the differences are generally negligible, in concordance with the global results shown in Table 3.

In all cases, our models significantly improve the results reported by GBM, AB, and RF, both in terms of at-time and at-distance metrics, with the exception of RMSE at 150 min, where GBM performs slightly better than LSTM models.

7.3 Generalization assessment

This section focuses on further evaluating the performance, quality and reusability of our approach, but on a more realistic scenario. For this purpose, we use trajectories that were flown at a later date than the trajectories used to train the models. These trajectories describe incoming flights to Madrid-Barajas (from the 40 selected departure airports) from 1 to 31 October 2022. The resulting dataset (using the generation process described in Sect. 6.1) consists of 2,224 trajectories and 2,512,429 state vectors.

Table 6 reports global MAE and at-time metrics for this scenario, including our best model configurations and baseline methods. The performance deteriorates for

Table 6 Evaluation with future data

Model	MAE	15 min	30 min	60 min	90 min	120 min
SP:30, LB:64, U:20	**200.13**	**101.48**	169.57	215.46	286.24	**337.46**
SP:60,LB:32,U:30	201.62	106.02	**167.43**	**208.54**	**282.08**	344.30
GBM (SP:30)	252.89	142.71	191.18	226.06	289.77	383.67
AB (SP:30)	267.37	121.84	221.11	264.02	299.01	419.57
RF (SP:30)	313.49	272.07	275.76	248.12	336.18	473.83

Global MAE and at-time metrics for the best LSTM configuration and the baseline models are included. Lowest values (in seconds) for each metric are bolded

all models, as expected, but ours still lead the comparison. The LSTM SP:30 model reporting a 25.7% higher global MAE, with an increment of 41 s in absolute terms, compared to the results shown in Table 3. The at-time metrics also increase, but each at a different rate: at 15, 30, 60, 90 and 120 min, the MAE increases by 8%, 18%, 24%, 39.5% and 27.5%, respectively. This analysis is also valid for the LSTM SP:60 model, since its figures are comparable. These results indicate that the model is more robust the closer the aircraft gets to the destination airport. The most likely reason is twofold. On the one hand, the manoeuvres around the airport are standardized, and there is less variability in how the flight should progress. On the other hand, there are much more data on how the aircraft will behave in the surroundings of the destination airport than in any other area of the airspace, so the model may have learned better about this section of the flights. Having more data for each route should help the model to reduce the gap in the generalizability at different time horizons.

LSTM remains the leading model by a wide margin, improving the MAE by 52 s over GBM, which is the most effective state-of-the-art model in this experiment. This comparison also applies for every at-time metric considered, demonstrating LSTM's superiority.

7.4 Individual airport models comparison

We conducted several experiments to assess whether a global approach would be better than training specific models for each individual route, as stated in [18]. In particular, we trained several (SP:30,LB:64,U:20) models using data from each of these individual routes. Each model was trained and evaluated on the subset of the global dataset corresponding to the trajectories that belong to its particular route. Therefore, the global model and each individual model share exactly the same data about the corresponding route. All models were trained for 40 epochs in the same conditions that were used to train the (SP:30,LB:64,U:20) global model, including the partition of the data in train, test and validation: the training data of each of these models were the data from the same route that were used to train the global model. The same holds true for test and validation datasets.

The results of these experiments are shown in Appendix 1 for all departure airports, but we focus on five of them: Frankfurt Main International (EDDF), Germany;

