IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 13 May 2024, accepted 30 May 2024, date of publication 28 June 2024, date of current version 29 November 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3415419

RESEARCH ARTICLE

Multi-Route Aircraft Trajectory Prediction Using Temporal Fusion Transformers

JORGE SILVESTRE[1], PAULA MIELGO[1], ANIBAL BREGON[1], MIGUEL A. MARTÍNEZ-PRIETO[1], AND PEDRO C. ÁLVAREZ-ESTEBAN[2]

[1]Department of Computer Science, University of Valladolid, 40005 Segovia, Spain
[2]Department of Statistics and Operational Research, University of Valladolid, 47011 Valladolid, Spain

Corresponding author: Jorge Silvestre (jorge.silvestre@uva.es)

The work of Jorge Silvestre was supported by the 2019 Call for Predoctoral Contracts granted by the University of Valladolid and Banco Santander, and by the Spanish Ministerio de Ciencia e Innovación under Grant PID2020-114635RB-I00. The work of Paula Mielgo was supported by the 2023 Call for Predoctoral Contracts granted by the University of Valladolid and Banco Santander. The work of Anibal Bregon was supported in part by Spanish Ministerio de Ciencia e Innovación under Grant PID2021-126659OB-I00. The work of Miguel A. Martínez-Prieto was supported in part by Spanish Ministerio de Ciencia e Innovación under Grant PID2020-114635RB-I00 and Grant PID2021-126659OB-I00. The work of Pedro C. Álvarez-Esteban was supported in part by the European Union under Grant PID2021-128314NB-I00, funded by MCIN/AEI/10.13039/501100011033/FEDER.

ABSTRACT Trajectory prediction plays a key role in modern air traffic management. The ability to predict the future position of aircraft in flight allows for greater predictability, safety and efficiency. In recent years, recurrent neural networks, and particularly LSTM (Long-Short Term Memory), have been successfully applied (alone or in combination with other kinds of network) to this problem. However, the criticality of the supervision of these operations and the difficulty of predicting trajectories in high density traffic zones, such the Terminal Area around the airports, require high accuracy methods that takes into account all factors involved in these operations. In this paper, we propose an architecture based on Temporal Fusion Transformer (TFT) for multi-route, long-term trajectory prediction using surveillance data (Automatic Dependent Surveillance - Broadcast, ADS-B). We conduct our experiments on the case study of the Madrid Barajas-Adolfo Suárez airport (Spain), using nine months worth of data. In particular, we focus on predicting the next 150 seconds at any point in the trajectory for flights arriving at this airport. Compared with other LSTM networks developed in this work, TFT provides an increased accuracy for 2D positioning, with mean absolute errors of 0.0091 and 0.0104 degrees for latitude and longitude, respectively, in the Terminal Area of the destination airport. These results have been shown to be competitive with, or even superior to, more consolidated approaches based on LSTM networks that have been proposed for single route, short-term predictions.

INDEX TERMS LSTM networks, temporal fusion transformer, air traffic management, trajectory prediction.

I. INTRODUCTION

Intelligent Transportation Systems have gained much attention in the latest decades, thanks to the unprecedented growth in global mobility. The volume and complexity of these movements have enforced the development of advanced management processes, mostly reliant on monitoring data via in-vehicle sensors and data analysis techniques. In this context, *Air Traffic Management* (ATM) constitutes one of the main challenges due to its particular characteristics and its current importance in the global transportation system. In particular, the area near the airports, often referred as *Terminal Maneuvering Area* or *Terminal Control Area* (TMA or TCA, respectively), is a highly complex environment to manage, due to the concentration of aircraft and the safety concerns that may arise during the maneuvers required by the landing and takeoff operations. Furthermore, a diverse set of factors, often very difficult to predict and analyse in real-time operations, may cause deviations from the original flight plan. As a consequence, a great degree of uncertainty is still present in ATM operations, and all of them are performed under intense supervision and continuous monitoring, and

The associate editor coordinating the review of this manuscript and approving it for publication was Ángel F. García-Fernández.

subject to a continuous, real-time decision making process, aiming at detecting and resolving any conflicts in the airspace in safe conditions.

Surveillance systems are one of the most important resources to assist air traffic controllers, by providing them with the states of the aircraft throughout the flight. ADS-B (*Automatic Dependent Surveillance-Broadcast*) [1] 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 recorded by the vehicle. Mandatory for aircraft operating commercial flights in the world's major airspaces, ADS-B equipment plays a key role in introducing the concept of *Trajectory Based Operations* (TBO) [2] into intelligent ATM systems. TBO goes 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 [3]. 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. This detailed description also supports advanced management tasks, such as airspace capacity management, enhanced tactical planning and improved safety assessment.

Trajectory prediction plays a fundamental role in all of these tasks, specially during the tactical phase of the flight (shortly before and during the flight) [4]. Traditionally, trajectory prediction has been addressed using kinematic model-based approaches. However, these models do not take into account the dynamics of the aircraft and the actual context in which the flight takes place, especially when the time dimension is considered. Delays, adverse weather conditions, diverted flights and complex interactions in high-density traffic areas (such as around departure and arrival airports) are all factors that have a profound effect on how a flight unfolds. In this context, data-driven methods can integrate all of these factors in the decision-making process. These methods rely on data describing the past states of the aircraft and the context in which the flight takes place (weather conditions, airspace congestion, etc.), to accurately predict the future states of the aircraft. To make these predictions, data-driven models learn from historical data that provide an exhaustive description of how flights take place depending on the flight conditions.

This paper explores how 4D trajectories can be used to improve predictions of in-flight aircraft trajectories, following a data-driven approach. Trajectory prediction enables better predictability in airspace operations, supporting the air traffic controllers to know in advance the future positions of all aircraft, and facilitating the timely detection of conflicting or potentially-risky situations. The ability to predict the future state of the airspace also allows better resource allocation, resulting in more effective and efficient air traffic management. In our approach, flight trajectories are modeled as time series (including the four dimensions mentioned above and some other features from surveillance and flight plan data), and a deep learning model is trained to make trajectory predictions. In particular, we propose two architectures that have been applied before to approach sequence-to-sequence problems with multivariate time series data: *Long-Short Term Memory* (LSTM) neural networks [5], and *Temporal Fusion Transformer* (TFT) [6]. LSTM have received much attention in trajectory prediction, as described in Section III, and their capabilities have been repeatedly proven. TFT has not been used before to predict aircraft trajectories, to the best of our knowledge. However, TFT should be particularly appropriate to perform trajectory prediction, given its ability to combine time series data (such as aircraft trajectory data) with additional features that help to improve the time series interpretation by providing the relevant context (such as weather conditions or the airspace situation). We evaluate our approach at the *Adolfo Suárez-Madrid Barajas* airport, using incoming flights (from 40 different airports) during the first three quarters of 2022. The results show that TFT can predict the next 10 states (equivalent to the next 150 seconds) on a flight with a mean absolute error (MAE) of 0.0133 and 0.0170 degrees, and a root mean squared error (RMSE) of 0.285 and 0.0536 degrees for latitude and longitude, respectively. These results demonstrate that TFT are a viable approach to trajectory prediction in ATM, and can outperform other state-of-the-art techniques for this task, such as LSTM-based architectures.

