

Document downloaded from:

<http://hdl.handle.net/10251/160616>

This paper must be cited as:

Jiménez-Campfens, N.; Colomer, A.; Núñez, J.; Mogollón, JM.; Rodríguez, AL.; Naranjo Ornedo, V. (2020). Deep Learning in Aeronautics: Air Traffic Trajectory Classification Based on Weather Reports. Springer. 148-155. https://doi.org/10.1007/978-3-030-62365-4_14



The final publication is available at

https://doi.org/10.1007/978-3-030-62365-4_14

Copyright Springer

Additional Information

Deep Learning in Aeronautics: Air Traffic Trajectory Classification Based on Weather Reports^{*}

Néstor Jiménez-Campfens¹, Adrián Colomer¹, Javier Núñez², Juan M. Mogollón², Antonio L. Rodríguez², and Valery Naranjo¹

¹ Institute of Research and Innovation in Bioengineering, *Universitat Politècnica de València*, Spain

jojicam@etsii.upv.es, {adcogra, vnaranjo}@i3b.upv.es

² SkyLife Engineering S.L., *Parque Científico y Tecnológico Cartuja*, Sevilla, Spain
{javier.nunez, jmmogollon, antoniorv}@skylife-eng.com

Abstract. New paradigms in aviation, as the expected shortage of qualified pilots and the increasing number of flights worldwide, present big challenges to aeronautic enterprises and regulators. In this sense, a concept known as Single Pilot Operations arises in the task of dealing with these challenges, for which, automation becomes necessary, especially in Air Traffic Management. In this regard, this paper presents a deep learning-based approach to leveraging the job of both ground controllers and pilots. Making use of Meteorological Terminal Air Reports, obtained regularly from every aerodrome worldwide, we created a model based on a multi-layer perceptron capable of determining the approach trajectory of an aircraft thirty minutes prior to the expected landing time. Experiments on aircraft trajectories from Toulouse to Seville, show an accuracy, recall and F1-score higher than 0.9 for the resultant predictive model.

Keywords: Air Traffic Management · Weather reports · METAR · Trajectory prediction · Deep learning.

1 Introduction

Since the 1950s, technological advances in engines, voice communication and navigation equipment have resulted in a decreasing number of cockpit members from 5 to 2 persons. In the last years, further automation and technological developments, added to an expected exacerbation of the existing global shortage of qualified pilots [1] have propitiated an arising interest in a concept known as Single Pilot Operations (SPO), with the aim of reducing the current commercial cockpit crew from 2 to a single pilot, favouring research within this topic [2, 3]. In this sense, it is argued that re-conceptualising the flight-deck and the role of

^{*} This work has received funding from the Clean Sky 2 Joint Undertaking (JU) under grant agreement No 831884. The Titan V used for this research was donated by the NVIDIA Corporation.

the pilot [3] along with an increasing support from a ground-operator [4], are necessary conditions for SPO.

There is great consensus that, in order to implement SPO, several co-pilot functions will have to migrate to ground control stations [2, 5] or be automated in the aircraft. In both cases, it is accepted that the use of automation, through digital assistants, will need to be increased both on-deck and on the ground in order for SPO to be successful [2, 6].

In this regard, decision-support systems have become a necessity with the aim of easing the increasing workload of both pilots and ground operators. In particular, artificial intelligence algorithms could be a great asset in solving difficult, non-trivial tasks such as air traffic conflict resolution, flight delay prediction and trajectory prediction.

With respect to air traffic conflict resolution, Deep Learning (DL) techniques have been proved to be useful to prevent conflicts between two aircraft by using neural networks and genetic algorithms while being time-efficient [7].

With the aim of predicting flight delays, different approaches have been studied based on weather data usage. In this sense, [8] proposed a Machine Learning (ML) based model able to classify airline delays induced by varying weather condition, comparing the efficiency of different algorithms, including Random Forest, AdaBoost, k-Nearest-Neighbours and Decision Trees. From a different perspective, [9] proposed a Long Short-Term Memory (LSTM) architecture to predict flight delays including airport and weather data, among other features.

With regard to trajectory prediction, literature shows that the number of studies applying ML or DL techniques to trajectory prediction and classification, based on weather data, is limited compared to that of delay prediction and conflict resolution. In this regard, in [10] an Encoder-Decoder Recurrent Neural Network (RNN) was used to predict a flight trajectory using the flight plan as input and incorporating weather data of areas close to the planned trajectory. In [11], a LSTM network was used to predict the trajectory of the flight using position data such as altitude, longitude and latitude, as input to the model, stating that not including weather data and its fluctuating features makes the model prone to sharp turns in the predicted trajectory and, in consequence inaccurate predictions. More extensively, [12] presented a wider comparison of multiple ML and DL supervised algorithms in the task of landing runway trajectory classification of arriving aircraft, based on trajectory features (latitude, longitude, speed, altitude and course angle), and specific characteristics of the aircraft and airport.