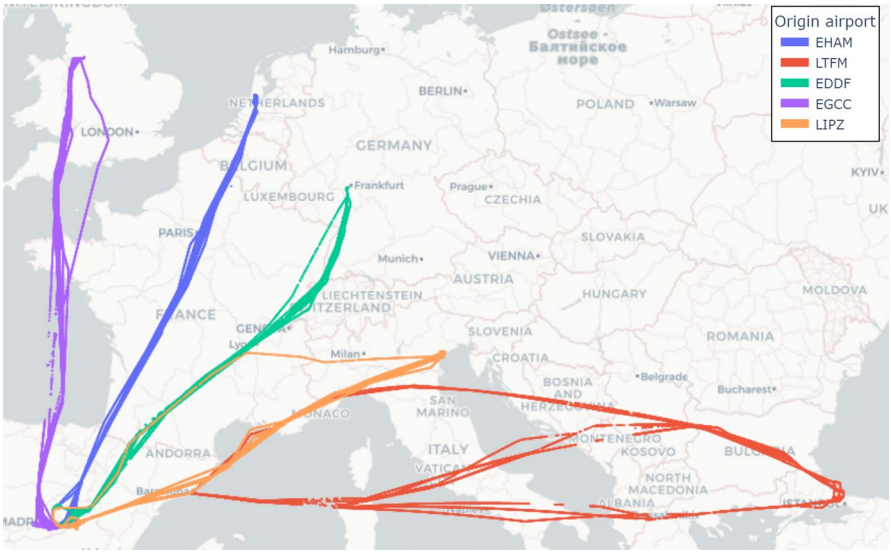

Fig. 9 A sample of the trajectories from the five airports considered in the study of the individual models

Table 7 Individual route model vs. global model approach

Airport	Model	MAE	15 min	30 min	60 min	90 min
EDDF	Global	136.29	79.90	129.24	158.21	187.53
	Individual	222.97	176.10	199.61	239.29	229.86
EGCC	Global	170.68	93.88	176.37	200.84	189.73
	Individual	254.59	259.82	267.79	275.49	236.30
EHAM	Global	155.87	93.98	136.58	179.45	244.70
	Individual	214.70	146.67	157.59	237.75	298.91
LIPZ	Global	147.73	98.33	128.35	179.14	184.72
	Individual	203.36	129.06	195.23	222.24	223.89
LTFM	Global	221.89	149.14	143.61	128.44	161.11
	Individual	269.67	230.73	163.20	167.40	202.82
Mean MAE reduction		61.76	68.57	38.98	57.56	21.65

All the trained models are (SP:30, LB:64, U:20). Units in seconds

Manchester Airport (EGCC), UK; Amsterdam Airport Schiphol (EHAM), Netherlands; Aeroporto Internazionale Marco Polo di Venezia (LIPZ), Italy; and Istanbul Airport (LTFM), which are chosen to cover the main European routes arriving at LEMD airport. A sample of the corresponding trajectories is depicted in Fig. 9. Nevertheless, it is worth noting that similar conclusions can be drawn for routes that follow the same airways to fly to Madrid.

Table 7 shows the results of evaluating global and individual models on the test datasets corresponding to each of the selected routes. The global model performed consistently better on every metric for each of the routes. The bottom of

Table 8 Results of the ablation test

Feature set	MAE	RMSE
Surveillance	208.77	326.09
Surveillance + Flight Plans	206.43	319.23
Surveillance + Weather	215.70	324.33
Full dataset	159.24	257.82

Units in seconds

the table reports the mean improvements observed on these five routes. The largest reductions in MAE can be observed closer to the airport. EDDF and EGCC benefit the most from using the global model: they achieve improvements of 86.7 and 83.9 s in global MAE. In particular, EDDF shows a great improvement at 15 min, with a reduction of MAE of 96 s. These routes have in common that the origin airport is located in the hearth of the European airspace, and thus the synergy with trajectories from other routes, which is observed in the surroundings of the airport, is extended along most of the trajectory. Other routes, such as LTFM, do not have as much path in common with the other trajectories used to train the global model, so they do not benefit as much as the rest of the considered individual routes, with a MAE improvement of 47.8 s. In a middle ground we find EHAM, which still achieves a MAE reduction of almost a minute with the global model.

Provided that global and individual models were trained with the same data from each route, it becomes clear that the global model takes advantage of the availability of data from other routes. The European airspace is structured as a route network, where flights converge on airways, which roughly determine the route that an aircraft must follow to fly to the destination airport. Once in an airway, most flights will behave similarly under similar conditions (aircraft model, weather conditions, etc.). This synergy becomes more apparent closer to the airport, given that flights follow standard procedures in the surroundings of the airport to approach to the runway and perform a landing procedure, so the model benefits from having data from more flights, even if they belong to different routes. Therefore, global models can learn a wider variety of patterns using data from different routes, thereby improving their performance on each route.