Therefore, this paper makes two main contributions:

- An approach to long-term flight trajectory prediction at any point in the flight that uses data from different routes, in contrast with other approaches that tackle either short term predictions, are applicable to a single route or focus only on the area around the destination airport.
- An effective implementation of this approach using LSTM and TFT networks, that achieve competitive results in this multi-route, multi-step, full-trajectory scenario. The proposal is evaluated on flights arriving at Madrid Barajas-Adolfo Suárez Airport (Spain) from 40 European airports.

The rest of the paper is organized as follows. Section II provides the basic background to understand our approach, and Section III gives a broad picture of the state of the art on aircraft trajectory prediction. Section IV explains the implemented architectures, and Section V describes the selected data used to develop and evaluate our models. Section VI describes our case study, the methodology behind our experiments and their results. Finally, Section VII presents our main conclusions, and devises our lines of future work.

II. BACKGROUND

Predicting trajectories from 4D-trajectory data can be approached intuitively as a sequence modelling problem. Each trajectory is described by several sequences of values, where each value depends on the previous values in the sequence. In addition, these data points have time information associated with them, which allows us to interpret them as a time series problem. Therefore, the analysis of 4d-trajectories using deep learning can be tackled using architectures based on *Recurrent Neural Networks* (RNN), which excel on processing this kind of data, or *attention-based architectures*, which have proven to be effective at sequence modelling.

In this section, we briefly describe how the LSTM and TFT architectures work to motivate their use in our approach, and introduce the different strategies that can be applied to predict sequences of data.

A. LONG SHORT-TERM MEMORY

In recent years, methods based on Recurrent Neural Networks (RNN) have shown good performance in time series modelling tasks, given that they are able to capture temporal dependencies in sequential data [7]. In contrast with traditional feed-forward neural networks, recurrent layers present cycles within them, that is, they have links between the neurons in the layer. This structure causes RNNs to form very deep structures that increase the risk of vanishing gradient problem [7], particularly when analyzing sequences with long-distance dependencies. The vanishing gradient problem consists on the error becoming too small or zero during the back-propagation step in model training, which hinders the network training.

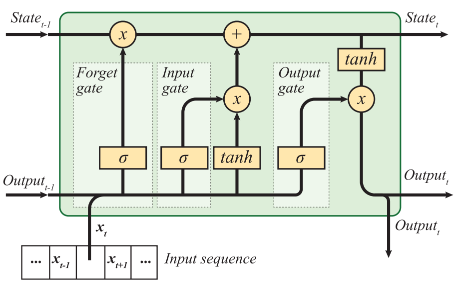

FIGURE 1. **LSTM unit internal structure.**

Long Short-Term Memory (LSTM) [5] 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. Three multiplicative units are defined within an LSTM cell, which act as gates with different purposes (see Figure 1): (i) The *forget gate* determines the extent to which the output of the previous iteration is used to process the next input element. (ii) 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, and (iii) 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 and output states are updated separately, helping to ensure long-term memory.

B. TEMPORAL FUSION TRANSFORMER

Attention mechanisms [8] help models to discover implicit patterns in sequences of data by identifying which elements are more important for performing a task. In this way, the model can modulate the importance of each element in the sequence by assigning weights to them, and focus more on the most relevant elements.

Transformer [9] is an architecture that replaces recurrence by attention in deep neural networks, with great success in tasks such as automatic translation. With the removal of recurrence, which enforces a sequential processing of the input sequences, Transformer allows for parallelization during the training process, although it loses the notion of the order of the elements within the input sequence. Transformer relies on three key concepts: self-attention, multi-head attention and the positional encoding of the elements in the input sequence. *Self-attention* takes into consideration the importance of each element in the context of the rest of the elements in the input sequence, rather than using a predefined set of keys. *Multi-head attention* allows for different interpretations of the input sequence by projecting the query vectors into different representation subspaces. Finally, *positional encoding* allows the model to account for the order of the elements in the sequence.

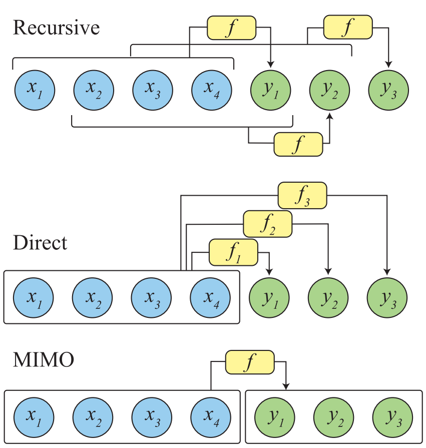

FIGURE 2. **Description of prediction strategies.**

Temporal Fusion Transformer (TFT) [6] combines the Transformer architecture with recurrent layers to perform multi-horizon forecasting in sequence-to-sequence prediction problems. TFT defines three different input types: static, known and observed; to account for the different types of information that intervene in forecasting problems: *static inputs* are data that is constant and known for any considered timestep; *known inputs* are time-varying inputs that are known beforehand; and *observed inputs* are time-varying past data.

C. SEQUENCE PREDICTION STRATEGIES

Sequence prediction in time series can be approached using different strategies. Taieb et al. [10] identified five strategies to make multi-step predictions in time series, the three main ones are illustrated in Figure 2.

1) RECURSIVE

The recursive strategy defines a single-output, one-step ahead model to predict the next timestep: $y_{m+1} = f(x_1, \ldots, x_m)$. In order to predict further timesteps, the previous predictions are appended to the input window; therefore, prediction error is propagated through successive predictions.

2) DIRECT

The direct strategy defines a single-output model specifically for ea ch time horizon that should be predicted. This method avoids error propagation, but implies a conditional independence between successive predictions: $(y_{m+1}, \ldots, y_{m+p}) = (f_1(x_1, \ldots, x_m), \ldots, f_p(x_1, \ldots, x_m))$.

3) DirRec

The DirRec strategy consists on predicting blocks of timesteps using a direct strategy, and appending these predictions to the input sequence, to predict the next block recursively.

4) MIMO

The MIMO (Multi-input multi-output) strategy defines a multi-output model to predict the next p timesteps. This approach conserves the stochastic dependence between the elements in the predicted sequence and avoids the error propagation, but has a reduced flexibility given that only one model is used for all time horizons: $(y_{m+1}, \ldots, y_{m+p}) = f(x_1, \ldots, x_m)$.

5) DIRMO

Finally, DIRMO combines Direct and MIMO strategies: multiple multi-output models are defined for different blocks, each of them calculated in a MIMO manner.

III. RELATED WORK

LSTM networks have been extensively applied to approach sequence prediction tasks, whether alone or combined with different techniques to improve the natural capabilities of these neural networks. Shi et al. [11] used LSTM to predict all aircraft trajectories in an airspace sector using surveillance data obtained from ADS-B sources. Their approach was later improved using LSTM networks with constraints [12], which force the model to take into account different aspects of the kinematics and behaviour of an aircraft depending on the flight stage. With this new proposal, they provided the best performance of all the studied LSTM-based approaches, with a mean absolute error (MAE) of 0.0050 and 0.0105 degrees for latitude and longitude predictions, and a MAE of 9.96 feet for altitude. However, these results correspond with a single route (that is, flights between two particular airports), in contrast with other works where multiple routes can be predicted with the same model. Zeng et al. [13] also applied LSTM networks to model the aircraft dynamics in a complex scenario, such as the environment around an airport, by analyzing landing and takeoff operations at the Guangzhou airport. More recently, Sahadevan et al. [14] successfully applied bidirectional LSTM to leverage both backwards and forward dependencies in the trajectories time series. They evaluated their approach on a single direction from a single route (from Chhatrapati Shivaji Maharaj airport (VABB) to Kempegowda airport (VOBL)), and reported MAE values of 0.0206 and 0.0160 degrees, and 33.75 feet, for latitude, longitude and altitude, respectively.