In this sense, our work proposes a Multi Layer Perceptron (MLP) algorithm trained to predict the approach trajectory of an aircraft thirty minutes prior to its landing, using solely weather reports generated by the destination airport. For our work, the trajectory between the airports of Toulouse (TLS) and Seville (SVQ) was considered.

2 Materials and Methods

For this study, two types of data were necessary: flight trajectory data, including position (latitude and longitude), altitude, speed and direction of each flight at every given moment; and weather data at the destination airport, encoded as Meteorological Terminal Air Reports (METAR). For each flight, two METAR reports were gathered previous to the expected landing time following the criteria in table 1 and being $X \in \{0 - 24\}$. In total, a set of 237 samples of flights travelling from Toulouse to Seville between the 4th of November, 2017, and the 14th of January, 2020. were included into the experiments.

Table 1: META Reports timestamps

Expected landing	METAR 1 Timestamp	METAR 2 Timestamp
Xh 00m – Xh 15m	(X-1)h 30m	(X-1)h 00m
Xh 15m – Xh 45m	Xh 00m	(X-1)h 30m
Xh 45m – Xh 59m	Xh 30m	Xh 00m

2.1 Data Acquisition

In order to gather the data, a web-scraping algorithm was used using the *flightradar24* website [13]. Indicating a flight reference, we automatically updated the generated database with any new flights information in *flightradar24*.

2.2 Data Curation

Landing trajectory for Ground-Truth Generation Working with a retrospective database, the complete trajectory is known. In order to train a classification model, we labelled each flight according to its Terminal Arrival Point (TAP) using a non-supervised clustering algorithm (*k*-means). The average of the last 20 sampled points for longitude and latitude were fed to the *k*-means method. As initial cluster centroids, we used the coordinates of each point of interest (one for every TAP - in the case of the Seville airport $k=2$, i.e. Rotex and Santa).

Weather Reports Standarization METAR reports encode weather information in a semi-structured way. These reports include a combination of multiple fields (separated by blank spaces), some of which are always present and some of which are optional, ones with fixed length and others with variable length depending on the weather condition, and some fields representing quantitative features (e.g. wind speed or temperature) and others qualitative features (e.g. cloud or rain types). A sample of a METAR report from the Airport of Seville can be seen in Table 2:

Table 2: A representative sample of METAR Report

LEZL 132230Z VRB01KT 0400 R27/0450D R09/0325N FG VV003 07/07 Q1023 TEMPO 3000 NSW
--

To structure this data, taking advantage of its particular disposition, a regular-expression search algorithm was used, obtaining an array of 64 features for each report. Quantitative information is combined with one-hot-encoded categorical information for qualitative data (see Table 3).

Table 3: Information extracted from the META Reports to build the predictive model

Quantitative data	Qualitative data
Expected landing hour	Presence of variable direction wind
Expected landing minute	Ceiling And Visibility OK (CAVOK)
Wind Direction (°)	Weather
Maximum wind direction (°)	Expected change in weather (TEMPO)
Minimum wind direction (°)	Fraction of sky covered by clouds
Wind speed (Knots)	Cloud type
Wind gusts (Knots)	
Visibility (m)	
Expected Visibility (m)	
Temperature (°C)	
Dew point (°C)	
Pressure (hPa)	
Cloud altitude (feet)	

2.3 Data Segregation

With the goal of training and validating our model, we shuffled and split the data into two subsets. The first one for training and validating the model (80% of the complete dataset), to fit the parameters w (weights) and b (bias) of the network and to optimise the MLP hyperparameters, respectively. The remaining 20% of the data comprises the testing subset used to assess the model performance. Our dataset was consequently split into 190 flights to train and validate the model and 47 samples for testing its performance.

Moreover, each field of the input data (structured weather data) was standardised calculating its z-score as $z = \frac{F-\mu}{\sigma}$, where F is the feature sample, and μ and σ are the mean and standard deviation of the feature distribution, respectively. It is important to note that test data was standardised using μ and σ parameters from the training and validation set.