These results show that using a global model is better than maintaining multiple models, one per route (i.e. between an origin airport and a destination airport), for three main reasons: (i) the global model can predict the ETA at any point in the flight more accurately than any of the individual models that only model flights for a specific route; (ii) maintaining a single model is less costly than maintaining a large number of smaller models; and (iii) the global approach can improve predictions even on routes where there are fewer flights due to a lower flight frequency or less data availability.

Table 9 Overview of the main results of the works reviewed in Sect. 2

Ap	Scope	Method	Metric	Main results	Our proposal
[8]	KDFW	RF	MAE_{20NM}	58 s	55.76 s
			MAE_{60NM}	75.4 s	102.85 s
[17]	KDEN	LR	MAE_{fut}	8.63 min	3.33 min
			RMSE	12.2 min	5.35 min
[19]	Two airports	Ensemble	MAE	4.31 min	2.65 min
			RMSE	5.9 min	4.29 min
[14]	Indiv. route	AB/GBM	$RMSE_{LECO}$	3.12 min	2.61 min
[13]	LIMC	GBM	$RSME_{60\ min}$	304 s	258 s
[12]	LSZH	GBM	$RMSE_{45NM}$	3.16 min	2.45 min
			$RMSE_{250NM}$	4.75 min	4.03 min
[11]	WSSS	ET	MAE_{100NM}	85–101 s	132 s
			$RMSE_{100NM}$	104–125 s	215 s
[10]	ZBAA	Ensemble	MAE_{25NM}	48 s	55 s
[9]	ZBAA	LSTM	$MAE_{20\ min}$	89.39 s	93.8 s
[18]	US territory	RF	No comparable results were provided		

Each entry includes the used metric, the value reported in the corresponding paper and the value from (SP:30,LB:64,U:20) model

7.5 Ablation test

Our approach combines data from different sources to account for different factors that influence the course of a flight (surveillance, flight plans and weather conditions). We have conducted an ablation test to determine how each factor affects the performance of our approach. In particular, we trained a LSTM model (SP:30,LB:64,U:20) with three different datasets: (i) a dataset containing only surveillance data, which is the core of 4D trajectories; (ii) a dataset enhancing surveillance with flight plan data; and (iii) a dataset combining surveillance and weather data. We replicated the training configuration used for LSTM models in previous experiments, changing only the features used by the model from the original dataset.

The results are shown in Table 8. Surveillance data are the backbone of the predictive power of the model, given that they provide detailed information about the trajectory itself, rather than factors influencing on it. On the one hand, adding *flight plan* features (time of day, departure airport, day of week and departure delay) slightly lowers the MAE results of the model. On the other hand, the *weather* features cause an increase in the error. However, flight plan and weather data in combination help the model to refine the results of using surveillance data by 49.5 s.

7.6 Discussion

Our previous experiments confirm LSTMs models as a good choice for ETA prediction, outperforming other machine learning methods that have been successfully used for the same purpose. The comparison with [14] is particularly interesting, because the behaviour of the LSTMs in their evaluation was rather unstable and, in all cases, their accuracy was lower than that reported by methods such as AB or GBM, contrary to what happens with our proposal.

We will now take a closer look at our results in comparison with the main results of the state of the art, which are summarised in Table 9. It is worth noting that these results may not be directly comparable in quantitative terms, since each proposal analyses different case studies, but it is valuable to consider these numerical results in order to elaborate the following qualitative analysis.

One of the strengths of our proposal is its ability to provide good predictions in both the long and short term (from few minutes to several hours), which is not common in the state of the art, where existing proposals focus on one or the other case. The approaches in [19] and [17] are the most similar to ours, given that they predict ETA during the whole trajectory and using one global model for all traffic incoming into a destination airport. [19] reports MAE/RMSE values (on the test set) of 4.31 and 5.9 min, respectively, using an ensemble model comprised of a linear regressor and several GBM models. In comparison, our LSTM model with the best configuration (SP:30,LB:64,U:20) achieves a MAE/RMSE of 2.65/4.29 min. In [17], the approach based on a nonparametric additive model reached a MAE and RMSE values of 8.63 and 12.2 min, respectively, far from those reported by LSTM. However, their model was trained on data from 2010 and tested on data from 2011, so a fairer comparison may be made with the results reported in Table 6; still, LSTM achieves 3.33/5.35 min when applied on data outside of the training dataset time frame.