Subsequent work has introduced various elements to enhance the LSTM capabilities. For instance, Convolutional Neural Networks (CNN) were combined with LSTM to exploit the spatial aspect inherent in trajectory data. Ma and Tian [15] proposed a hybrid model where spatial and temporal inputs are processed sequentially using CNN and LSTM networks. Each component is responsible for extracting different patterns in order to characterize better the current and future state of the aircraft. The presented case study is similar to [14] in terms of characteristics (single direction on a single route) and results. Shafienya and Regan [16] proposed CG3D, a more complex architecture that combines conventional CNN, GRU and 3D CNN to analyze the different types of information from surveillance data. GRU (Gated Recurrent Units) [17] are similar to LSTM, but with a simpler internal structure. As in other proposals, spatial and temporal features are processed separately by CNN and GRU networks, respectively; however, they also add a 3D-CNN network that combines both sets of features, which in combination with the CNN-GRU defines the proposed CG3D architecture. Their results report a global MAE of 0.1176 for all predicted features.

More recently, attention mechanisms have gained prominence in sequence prediction problems, and therefore have also been applied to trajectory prediction. Jia et al. [18] added self-attention layers after LSTM layers to refine their output by promoting their main features and improving their prediction accuracy. LSTM have also been combined with physical simulation models [19], where the LSTM network processes dynamics information about the aircraft (e.g. surveillance data) and its output is used to refine the result of the RM-IMM estimation algorithm. This approach yielded good results in terms of predicting altitude (MAE of 2.94), although it is outperformed by other works given that latitude and longitude are predicted with a MAE of 0.0373 and 0.0397 degrees, respectively.

With respect to the implemented prediction strategy, most of the analyzed works aim to predict only the next state update given the m last updates: $y_{m+1} = f(x_1, \ldots, x_m)$. However, this approach has limited use given that the prediction is too short-term and immediate, so many of them extend this approach to perform multiple timestep-ahead predictions in a recursive manner. Sahadevan et al. [14] studied the degradation of the prediction accuracy with the number of predicted timesteps when following a direct strategy. The results are best at 1 timestep for all features considered (latitude, longitude, altitude and time), after which the results rapidly deteriorate as the predicted sequence lengthens. Similar conclusions can be drawn when using a recursive strategy, as demonstrated in [19], due to the prediction error propagation. Reference [16] is the only work that proposes a MIMO approach, similar to ours, where 5 timesteps are predicted based on the previous 100 timesteps.

With regard to the context in which each study is carried out, most of the analyzed proposals focus on the prediction of trajectories on individual routes [11], [14], [15]; i.e. flights between two particular airports, a specific airspace sector [20] or at the TMA of an airport [13], [16], [19], which may include both landing operations, take-off operations, or both. In terms of the inputs for the prediction models, all previous work uses surveillance data (mainly ADS-B data) such as latitude, longitude and altitude. Most of them also account for the speed of displacement (vertical or horizontal) and the heading of the aircraft, which have demonstrated to be determinant in trajectory prediction. Zeng et al. [13] also utilize the aircraft type, and the speed and acceleration of change of the considered features.

IV. 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 Section II, 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 sequences describing the next states of the aircraft based on the last available states. This architecture consists of a single LSTM layer, with a hidden state of dimension n, and a fully connected (FC) layer to transform the output of the LSTM layer (a vector of length n) into an interpretable trajectory prediction. In this work, this prediction include multiple future states (i.e. timesteps) characterized by three features: latitude, longitude and altitude, in order to determine the short-term future positions of the aircraft. In particular, we aim to predict the next 10 timesteps (i.e. 150 seconds of flight time). The predictions made by LSTM networks are sometimes unstable, and describe maneuvers that are unrealistic for an aircraft in the air (such as a jagged paths that oscillate around the true trajectory followed by the aircraft). We experimented with a LSTM-based architecture with two additional FC layers after the LSTM layer to apply a quadratic polynomial interpolation as a curve smoothing operation [21] on the output of the network. Thus, LSTM layers are responsible of extracting and interpreting time-dependent patterns, and FC layers should transform this interpretation into more feasible trajectory predictions.

We also applied the original TFT architecture [6] on 4D trajectory data to explore its capabilities at the trajectory prediction task. While the architecture also involves LSTM, the multi-head attention mechanisms should extract the key information from the output of LSTM layers to further improve the trajectory predictions.

V. DATA DESCRIPTION

This section describes the data used in our approach, as well as the necessary means to obtain them. The selected features are summarized in Table 2.

A. DATA SOURCES

ATM is a complex field of study in multiple aspects, many of which are not covered in any publicly accessible data source. Therefore, we need to extract the necessary data from diverse data sources, in order to integrate and refine them for the purposes of this work. In particular, we utilize surveillance and flight plan data.

1) SURVEILLANCE DATA

Surveillance data describe the flight state over time according to the considered dimensions. Each data point has an associated *timestamp* (assigned by the ground receiver at the reception time), and contains information about the 3D-position of the aircraft (*longitude*, *latitude* and *altitude*), the *horizontal speed*, the *vertical rate* (the rate at which the aircraft climbs or descends) and the direction the aircraft is heading (*track*). Information about the *aircraft* and the *airline* that operates the flight can be extracted from the transponder code (hexadecimal unique code) and the *callsign* (ID code of the flight), respectively. Additionally, we calculate a *distance* feature, which is the Haversine distance of the aircraft with respect to the destination airport.

We use surveillance data provided by OpenSky [22], 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. OpenSky post-processes ADS-B messages converting them to *state vectors* [1], which preserve the most important surveillance information (identification, position and speed) of the aircraft (sampled every 5s), and assign the corresponding flight callsign. However, there may be several irregularities in the resulting data. To further reduce them and improve data quality, we perform additional data processing tasks, described in the Section V-B.

2) FLIGHT PLAN DATA

Flight plans provide with scheduling data such as the *flight ID* data, and the *departure* and *destination airports*. We use this information to filter the relevant data for the selected case study, and as input feature in the case of the origin airport.

We use the EUROCONTROL Network Manager[1] as the source for flight plan data. In particular, it provides the

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

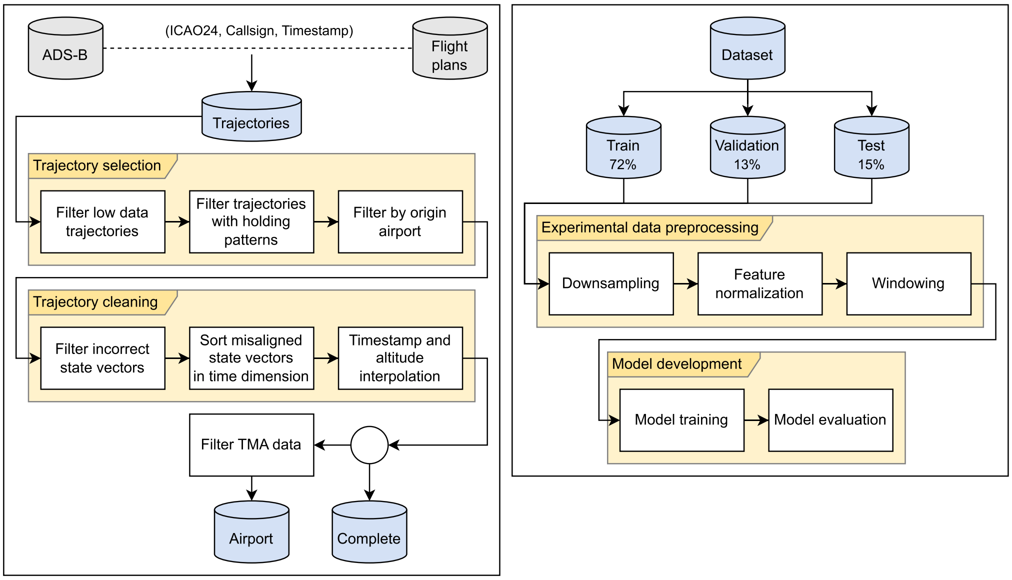

FIGURE 3. Dataset construction (left) and model training (right) workflows.