2.4 Input data selection

To be useful, a predictive model should be able to classify the approach and landing trajectory with such anticipation that the final trajectory is still unknown. Looking at all 237 trajectories in Fig. 1, we can assume at 30 minutes before landing the final trajectory is not yet discerned.

Consequently, input data should be taken from the last METAR report available acquired at least 30 minutes before the expected flight landing time. With this assumption, from the two METAR reports available for each flight, we selected one as input data to the model following the rules presented in Table 4 and being $X \in \{0 - 24\}$:

Table 4: META Report selection

Expected landing	Selected report
Xh 00m – Xh 15m	METAR 1
Xh 15m – Xh 30m	METAR 2
Xh 30m – Xh 45m	METAR 1
Xh 45m – Xh 59m	METAR 2

2.5 Classification Model

In this study, we used a DL model based on a MLP with descending number of neurons. Starting with an input layer of 64 neurons (one for each input item from the structured weather report), 4 hidden layers with 32, 16, 8 and 4 neurons, respectively, and an output layer of 2 neurons (one for each possible class/TAP).

We used the ReLU activation function for every fully connected layer except for the last one, for which we used the SoftMax activation function, to calculate the probability of each sample to belong to each class.

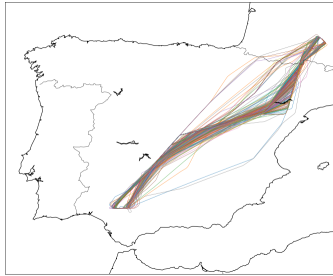


Fig. 1: Visualization of the 237 available trajectories (Toulouse-Seville) in our dataset.

After each layer, we applied Batch Normalization in order to prevent Internal Co-variance Shift [14], and a Dropout layer (0.25) with the aim of randomly ignoring 25% of the neurons in each layer during training stage in preventing over-fitting [15]. We initialised the batch size to $\beta=16$ in order to update the weight parameters of our model after forward propagating 16 samples through the net and calculating its error using the categorical cross-entropy loss function (1):

$$\mathcal{L}(y, \hat{y}) = - \sum_i y_i \log \hat{y}_i \quad (1)$$

where y is the ground-truth label of a specific sample and \hat{y} is the predicted label of such sample among the i possible classes (i.e. $i = k = 2$). Furthermore, we applied the Adam optimiser [16] to update the network weights after forwarding the inputs of each batch and calculating the error. We trained our model over 200 epochs, using a validation split of 0.2 (20% of the training dataset -38 samples- was used to validate the model).

Note that the experimental part of this work was performed in Python 3.5, using TensorFlow 2.0 and using the high-level Keras API [17] for building and training the MLP.

3 Results

3.1 Ground-truth clustering stage

Regarding the ground-truth generation stage, the k -means algorithm is able to successfully cluster each trajectory to its appropriate class. The results of the classification algorithm are reported in Fig. 2. In green, the aircraft trajectories whose TAP is located at west (Santa) are observed while in red the trajectories entering by the east TAP (Rotex) of the destination aerodrome are represented. This fact allows us to train a predictive model from a gold standard automatically generated using the k -means clustering algorithm taking into advance the longitude and latitude of the flight trajectories.

3.2 MLP training stage

This staged is performed after the ground-truth generation for each flight's trajectory and after shuffling and partitioning our METAR dataset (in which each instance learning is composed of 64 features) in two subsets (see Table 5).

With regard to the training and validation stage, we can affirm that, after 200 epochs, no signs of over-fitting are evidenced.

3.3 Inference

When checking the performance of our trained model on new data, belonging to the test set (the 47 data samples separated and not used to train nor validate the model), we obtained the results reported in Table 6.

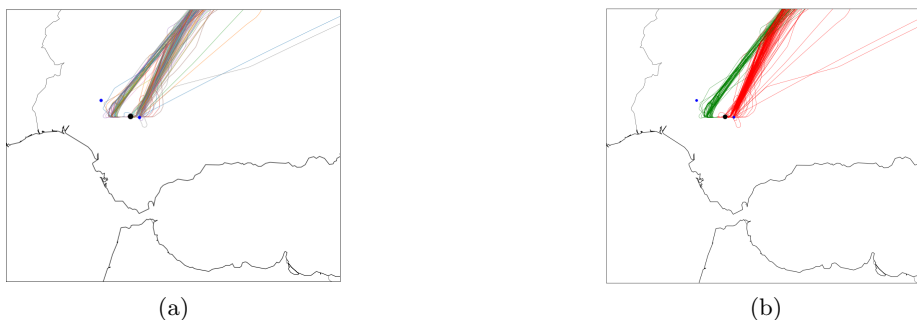


Fig. 2: Automatic ground-truth generation (clustering results). (a) Raw trajectories of the training subset and (b) classified trajectories according to its terminal arrival point.