Nevertheless, the state of the art is mainly focused on short-term predictions. Muñoz et al. [13] report an RMSE value of 304 s at 60 min before landing, while our best configuration achieves 258 s for the same metric. Wang et al. [10] report 48 s of MAE at 25NM from the destination airport, while we achieve 55 s for the same distance. It is worth noting that this approach trains specific models for each runway, which removes one of the main sources of uncertainty in the short term, and provides only very short-term predictions in the TMA of the destination airport. Strottmann Kern and Medeiros [18] also used Random Forest and reported an reduction of 42.7% in MAE with respect to the ETMF predictions (the ATM system used by the Federal Aviation Administration). However, they do not provide any result that enables direct comparison with the results of our study. Glina et al. [8] applied Random Forests to predict short-term ETA in the surroundings of the airport using 5 days of data. The authors indicate that, during those days, there was only one active configuration in the destination airport and good weather conditions, which may reduce the complexity of the problem. The model is evaluated in

a short-term scenario, at 20 and 60 NM away from the destination airport, with a MAE of 58 and 75 s, respectively. Our LSTM model reaches slightly better results at 20 NM and worse at 60 NM, with 56 and 103 s. This difference may be due to the airport characteristics: in LEMD, the 60 NM radious falls inside the area where the aircraft usually manoeuvres, and therefore is the most difficult part of the flight to be predicted.

Dhief et al. [11] report competitive numbers at 100NM: MAE of 85–101 s and RMSE of 104–125 s, depending on the runway and the model, outperforming ours: 132 and 215 s of MAE and RMSE. The difference between both errors is particularly interesting in our case, because it denotes the presence of outliers (which are more heavily penalized by RMSE than by MAE). This is due to the fact that, in our dataset, we kept trajectories with single-loop holding procedures in our dataset (while in [11] they are removed). However, the exact impact of holdings in our results is yet to be determined and will be addressed as part of our future work. Finally, the GBM model presented in [12] reports 3.16 and 4.75 min of RMSE at 45 NM and 250 NM, while our model reports 2.45 and 4.03 min at the same distances.

Ma and Du [9] proposed a complex model that combines trajectory clustering, convolutional neural networks, LSTM and attention mechanisms to predict ETA in the TMA using surveillance, congestion and weather data. That is, their results are based on the elapsed time since the aircraft enters into the TMA of the airport. As indicated in the paper, the average time between the entering in the TMA and the landing time is approximately 20 min, so their results (MAE of 89.39 s) are comparable to our 15 min at-time metric, which is a slightly higher (93.80 s).

Finally, Ayhan et al. [14] created different models for a sample of individual routes between Spanish airports, and the results supported the authors' claim that LSTM produced unstable predictions. The results of this work may not be directly comparable with our proposal since they make the prediction before the aircraft takes off; however, we include the following observation. In this paper, there is only one route in common with [14]; i.e. the route between LECO (A Coruña Airport, Spain) and LEMD. They reported that AdaBoost performed best at predicting this route, with a RMSE value of 3.12 and 3.71 min for their AdaBoost and LSTM models, respectively. For reference, our global LSTM model yields a RMSE value of 2.61 min, which is the same than that of the individual model for this route. This route was not included in Sect. 7.4 for a deeper analysis because the short length of its trajectories did not allow the calculation of at-time metrics beyond 15 min.