TABLE 1. Examples of surveillance ADS-B data as provided by OpenSky (flight state vectors).

Timestamp	ICAO24	Callsign	Origin country	Last position update	Last update	Longitude	Latitude	Baro-altitude
1688335192	a36898	DAL2926	United States	1688335191	1688335191	-105.7453	42.5375	10675.62
1688335192	e8027c	LPE2280	Chile	1688335191	1688335192	-77.4018	-11.7594	4000.50
1688335192	34444b	IBE05ZB	Spain	1688335192	1688335192	-7.0408	41.9622	8389.62
1688335192	39dd40	CCM88CE	France	1688335183	1688335184	7.7983	43.0893	3962.4

On ground	Speed	Track	Vertical rate	Sensors	Geo-altitude	SSR code	Special indicator	Position source
false	238.66	71.14	0.00	null	11163.3	null	false	0
false	184.07	321.24	9.10	null	4244.34	2604	false	0
false	234.05	111.94	8.13	null	8869.68	5117	false	0
false	143.8	152.98	0	null	4152.90	null	false	0

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. We identify the last version of the flight plan (which contains the most up-to-date information) to extract the features explained above.

3) TARGET VARIABLE

We aim to predict the position of the aircraft in the short-term future, that is, in the n next time-steps. Therefore, we define the longitude, latitude and altitude values for the next n timesteps as the labels for each of the input elements.

TABLE 2. Model features extracted from the data sources.

Feature	Description	Sample value
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
Track	Angle between aircraft heading and north	267
Sector	Approach direction to destination airport	1
Departure airport	ICAO code of origin airport	*EGLL*

B. DATA PROCESSING

The acquired raw data collection needs to be transformed to ensure high-quality 4D trajectories. A high-level description of this workflow is provided in Figure 3 and its three transformation stages are described as follows.

1) TRAJECTORY SELECTION AND CLEANING

We first identify and filter relevant trajectories for our case study using flight plan information (such as departure and arrival times and airports) and flight ID information (the callsign of the flight and the aircraft ICAO code). Each trajectory is associated with the corresponding surveillance data, captured from that aircraft during the flight. Trajectories with less than 300 state vectors and trajectories with multiple loops during a holding procedure are discarded. In the latter case note that holding procedures force an aircraft to wait in the air until it is cleared to land at the airport, and are characterized by their looping trajectory pattern near the airport.

The identified 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…). The process 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 recalculation of timestamp and altitude values for reordered vectors.

2) DOWNSAMPLING

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 15 seconds according to their timestamp: the first vector of each bucket (in chronological order) is kept, and the rest are discarded. We set this sampling rate at 15 seconds, which we have found to be a good compromise between reducing the computational cost of the model training, and the representativeness of the original trajectory data.

3) TRANSFORMATION AND WINDOWING

Data is further transformed to fit the model requirements. First, the track feature is transformed according to $f(x) = \sin\left(\frac{x}{2}\right)$ in order to reinforce the similarity between 0º and 359º. The track is divided by 2 to fit the output in the [0, 1] range. This feature is complemented by an additional binary feature, *sector*, in order to differentiate between tracks in the range [0, 180) and [180, 360), given that the defined transformation is not injective in this range.

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 Equation 1, where v is the original value of the feature f, and v^f_{min} and v^f_{max} are the observed minimum and maximum values.

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

Finally, trajectories are transformed into fixed-length sequences of n elements. For each trajectory, we generate all possible subsequences of m neighbouring vectors (using a sliding window of equal length) and assign them the next p vectors in the sequence (the number of timesteps we want to predict). The sub-sequences that contain a gap of more than 180 seconds between adjacent vectors are also removed to ensure the continuity within each window.

VI. 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 evaluate the performance and generalizability of our model. We then present our main findings and discuss the results and their contribution to the state of the art.

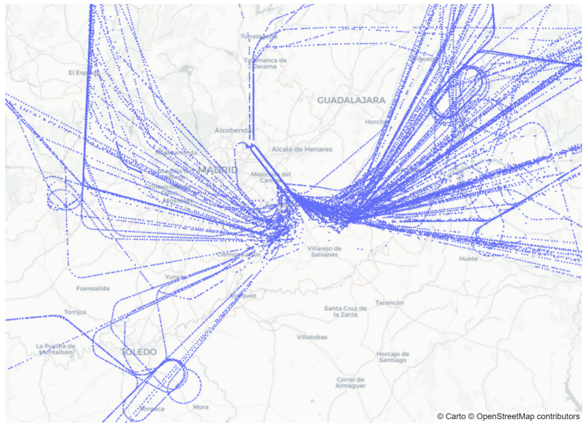

FIGURE 4. Sample of 200 flights arriving at LEMD (Jan, 2022).

A. CASE STUDY

Our case study focuses 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 2023. 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 are illustrated in Figure 4, which shows prevailing the runway configuration at that moment.

B. DATASET GENERATION

Our dataset (referred to as *Complete*) includes data from incoming flights at LEMD airport in the first nine months of 2022 (January 1 to September 30, 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 reduce this effect, and limit each route to have a maximum of 50 trajectories per month. As a result, the dataset comprises data from 7,146 trajectories (2,971,108 state vectors).

A second dataset is defined (called *Airport* from now on), which is a subset of the dataset described above. This dataset contains all state vectors that were emitted closer than 80NM to the destination airport. The purpose of *Airport* dataset is two-fold: (i) to evaluate the performance of the developed models in the TMA of the airport; (ii) to produce results that are comparable to those reported in the state of the art, which mainly focus on this area around the airport, and not on the whole trajectories.

C. 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 6-core Intel(R) Xeon(R) Gold 6226R CPU at 2.90GHz, with 16GB RAM with GPU acceleration on a Nvidia RTX A40. The execution environment includes Python 3.9; Tensorflow 2.9.1 and Keras 2.11.0 for LSTM models; and Pytorch 1.13.1 for TFT models. We used a publicly accessible TFT implementation[2] that is based on the original implementation from [6].

1) DATASET

The dataset is divided into the usual *train*, *validation* and *test* subsets (containing 72.25%, 12.75%, and 15% of the trajectories respectively), as depicted in in Figure 3. 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 is adapted to the particular model configuration, according to the process described in Section V-B.

2) LSTM MODEL

We evaluated two parameters: the input window size (m) and the number of units of the LSTM network. On the one hand, input window size determines the number of individual elements (state vectors) the model expects to receive 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. 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. We tested window sizes between *25* and *60*, in increments of *5*, and a range of different numbers of units between *10* and *35*, based on previous experiences with this task.