Table 5: Data distribution structure

Landing trajectory	Training & Validation set	Test set	Total
Rotex	131	29	160
Santa	59	18	77
Total	190	47	237

Table 6: Test results: most suitable terminal arrival point for the 47 samples

Terminal Arrival Point	Accuracy	Recall	F1-score
Rotex	0.94	0.97	0.96
Santa	0.92	0.85	0.88
Weighted average	0.94	0.94	0.94

Analysing the results, overall values show a promising modelling of the selection of the most suitable TAP according to the weather information of the destination airport. A slightly out-performance of the model when predicting the Rotex TAP is registered. This fact may be due to the data distribution shown in Table 5, where we can see that our dataset is not equally split between the two target classes (160 Rotex samples and 77 Santa learning instances).

4 Conclusion

In this work, we proposed a DL-based classification model able to predict flight approach trajectories based exclusively on meteorological reports. More particularly, this model was trained with historical weather reports of the destination airport and trajectory data of a particular air route, making use of a MLP architecture.

In future works, with the aim of improving the model, a bigger and more balanced dataset should be used in the training and testing stages. In this sense,

more trajectories concerning the same destination airport could be added. Using this work as a baseline, future investigations should try to apply the proposed model to different airports with more TAPs. Furthermore, this work could be used as a first step for trajectory regression algorithms making use of RNN architectures to predict the entire sequential approach trajectory.

References

1. Pilot and technical outlook, Seattle, WA:Boeing Commercial Airplanes, 2015.
2. C. A. Wolter and B. F. Gore, “NASA/TM-2015-218480: A validated task analysis of the Single Pilot Operations concept,” no. January 2015, 2015.
3. D. Harris, “A human-centred design agenda for the development of single crew operated commercial aircraft,” *Aircr. Eng. Aerosp. Technol.*, vol. 79, no. 5, pp. 518–526, 2007.
4. R. E. Bailey, L. J. Kramer, K. D. Kennedy, C. L. Stephens, and T. J. Etherington, “An assessment of reduced crew and single pilot operations in commercial transport aircraft operations,” *AIAA/IEEE Digit. Avion. Syst. Conf. - Proc.*, vol. 2017-Septe, no. February 2018, 2017.
5. J. Lachter, S. L. Brandt, V. Battiste, S. V Ligda, M. Matessa, and W. W. Johnson, “Toward Single Pilot Operations: Developing a Ground Station,” *Proc. Int. Conf. Human-Computer Interact. Aerosp.*, no. August, 2014.
6. D. Comerford, S. L. Brandt, and R. Mogford, “NASA / CP — 2013 – 216513 NASA ’ s Single -Pilot Operations Technical Interchange Meeting : Proceedings and Findings,” no. April, p. 89, 2013.
7. N. Durand, J. M. Alliot, and F. Médioni. (2000). Neural nets trained by genetic algorithms for collision avoidance. *Applied Intelligence*, 13(3), 205-213.
8. S. Choi, Y. J. Kim, S. Briceno, and D. Mavris. (2016, September). Prediction of weather-induced airline delays based on machine learning algorithms. In 2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC) (pp. 1-6). IEEE.
9. G. Gui, F. Liu, J. Sun, J. Yang, Z. Zhou, and D. Zhao. (2019). Flight delay prediction based on aviation big data and machine learning. *IEEE Transactions on Vehicular Technology*.
10. Y. Liu, and M. Hansen. (2018). Predicting aircraft trajectories: a deep generative convolutional recurrent neural networks approach. arXiv preprint arXiv:1812.11670.
11. Z. Shi, M. Xu, Q. Pan, B. Yan, and H. Zhang. (2018, July). LSTM-based flight trajectory prediction. In 2018 IEEE International Joint Conference on Neural Networks (IJCNN) (pp. 1-8).
12. C. D. Bosson, and T. Nikoleris. (2018). Supervised learning applied to air traffic trajectory classification. In 2018 AIAA Information Systems-AIAA Infotech@ Aerospace (p. 1637).
13. FlightRadar24 website (<https://www.flightradar24.com/>).
14. S. Ioffe, and C. Szegedy. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167.
15. N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1), 1929-1958.
16. D. P. Kingma, J. and Ba. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.
17. F. Chollet and others. (2015). Keras (<https://keras.io>).