Table 10 Number of total trajectories for each airport and month

Airport	Total	Month								
		01–22	02–22	03–22	04–22	05–22	06–22	07–22	08–22	09–22
EBBR	628	70	70	71	76	72	69	65	65	70
EDDB	571	73	69	72	67	43	47	58	70	72
EDDF	628	69	65	65	67	66	66	74	77	79
EDDH	345	31	35	26	46	43	31	50	41	42
EDDL	563	73	68	59	68	70	42	51	62	70
EDDM	616	66	68	78	68	66	74	67	60	69
EDDP	499	71	66	56	53	41	54	66	44	48
EGCC	252	39	39	32	35	31	14	16	25	21
EGKK	615	67	69	71	60	64	70	73	71	70
EGLL	612	59	68	74	60	61	69	66	82	73
EHAM	657	68	72	81	77	81	69	75	69	65
EIDW	649	71	75	72	69	77	73	76	70	66
EKCH	284	45	41	28	32	28	18	24	35	33
EPWA	200	32	22	23	20	9	13	18	28	35
LBSF	186	23	12	18	28	18	15	22	22	28
LEBB	587	69	69	68	63	65	62	65	69	57
LEBL	647	74	69	74	71	64	75	77	66	77
LECO	595	58	66	70	62	59	67	70	76	67
LEVC	541	58	70	34	54	66	49	73	71	66
LEZL	596	71	64	51	74	69	56	70	72	69
LFLL	601	70	70	58	69	70	61	63	71	69
LFML	606	67	62	68	72	71	64	64	71	67
LFMN	515	49	44	34	71	70	55	65	72	55
LFPG	555	62	56	59	66	54	66	63	60	69
LFPO	592	59	64	76	76	66	66	63	67	55
LFRS	466	32	28	22	72	73	68	58	62	51
LGAV	401	34	20	29	23	21	64	74	66	70
LHBP	338	31	11	28	53	44	37	34	55	45
LIMC	595	70	40	62	75	79	67	81	56	65
LIPE	561	70	16	62	63	65	66	73	69	77
LIPZ	574	66	27	60	64	76	70	72	66	73
LIRF	585	71	70	64	56	57	70	62	69	66
LIRN	356	41	10	35	–	27	56	59	68	60
LKPR	197	18	20	16	28	16	18	20	34	27
LOWW	570	71	18	62	76	67	73	67	71	65
LPPR	611	70	65	62	74	63	69	68	65	75
LPPT	560	58	55	72	54	63	67	68	61	62
LROP	440	56	30	38	62	50	48	48	58	50
LSGG	615	75	68	62	61	68	70	72	66	73
LTFM	551	50	33	50	72	67	69	70	69	71
Total	20,560	2307	1984	2142	2337	2260	2257	2400	2451	2422

Table 11 Results of the evaluation of individual and global models. Units in seconds

		MAE 15	MAE 30	MAE 60	MAE 90	MAE all
EBBR	Individual	130.32	182.54	263.29	410.46	224.42
	Global	107.58	161.66	211.04	376.60	185.19
EDDB	Individual	200.19	206.30	221.80	283.89	254.63
	Global	158.10	177.43	193.63	249.94	215.65
EDDF	Individual	160.81	190.38	235.26	243.22	217.95
	Global	80.98	136.54	166.99	206.64	142.90
EDDH	Individual	156.38	211.76	196.31	219.33	201.22
	Global	73.39	155.37	136.17	157.75	134.54
EDDL	Individual	177.09	249.97	276.45	353.59	274.62
	Global	108.79	214.02	211.17	238.49	196.62
EDDM	Individual	150.57	186.80	235.65	299.33	232.37
	Global	74.07	120.54	143.25	187.94	129.68
EDDP	Individual	111.68	134.04	151.12	224.01	181.29
	Global	83.39	124.43	147.30	200.04	164.77
EGCC	Individual	259.82	267.79	275.49	236.30	254.59
	Global	93.88	176.37	200.84	189.73	170.68
EGKK	Individual	157.17	222.29	200.15	508.26	197.85
	Global	109.61	206.12	189.18	520.55	176.48
EGLL	Individual	163.23	274.88	206.18	467.70	212.00
	Global	99.46	249.22	176.81	303.23	172.41
EHAM	Individual	135.99	148.01	225.03	279.65	205.32
	Global	98.95	141.74	190.47	249.56	163.76
EIDW	Individual	236.97	181.19	216.53	222.39	216.39
	Global	104.65	176.44	174.77	184.62	160.77
EKCH	Individual	147.00	147.99	152.89	203.44	199.87
	Global	81.64	114.08	106.48	151.01	142.20
EPWA	Individual	151.12	177.98	243.64	269.82	293.41
	Global	88.88	112.20	166.09	210.62	204.38
LBSF	Individual	276.49	331.08	370.60	355.86	399.45
	Global	103.98	158.41	158.77	190.08	217.38
LEBB	Individual	–	–	–	–	80.49
	Global	–	–	–	–	54.08
LEBL	Individual	115.01	–	–	–	101.52
	Global	78.41	–	–	–	60.03
LECO	Individual	77.05	–	–	–	79.25
	Global	106.91	–	–	–	83.33
LEVC	Individual	–	–	–	–	339.24
	Global	–	–	–	–	33.11
LEZL	Individual	119.62	–	–	–	66.00
	Global	87.74	–	–	–	64.50
LFLL	Individual	83.09	119.58	396.09	–	110.74
	Global	68.43	106.84	276.51	–	89.39