We assigned fixed values to the other hyperparameters of the model: we set the *sigmoid function* as the activation function; the batch size to *128*; the *loss function* to mean absolute error (MAE), and Adam [23] is used as optimizer. We experimented with different activation functions (namely, linear, hyperbolic tangent and sigmoid), but the sigmoid function provided the best results in terms of training stability and convergence speed. During training, early stopping was used as a regularization measure to prevent overfitting. The models were trained for *40* epochs, and the version with the smallest validation loss was selected for evaluation. The best model was configured with 30 LSTM units and input sequences of size 55.

[2] https://github.com/andresC98/TSF_Transformers_TFM

3) TFT

We tuned three hyperparameters in the TFT model: head attention size, number of elements in the hidden layer, and the input window size. The head attention size is the number of parallel attention layers applied to the input of the model. Each of these layers may learn to detect different patterns, effectively ''paying attention'' to different aspects in the time series. The number of hidden elements corresponds with the number of units in the LSTM network that is integrated in the TFT architecture. The evaluated head attention sizes were 1, 2 and 4, and the number of hidden elements ranged from 140 to 380, in increments of 60 units. Finally, for the window size we experimented with values of m between *50* and *65* elements. The rest of the hyperparameters were set to the default values defined in the used implementation. The best results were achieved using 4 attention heads, a hidden size of 320 and input sequences of 60 state vectors.

4) METRICS

The models are evaluated using MAE (Mean Absolute Error), RMSE (Root Mean Squared Error) and MSE (Mean Squared Error) metrics.

- MAE is the mean of the absolute values of the differences between each objective sequence, $(x_{t+1}, \ldots, x_{t+m})$, and the predicted sequence, $(y_{t+1}, \ldots, y_{t+m})$, for every input $(x_{t-p}, \ldots, x_t)$ (Equation 2).
- MSE is the mean value of the squared differences between each objective value and the predicted value (Equation 3).
- 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 (Equation 4).

$$\mathrm{MAE} = \frac{1}{n}\sum\nolimits_{i=1}^{n} |y_i - x_i| \tag{2}$$

$$\mathrm{MSE} = \frac{1}{n}\sum\nolimits_{i=1}^{n} (y_i - x_i)^2 \tag{3}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum\nolimits_{i=1}^{n} (y_i - x_i)^2} \tag{4}$$

where n is the number of examples in the evaluation dataset, y_i is the tensor comprising the predicted next p future states, and x_i is the tensor that contains the actual next p timesteps.

Due to its linear nature, MAE weights equally each example regardless of its error value. In RMSE and MSE 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 MAE and RMSE values are provided in degrees (latitude and longitude) and feet (altitude) to enable direct comparisons. MSE values are normalized in the range [0,1).

MSE was used as metric during the preliminary experiments. MAE and RMSE are used in the evaluation of the final

TABLE 3. MSE (scaled) on the validation set for all feature sets.

Approach	MSE LSTM	MSE LSTM-FC	MSE TFT
Basic	0.000126	0.000124	**0.0000749**
Basic+sector	0.000125	0.000119	0.0000877
Expanded	0.000118	0.000124	0.0000942
Expanded+sector	**0.000117**	**0.000117**	0.0000768

models in order to enable direct comparisons with the state of the art.

D. PRELIMINARY STUDIES

In this section, we report some findings in our preliminary study to guide the experimentation process.

1) ITERATIVE VERSUS COMPLETE SEQUENCE

In our first tests, we evaluated the recursive and MIMO strategies to approach the trajectory prediction task. We implemented two simple LSTM networks, which predicted one timestep and ten timesteps ahead, respectively. The results confirmed the conclusions exposed in [10]. The recursive approach performed worse, with a MSE of 0.004400. This strategy led to an accumulation of errors that increased with successive predictions. In particular, the predicted trajectory often showed non-existent turns after a small deviation in the predicted values (mainly latitude or longitude). On the other hand, the MIMO strategy, while making the model more difficult to fit, had a much lower MSE value: 0.000129 (an order of magnitude lower). Therefore, we adopted MIMO strategy for the rest of our experiments.

2) FEATURE SET

Two different sets of input features were put under evaluation: *Basic* (latitude, longitude, altitude and track), and *Expanded* (basic features plus speed, departure airport, and distance to destination airport). We also evaluated the effect of the sector feature in the prediction capabilities of the models.

The results are shown in Table 3. Both LSTM models benefited from having the additional features and improved their MSE values with the *Expanded+sector* feature set, while the TFT model performed best with the *Basic* feature set. However, in both cases the differences were rather small. The best sets of features for each architecture are used for the rest of the experimentation.

E. RESULTS

Table 4 shows the main results of our experiments on the *Complete* dataset, corresponding to the complete trajectories from flights arriving at LEMD airport, and the *Airport* dataset, where only the vectors within the destination airport TMA are considered. The MAE and RMSE values on each dataset are also presented in Figure 5 (top and bottom, respectively). We report per-feature values in their original scale and units (i.e. latitude and longitude in degrees, and altitude in feet) to facilitate their interpretation and analysis.

On the one hand, the LSTM and LSTM-FC models report very similar numbers for the Complete dataset on all the considered features, in terms of both MAE and RMSE. There are small differences depending on the predicted feature: LSTM performs better in longitude (0.0364 vs 0.0373 degrees, equivalent to 4.04 and 4.14 km, respectively) and altitude (242.8 vs 259.2 feet), while LSTM-FC has a slight advantage at predicting the latitude (0.0216 vs 0.0208 degrees, around 2.3 kilometers). The TFT model report different results. The MAE for the latitude is the same as the achieved by LSTM, although with half the variance according to the RMSE (0.0403 vs 0.0910 degrees). The MAE for longitude is worse, with 0.0574 degrees (6.38 km), but the RMSE halves those presented by the LSTM and LSTM-FC models. Finally, the results for altitude are better in terms of MAE and RMSE, with the lowest values of the three models (189.8 and 465.9 feet, respectively). The difference between MAE and RMSE leads to the conclusion that LSTM models perform better than TFT in the individual trajectories that are closer to the most frequent trajectory, and worse in less frequent scenarios (such as adverse weather conditions, diverted flights or special procedures like holdings or go-arounds), where they make larger errors than TFT and therefore get penalized by the RMSE metric.

We also trained and evaluated the models on the *Airport* dataset (with the same model configuration used for the Complete dataset). The bottom line in Table 4 shows the corresponding results. Here, the TFT model provides the best performance at predicting the 2D position (latitude and longitude), and improves significantly its altitude predictions with respect to the previous experiment. Our intuition is that data from outside the TMA is too disperse and scarce to train the TFT model correctly, so its performance should improve as we increase the size of the dataset. This hypothesis is further discussed in a later section. Inside the TMA, all the trajectories converge, which reduces the variability found in the data, and the size of the dataset is large enough for the model to capture the movement patterns around the destination airport. On the other hand, the LSTM models provide error values that are on-par than those observed on the other experiments in terms of latitude and latitude, while the longitude predictions are worse.

F. ANALYSIS

Table 5 summarizes the main results of the state of the art (Section III), including those reported by the best LSTM and TFT-based approaches presented in this paper.

Although none of these approaches is directly comparable, as they all represent different case studies (different airports, number and type of routes, etc.), we think it is interesting to analyze these numbers to position our current contributions with respect to the existing literature. To do this, we have arranged the results in Table III according to the number of routes that each approach able to predict: the top part shows the results of approaches that work with data from a single route, while the bottom part shows the reported results

TABLE 4. Evaluation results for the Complete and Airport datasets. Values in original units (degrees for latitude and longitude, and feet for altitude). The best values for each result set and feature are marked in bold.

Model	Dataset	MAE			RMSE		
		lat	lon	alt	lat	lon	alt
LSTM	Complete	0.0216	**0.0364**	242.8	0.0910	0.1470	820.2
LSTM-FC	Complete	**0.0208**	0.0373	259.2	0.0910	0.1490	820.2
TFT	Complete	0.0216	0.0574	**189.8**	**0.0403**	**0.0834**	**465.9**
LSTM	Airport	0.0238	0.0544	360.9	0.0350	0.0940	557.7
LSTM-FC	Airport	0.0255	0.0855	367.5	0.0360	0.1430	590.6
TFT	Airport	**0.0091**	**0.0104**	**225.3**	**0.0162**	**0.0238**	**381.9**

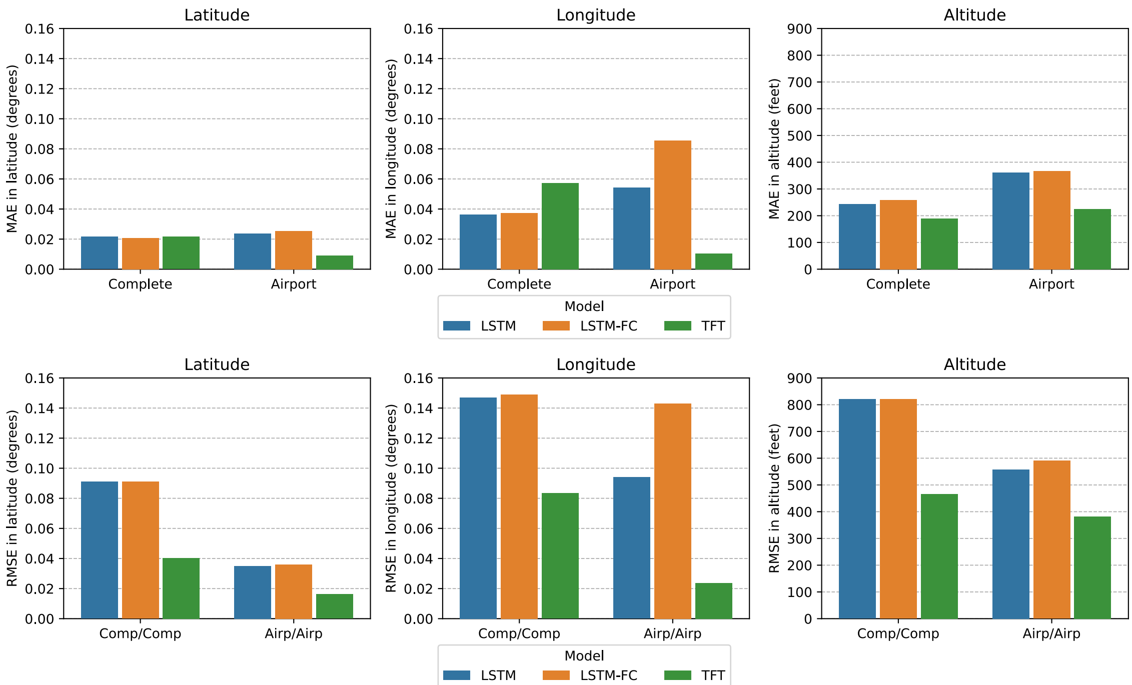

FIGURE 5. Comparison of the MAE (top) and RMSE (bottom) values of the models in the considered scenarios.

TABLE 5. MAE and RMSE values for latitude, longitude and altitude features from comparable proposals.

Ref	Year	Approach	Strategy	Scope	MAE			RMSE		
					lat	lon	alt	lat	lon	alt
Single route										
[12]	2021	ConstLSTM	One-step	Complete trajectory	**0.0050**	**0.0105**	9.96	**0.0203**	0.0482	45.34
[15]	2020	CNN-LSTM	One-step	Complete trajectory	0.0170	0.0710	33.83	0.0230	0.0860	40.14
[18]	2022	LSTM+Att	One-step	Not indicated	0.0373	0.0397	**2.94**	0.0464	0.0494	**3.92**
[14]	2022	BiLSTM	Direct	Complete trajectory	0.0206	0.0160	33.75	0.0264	**0.0222**	52.34
Multiple routes										
[11]	2018	LSTM	One-step	Airspace sector	0.0725	0.0552	**77.95**	0.2295	0.1337	**134.51**
Ours		TFT	MIMO	TMA	**0.0091**	**0.0104**	225.3	**0.0162**	**0.0238**	381.9
		LSTM	MIMO	Complete trajectory	0.0216	0.0364	242.80	0.0910	0.1470	820.20

of techniques (including ours) that are able to predict the trajectory for multiple routes. On the other hand, it is worth noting that most of the studied approaches implement an one-timestep prediction (with the exception of [14] and its Direct approach), while ours uses a MIMO approach. This means that the errors reported in those papers correspond to the error in the prediction of the next timestep (which is the one with the least uncertainty), while our results describe the errors for the next 10 timesteps, whose uncertainty increases progressively. In consequence, the error made by

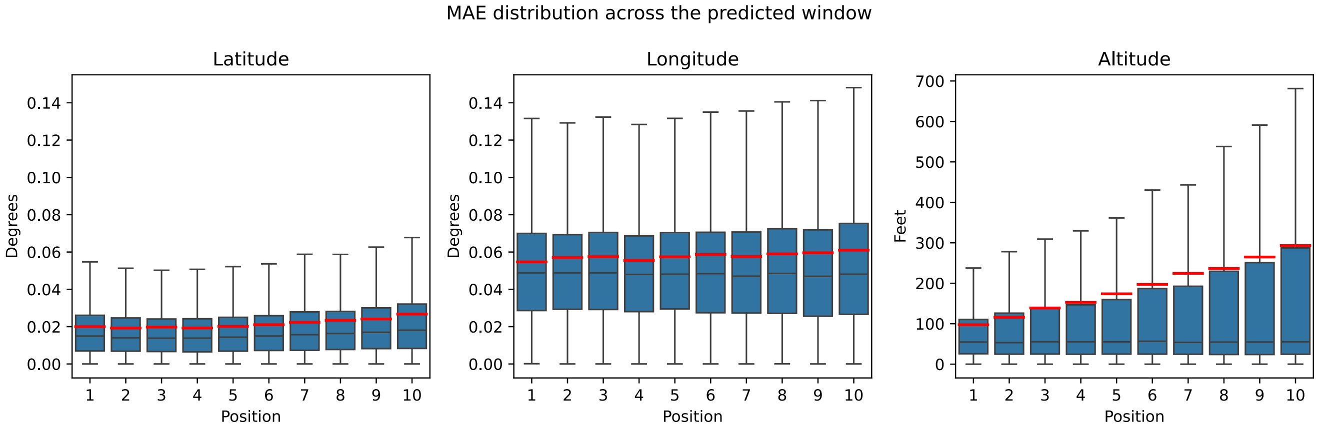

FIGURE 6. Distribution of the mean absolute error of the TFT model across the predicted window. The red lines indicate the mean values across the vectors in each position.

TABLE 6. Mean and standard deviation values for the MAE values on each of the positions in the predicted window.