Table 11 (continued)

		MAE 15	MAE 30	MAE 60	MAE 90	MAE all
LFML	Individual	118.67	145.67	–	–	143.32
	Global	101.13	143.22	–	–	123.06
LFMN	Individual	122.94	185.06	298.67	–	175.55
	Global	99.78	162.31	235.79	–	147.18
LFPG	Individual	84.57	140.32	203.30	–	152.01
	Global	73.03	133.18	172.34	–	126.55
LFPO	Individual	136.66	181.16	216.72	–	185.63
	Global	97.48	137.69	208.40	–	146.33
LFRS	Individual	103.34	120.50	–	–	118.34
	Global	84.64	109.34	–	–	89.53
LGAV	Individual	173.54	194.06	283.32	291.90	283.07
	Global	121.66	167.63	252.61	152.02	186.60
LHBP	Individual	207.05	176.42	216.33	255.85	243.84
	Global	63.92	112.86	120.76	184.79	154.41
LIMC	Individual	148.90	147.00	201.68	–	185.53
	Global	91.72	115.08	144.50	–	125.07
LIPE	Individual	133.63	176.53	221.08	493.67	209.38
	Global	82.96	158.42	186.99	362.37	165.45
LIPZ	Individual	129.06	195.23	222.24	223.89	203.36
	Global	98.33	128.35	179.14	184.72	147.73
LIRF	Individual	131.07	181.82	320.35	343.06	178.70
	Global	114.27	163.34	234.84	216.52	145.39
LIRN	Individual	170.17	180.10	267.35	180.26	201.18
	Global	81.68	115.53	193.88	131.63	121.94
LKPR	Individual	282.28	232.62	282.53	325.89	303.93
	Global	101.44	145.04	188.74	171.69	154.58
LOWW	Individual	136.06	188.05	246.55	326.52	287.22
	Global	92.00	169.82	184.31	211.42	202.01
LPPR	Individual	100.28	–	–	–	79.89
	Global	88.14	–	–	–	70.82
LPPT	Individual	125.42	–	–	–	118.41
	Global	81.08	–	–	–	95.34
LROP	Individual	271.66	201.49	217.56	272.13	292.23
	Global	97.87	101.55	126.45	173.66	176.08
LSGG	Individual	66.51	101.89	141.21	–	103.82
	Global	67.87	106.49	125.93	–	92.78
LTFM	Individual	224.68	148.71	166.42	213.30	277.56
	Global	141.92	137.14	140.70	160.26	214.47

8 Conclusions and future work

We have presented a novel approach to predict the estimated time of arrival using LSTM neural networks, with the aim of taking a further step towards the predictability of air traffic management operations. We have used surveillance, flight plan and weather data to describe incoming flights to the Adolfo Suárez-Madrid Barajas airport, which, to the best of our knowledge, has never been addressed before, despite its considerable importance at national and international level. We have conducted an exhaustive evaluation of different model configurations to better exploit the features of the generated dataset, reporting competitive numbers. Our proposal achieved overall MAE and RMSE values of 2.5 and 4.25 min, respectively, outperforming a baseline consisting of leading models such as RF, GBM and AB, which have reported competitive results for ETA prediction. Besides, we are able to provide accurate predictions at the entire flight, being competitive with solutions specifically designed for short- or long-term predictions.

Our future work will focus on two main lines of research. On the one hand, we consider to enhance our approach with more advanced deep learning techniques, such as attention mechanisms, to improve the detection of relevant information in the input sequences, or convolutional neural networks, to improve the interpretation of positional data, which is key for ETA prediction purposes. On the other hand, we plan to design a more ambitious case study to gain a deeper understanding of the unique challenges of ETA estimation, to identify new sources of uncertainty and to build a portable solution that can be used in different airspace domains.