	Latitude		Longitude		Altitude	
Position	Mean	StD	Mean	StD	Mean	StD
1	0.0205	0.0251	0.0559	0.0506	99.70	195.86
2	0.0193	0.0255	0.0559	0.0507	116.93	246.48
3	0.0194	0.0276	0.0562	0.0537	137.73	294.76
4	0.0196	0.0299	0.0564	0.0562	157.96	340.22
5	0.0202	0.0322	0.0570	0.0590	179.30	386.18
6	0.0210	0.0342	0.0576	0.0615	200.48	429.71
7	0.0221	0.0363	0.0581	0.0637	221.22	470.83
8	0.0233	0.0385	0.0588	0.0660	241.56	510.90
9	0.0247	0.0405	0.0594	0.0686	261.88	551.68
10	0.0260	0.0439	0.0592	0.0711	281.24	594.38

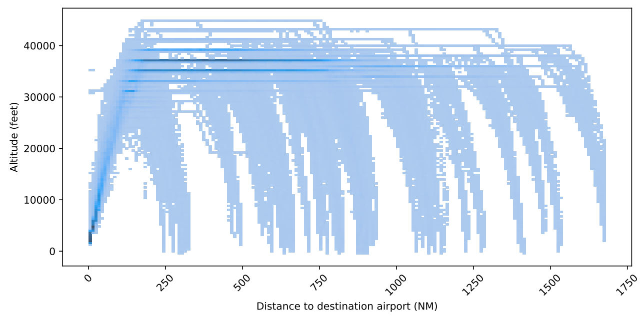

FIGURE 8. Heatmap describing the correlation between altitude and distance values.

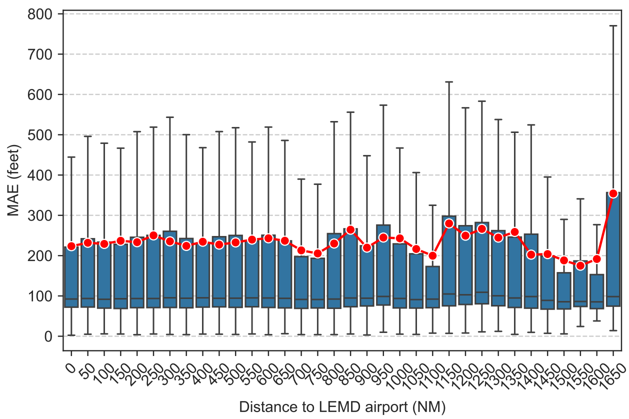

FIGURE 7. Distribution of the mean absolute error values of the TFT model on the altitude feature at different distances to the destination airport. The mean value at each distance is indicated in red.

the model would vary across the predicted window. This intuition is confirmed in Figure 6, where the mean absolute error distribution on each of the predicted window positions is shown for the Complete dataset. The outliers (values that exceed 1.5 times the interquartile range) have been removed from the graph for clarity. The mean and standard deviation values (without excluding the outliers) are shown in Table 6. The mean MAE for altitude increases monotonically from 100 to 281 feet. The dispersion of the values also increases progressively due to higher uncertainty the further in time we aim to predict. The error distribution in latitude and longitude is more uniform along the predicted window in terms of mean errors and dispersion. However, there is still a slight but steady increment of the error the further we advance within the window. It is worth noting that this fact is not evident in the Figure 6 due to outlier removal, although it becomes clear in Table 6. The only exception is that the error in latitude is slightly higher in the first predicted vector than in the second one.

Taking these aspects into account, our numbers are competitive with those reported in the state of the art in terms of latitude and longitude. TFT outperforms all the considered works in terms of the MAE in 2D positioning except [12], albeit this case study is performed on a single route, which significantly reduces the variance of the predicted trajectories, and only predicts the next timestep. Similar conclusions can be made for our LSTM model, which manages to compete with, or even outperform, most of the analyzed LSTM-based proposals in terms of MAE.

The situation is different when it comes to altitude, with our models performing worse than the state of the art. Figure 7 shows the distribution of the MAE values of the TFT model at different distances to the destination airport.

The mean and median error values are mostly uniform up to 650 NM, although the dispersion of the error values fluctuates significantly. These variations in the metrics are aligned with the start of the routes: as shown in Figure 8, many trajectories manifest ascending patterns at several points within the considered geographical scope, introducing additional variability into the data of flights occurring in those areas. As the aircraft gets closer to the airport, the flight patterns become more clearly defined, which help the model to provide more consistent results. This can be observed in the range (200-650), where most of the air traffic organizes at different, but defined, flight levels. However, the error is still lower than acceptable from a domain point of view: MAE in altitude is 318 feet, which is much lower than the minimum vertical separation for aircraft crossing paths standardized by ICAO. ICAO specify a minimum separation of 1,000 feet for aircraft flying at 29,000 feet at most, and 2,000 feet if they fly above this value [24].

VII. CONCLUSION AND FUTURE WORK

This paper reports the first experience of applying the TFT architecture to the aircraft trajectory prediction problem. This architecture is particularly suitable for this task, thanks to its combination of attention mechanisms, LSTM internal networks and features management. We also developed several LSTM models to serve as a baseline and help to interpret the results of TFT. Our approach report competitive results, outperforming similar works even in a more complex context. In particular, TFT is able to predict 2D position with high accuracy, with a MAE of 0.0133 and 0.0170 degrees for latitude and longitude, respectively. The results for altitude are less competitive due to the inclusion of the complete trajectories from multiple routes, which introduce a higher variance of the altitude across the used dataset. Further investigation of the causes that motivate this fact and the strategies to overcome it are part of our future work. Our first steps towards this objective will be devoted to construct enriched trajectories with weather and flight planning data to further refine predictions and improve our current results for altitude. Note that these data play a key role in tactical flight planning within the TMA, and have great influence how flight trajectories develop. TFT architecture, whose performance for trajectory prediction has been explored in this paper, should be able to effectively model the relationship between individual aircraft dynamics and these contextual factors in such a complex environment. Additionally, we will leverage the focus on explainability of the TFT architecture to deepen our knowledge about the data and the performance of these models.

ACKNOWLEDGMENT

The authors acknowledge Boeing Research and Technology Europe (BR&T-Europe) for granting them access to their ADAPT platform.

REFERENCES

[1] *Minimum Aviation System Performance Standards for Automatic Dependent Surveillance Broadcast (ADS-B)*, document DO-242A, RTCA, 2006.
[2] ICAO. (2018). *Global TBO Concept V0.11*. [Online]. Available: https://www.icao.int/airnavigation/tbo/Pages/Why-Global-TBO-Concept.aspx
[3] G. Enea and M. Porretta, "A comparison of 4D-trajectory operations envisioned for NextGen and SESAR, some preliminary findings," in *Proc. 28th Congr. Int. Council Aeronaut. Sci.*, vol. 5, 2012, pp. 4152–4165.
[4] W. Zeng, X. Chu, Z. Xu, Y. Liu, and Z. Quan, "Aircraft 4D trajectory prediction in civil aviation: A review," *Aerospace*, vol. 9, no. 2, p. 91, Feb. 2022.
[5] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.
[6] B. Lim, S. Ö. Arık, N. Loeff, and T. Pfister, "Temporal fusion transformers for interpretable multi-horizon time series forecasting," *Int. J. Forecasting*, vol. 37, no. 4, pp. 1748–1764, Oct. 2021.
[7] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *Int. J. Uncertainty, Fuzziness Knowl. Based Syst.*, vol. 6, no. 2, pp. 107–116, Apr. 1998.
[8] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2016, *arXiv:1409.0473*.
[9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2017, pp. 1–20.
[10] S. Ben Taieb, G. Bontempi, A. F. Atiya, and A. Sorjamaa, "A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition," *Expert Syst. Appl.*, vol. 39, no. 8, pp. 7067–7083, Jun. 2012.
[11] Z. Shi, M. Xu, Q. Pan, B. Yan, and H. Zhang, "LSTM-based flight trajectory prediction," in *Proc. Int. Joint Conf. Neural Netw.*, Jul. 2018, pp. 1–8.
[12] Z. Shi, M. Xu, and Q. Pan, "4-D flight trajectory prediction with constrained LSTM network," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 11, pp. 7242–7255, Nov. 2021.
[13] W. Zeng, Z. Quan, Z. Zhao, C. Xie, and X. Lu, "A deep learning approach for aircraft trajectory prediction in terminal airspace," *IEEE Access*, vol. 8, pp. 151250–151266, 2020.
[14] D. Sahadevan, H. P. M, P. Ponnusamy, V. P. Gopi, and M. K. Nelli, "Ground-based 4d trajectory prediction using bi-directional LSTM networks," *Int. J. Speech Technol.*, vol. 52, no. 14, pp. 16417–16434, Nov. 2022.
[15] L. Ma and S. Tian, "A hybrid CNN-LSTM model for aircraft 4D trajectory prediction," *IEEE Access*, vol. 8, pp. 134668–134680, 2020.
[16] H. Shafienya and A. C. Regan, "4D flight trajectory prediction using a hybrid deep learning prediction method based on ADS-B technology: A case study of Hartsfield–Jackson Atlanta international airport (ATL)," *Transp. Res. Part C, Emerg. Technol.*, vol. 144, Nov. 2022, Art. no. 103878.
[17] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1724–1734.
[18] P. Jia, H. Chen, L. Zhang, and D. Han, "Attention-LSTM based prediction model for aircraft 4-D trajectory," *Sci. Rep.*, vol. 12, no. 1, p. 15533, Sep. 2022.
[19] H.-C. Choi, C. Deng, and I. Hwang, "Hybrid machine learning and estimation-based flight trajectory prediction in terminal airspace," *IEEE Access*, vol. 9, pp. 151186–151197, 2021.
[20] T.-H. Le, P. Tran, D.-T. Pham, M. Schultz, and S. Alam, "Short-term trajectory prediction using generative machine learning methods," in *Proc. 9th Int. Conf. Res. Air Transp.*, vol. 15, Jul. 2020, pp. 1–19.
[21] X. Cheng, B. Khomtchouk, N. Matloff, and P. Mohanty, "Polynomial regression as an alternative to neural nets," Apr. 2019.
[22] M. Schäfer, M. Strohmeier, V. Lenders, I. Martinovic, and M. Wilhelm, "Bringing up OpenSky: A large-scale ADS-B sensor network for research," in *IPSN-14 Proc. 13th Int. Symp. Inf. Process. Sensor Netw.*, Apr. 2014, pp. 83–94.
[23] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, San Diego, CA, USA, 2015. [Online]. Available: http://arxiv.org/abs/1412.6980
[24] ICAO, *Procedures for Air Navigation Services: Air Traffic Management*, document Doc 4444, 2016.

JORGE SILVESTRE received the Bachelor of Engineering degree in computer science, in 2017, and the master's degree in business intelligence, big data, and cyber-security, in 2018. He is currently pursuing the Ph.D. degree with the Computer Science Department, University of Valladolid, Spain. Since 2016, he has earned three investigation grants (one of them from the Regional Government of Castilla y León) for research in natural language processing. He is collaborating, as a Graduate Teaching Assistant with the Computer Science School, Segovia, as a result of a research grant from the University of Valladolid. His research interests include machine learning and big data, and leaded him into his thesis project on applying these approaches to support air traffic management operations. He received the award for the best academic results in both bachelor's and master's degrees. Since then, he has been investigating in the Ph.D. thesis in computer science.

PAULA MIELGO received the dual bachelor's degree in mathematics and computer science engineering, in 2022, and the master's degree in computer science engineering, in 2023. She is currently pursuing the Ph.D. degree with the Computer Science Department, University of Valladolid, Spain. During her studies, she received several academic awards, including the award for the best academic results in both of her CS programs and a regional award for her final degree thesis. Later that year, she started the Ph.D. thesis in computer science. Since 2022, she has been awarded two investigation grants for her research on predictive tasks in air traffic management and computer vision using deep learning and big data. In the Ph.D. thesis project, she delves in these areas by investigating their applications to industry 4.0.

ANIBAL BREGON received the B.Sc., M.Sc., and Ph.D. degrees in computer science from the University of Valladolid, Spain, in 2005, 2007, and 2010, respectively. He joined the Department of Computer Science, University of Valladolid, in 2011, where he has been an Associate Professor, since February 2018. He has been a Guest Researcher with the Intelligent Systems Division, NASA Ames Research Center, and the Institute for Software Integrated Systems, Vanderbilt University. He has carried out both basic and applied research in the areas of fault diagnosis and prognosis for aerospace and industrial systems, and has coauthored more than 85 journal and conference papers. He is currently leading a national funded project on advanced learning for smart manufacturing and several technology transfer contracts on deep learning. He has also participated as a Researcher on several funded projects, networks, and contracts on fault diagnosis and prognosis topics, with a focus on big data analytics and deep learning. His current research interests include model-based reasoning for diagnosis and prognosis, health-management, big data, industry 4.0, and deep learning. Among various other professional activities, he has held different chair positions with PHM and PHME conferences; has been co-administrator of several courses and summer schools on diagnosis, prognosis, and artificial intelligence; and has been the Local Chair of the 2016 European Conference of the Prognostics and Health Management Society.

MIGUEL A. MARTÍNEZ-PRIETO received the B.Sc., M.Sc., and Ph.D. degrees in computer science from the University of Valladolid, Valladolid, Spain, in 2005, 2007, and 2010, respectively. He held a postdoctoral position with the Department of Computer Science, University of Chile, from 2010 to 2012. He is currently an Associate Professor and a Researcher in computer science with the Department of Computer Science, University of Valladolid. His research has been in the area of data management, mainly in data compression and indexing of semantic, text and biological data, and the resolution of specific queries in each of these scenarios. He has coauthored more than 90 peer-reviewed articles on these topics and has been involved in several European and national funded projects, and several transfer contracts with companies and government institutions. His current research interests include data science, with an applications in air traffic management and industry 4.0.

PEDRO C. ÁLVAREZ-ESTEBAN received the B.Sc., M.Sc., and Ph.D. degrees in mathematics from the University of Valladolid, Valladolid, Spain, in 1992, 1994, and 2009, respectively. He is currently an Associate Professor and a Researcher in probability and statistics with the Department of Statistics and Operations Research, University of Valladolid, Spain. He has carried out both basic and applied research in the areas of probability, mathematical statistics, and data science, has coauthored more than 50 peer-reviewed articles, and has participated on several national funded projects on mathematical statistics. He has developed an important collaboration activity with the private sector which has led him to lead more than 20 research contracts in a wide variety of topics, such as genomics/metabolomics, smart grids, automatic loss assessment systems for motor vehicle accidents, and analysis of non-technical losses in power grids or flight trajectory analysis. His current research interests include the analysis of complex data, robust statistical methods, the study of probabilistic metrics, their applications, and data science in general.