A Data distribution among routes

This Appendix presents the distribution of trajectories among the airports considered in the study. The data are divided in months, since the data were sampled to ensure a homogeneous time distribution, and an equivalent representation of each route (Table 10).

Results of the individual models

This Appendix presents the results of the evaluation of each individual model, and the corresponding results of the global model, on their respective test datasets, in order to show the improvements explained in the Section 7.5. Where the test dataset had too few examples (less than 10), the results of the evaluation were not included in the table. This happens for some at-time metrics when the route is too short to produce long enough windows, or some windows were discarded due to gaps in the trajectories (windows with gaps longer than 180 s with no data are discarded) (Table 11).

Acknowledgements We acknowledge Boeing Research and Technology Europe (BR &T-Europe) for granting us access to their ADAPT platform.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature. Jorge Silvestre currently holds a predoctoral contract granted by the University of Valladolid and Banco Santander. Jorge Silvestre and Miguel A. Martínez-Prieto were funded by the Spanish Ministry of Science and Innovation, grant PID2020-114635RB-I00. Miguel A. Martínez-Prieto and Anibal Bregon's work was partially supported by the Spanish Ministerio de Ciencia e Innovación under grant PID2021-126659OB-I00.

Data Availability The datasets generated and analysed during the current study are not publicly available due to the licensing terms established for the Network Manager data source between Eurocontrol and our data provider, which does not allow for the distribution of their data.

Declarations

Reproducibility information The code used to process the data and generate the experiments, as well as a README with the necessary instructions for its use, is available in a public repository on Github: https://github.com/JorgeSilvestre/SupComp24-rta-prediction.

References

1. Eurocontrol: Eurocontrol European Aviation Overview (2023). https://www.eurocontrol.int/publication/eurocontrol-european-aviation-overview
2. Eurocontrol: All-causes delays to Air Transport in Europe - Quarter 3 (2022). https://www.eurocontrol.int/publication/all-causes-delays-air-transport-europe-quarter-3-2022
3. ICAO: Annex 2 to the Convention on International Civil Aviation. Rules of the Air. International Civil Aviation Organization (2005)
4. Zhang X, Yuan L, Zhao M, Bai P (2019) Effect of fatigue and stress on air traffic control performance. In: Proceedings of 5th International Conference on Transportation Information and Safety (ICTIS), pp. 977–983. IEEE
5. RTCA: Minimum Aviation System Performance Standards for Automatic Dependent Surveillance Broadcast (ADS-B). Report DO-242A (2006)
6. ICAO: Global TBO Concept v0.11 (2018). https://www.icao.int/airnavigation/tbo/Pages/Why-Global-TBO-Concept.aspx
7. Enea G, Porretta M (2012) A comparison of 4D-Trajectory operations envisioned for NextGen and SESAR, some preliminary findings. In: Proceedings of 28th Congress of the International Council of the Aeronautical Sciences (ICAS), vol. 5, pp. 4152–4165
8. Glina Y, Jordan R, Ishutkina M (2012) A tree-based ensemble method for the prediction and uncertainty quantification of aircraft landing times. In: Proceedings of 10th Conference on Artificial and Computational Intelligence, p. 6
9. Ma Y, Du W, Chen J, Zhang Y, Lv Y, Cao X (2022) A spatiotemporal neural network model for estimated-time-of-arrival prediction of flights in a terminal maneuvering area. IEEE Intell Transp Syst Mag 15(1):285–299. https://doi.org/10.1109/MITS.2021.3132766
10. Wang Z, Liang M, Delahaye D (2020) Automated data-driven prediction on aircraft estimated time of arrival. J Air Transp Manag 88:101840. https://doi.org/10.1016/j.jairtraman.2020.101840

11. Dhief I, Wang Z, Liang M, Alam S, Schultz M, Delahaye D (2020) Predicting aircraft landing time in extended-TMA using machine learning methods. In: Proceedings of 9th International Conference for Research in Air Transportation (ICRAT), p. 9
12. Chen G, Rosenow J, Schultz M, Okhrin O (2020) Using open source data for landing time prediction with machine learning methods. In: Proceedings of 8th OpenSky Symposium, vol. 59, p. 5. https://doi.org/10.3390/proceedings2020059005
13. Muñoz A, Scarlatti D, Costas P (2019) Real-time estimated time of arrival prediction system using historical surveillance data. In: Proceedings of 45th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), pp. 174–177. https://doi.org/10.1109/SEAA.2019.00035
14. Ayhan S, Costas P, Samet H (2018) Predicting estimated time of arrival for commercial flights. In: Proceedings of ACM 24th International Conference on Knowledge Discovery and Data Mining (SIGKDD), vol. 18, pp. 33–42. https://doi.org/10.1145/3219819.3219874
15. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9:1735–80. https://doi.org/10.1162/neco.1997.9.8.1735
16. Murphy J, Reisman R, Clayton J, Wright R (2003) Physics-based and parametric trajectory prediction performance comparison for traffic flow management. In: AIAA Guidance, Navigation, and Control Conference and Exhibit, p. 11. https://doi.org/10.2514/6.2003-5629
17. Kim MS (2016) Analysis of short-term forecasting for flight arrival time. J Air Transp Manag 52:35–41. https://doi.org/10.1016/j.jairtraman.2015.12.002
18. Strottmann Kern C, de Medeiros IP, Yoneyama T (2015) Data-driven aircraft estimated time of arrival prediction. In: Proceedings of 9th Annual IEEE Systems Conference (SysCon), pp. 727–733. https://doi.org/10.1109/SYSCON.2015.7116837
19. Achenbach A, Spinler S (2018) Prescriptive analytics in airline operations: Arrival time prediction and cost index optimization for short-haul flights. Oper Res Perspect 5:265–279. https://doi.org/10.1016/j.orp.2018.08.004
20. Breiman L (2001) Random forests. Mach Learn 45(1):5–32. https://doi.org/10.1023/A:1010933404324
21. Geurts P, Ernst D, Wehenkel L (2006) Extremely randomized trees. Mach Learn 63(1):3–42. https://doi.org/10.1007/s10994-006-6226-1
22. Friedman JH (2001) Greedy function approximation: A gradient boosting machine. Ann Stat 29(5):1189–1232. https://doi.org/10.1214/aos/1013203451
23. Freund Y, Schapire RE (1997) A decision-theoretic generalization of on-line learning and an application to boosting. J Comput Syst Sci 55(1):119–139. https://doi.org/10.1006/jcss.1997.1504
24. Bebis G, Georgiopoulos M (1994) Feed-forward neural networks. IEEE Potentials 13(4):27–31
25. Hochreiter S (1998) The vanishing gradient problem during learning recurrent neural nets and problem solutions. Int J Uncertainty Fuzziness Knowl-Based Syst 6(02):107–116
26. Bolton S, Dill R, Grimaila MR, Hodson D (2023) ADS-B classification using multivariate long short-term memory-fully convolutional networks and data reduction techniques. J Supercomput 79(2):2281–2307
27. Shi Z, Xu M, Pan Q, Yan B, Zhang H (2018) LSTM-based flight trajectory prediction. In: 2018 International Joint Conference on Neural Networks (IJCNN), pp. 1–8. IEEE
28. Cho K, van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, Bengio Y (2014) Learning phrase representations using RNN encoder for statistical machine translation. In: Proceedings of 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1724–1734. https://doi.org/10.3115/v1/D14-1179
29. Schäfer M, Strohmeier M, Lenders V, Martinovic I, Wilhelm M (2014) Bringing up OpenSky: A large-scale ADS-B sensor network for research. In: Proceedings of 13th International Symposium on Information Processing in Sensor Networks (IPSN), pp. 83–94. https://doi.org/10.1109/IPSN.2014.6846743
30. Weerakody PB, Wong KW, Wang G, Ela W (2021) A review of irregular time series data handling with gated recurrent neural networks. Neurocomputing 441:161–178. https://doi.org/10.1016/j.neucom.2021.02.046
31. Dhief I, Alam S, Lilith N, Mean CC (2022) A machine learned go-around prediction model using pilot-in-the-loop simulations. Transp Res Part C: Emerg Technol 140:103704. https://doi.org/10.1016/j.trc.2022.103704
32. Kingma DP, Ba J (2017) Adam: A Method for Stochastic Optimization. arXiv. arXiv:1048550/arXiv.1412.6980