



HAL
open science

Detecting Controllers' Actions in Past Mode S Data by Autoencoder-Based Anomaly Detection

Xavier Olive, Jeremy Grignard, Thomas Dubot, Julie Saint-Lot

► **To cite this version:**

Xavier Olive, Jeremy Grignard, Thomas Dubot, Julie Saint-Lot. Detecting Controllers' Actions in Past Mode S Data by Autoencoder-Based Anomaly Detection. SID 2018, 8th SESAR Innovation Days, Dec 2018, Salzburg, Austria. <hal-02338690>

HAL Id: hal-02338690

<https://hal.science/hal-02338690v1>

Submitted on 30 Oct 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

Detecting Controllers' Actions in Past Mode S Data by Autoencoder-Based Anomaly Detection

Xavier Olive, Jeremy Grignard, Thomas Dubot
ONERA/DTIS, Université de Toulouse
Toulouse, France

Julie Saint-Lot
ENAC, Université de Toulouse
Toulouse, France

Abstract—The preparation and execution of training simulations for Air Traffic Control (ATC) and pilots requires a significant commitment of operational experts. Such a mobilisation could be alleviated by a decision support tool trained to generate a realistic environment based on historical data. Prior to studying methods able to learn from a dataset of traffic patterns and ATC orders observed in the past, we focus here on the constitution of such a database from a history of trajectories: the difficulty lies in the fact that past flown trajectories are properly regulated, that observed situations may depend on a wide range of potentially unknown factors and that ownership rules apply on parts of the data. We present here a method to analyse flight trajectories, detect unusual flight behaviours and infer ATC actions. When an anomaly is detected, we place the trajectory in context, then assess whether such anomaly could correspond to an ATC action. The trajectory outlier detection method is based on autoencoder Machine Learning models. It determines trajectory outliers and quantifies a level of abnormality, therefore giving hints about the nature of the detected situations.

Results obtained on three different scenarios, with Mode S flight data collected over one year, show that this method is well suited to efficiently detect anomalous situations, ranging from classic air traffic controllers orders to more significant deviations. Detecting such situations is not only a necessary milestone for the generation of ATC orders in a realistic environment; this methodology could also be useful in safety studies for anomaly detection and estimation of probabilities of rare events; and in complexity and performance analyses for detecting actions in neighbouring sectors or estimating ATC workload.

Keywords—trajectory analysis; anomaly detection; ATC order detection; autoencoder

I. INTRODUCTION

The preparation and execution of training simulations for controllers and pilots relies on realistic environments which are based on common trajectory patterns, involving ATC orders typical of a context and traffic situations in a given area of interest. In the context of pilot simulations, the generation of relevant ATC communications consistent with traffic surrounding a simulated flight is a costly task requiring the involvement of an operational expert, if not an air traffic controller certified on this area. A decision support tool, able to generate realistic traffic situations and associated ATC actions and to adapt to the behaviour and actions of the pilot student could ease the process. Similarly, a tool assisting air traffic controllers, highlighting and suggesting required actions based on how peers would react in a similar situation could be a significant asset in ATC training.

Current state of the art regarding optimisation methods solving common ATC problems mainly focus on how to efficiently deconflict traffic, preferring optimality at the expense of realism of the inherent actions in a given context. Optimisation techniques have proven to be efficient at solving a wide range of conflicts at a tactical or strategic level: they are often illustrated with toy problem scenarios like the round-about or the wall problems, proven to be scalable to a large number of aircraft. Ensuring such methods are still efficient in corner-case situations is a necessary validation step. Nevertheless, it turns out to be difficult to relate to operational situations. In a context where realistic simulations are key for a safe and efficient training for pilot and controller students, such techniques require to be fine-tuned with operational feedbacks to guarantee that the generated orders resemble those an expert would have given in a similar context.

On the other hand, Machine Learning (ML) methods properly trained to learn from a dataset of historical situations seem appealing to generate realistic ATC orders. However, they require building a catalog of past ATC actions. Such a dataset is not easy to build from the observation of real traffic: flight plans are optimised at a strategic level; regulations are emitted upstream by ATC centers to avoid overload situations; and above all, no loss of separation is supposed to be observed as ATC process traffic based on surrounding aircraft, meteorological events and usual practices.

A first semi-supervised learning framework [1] combining speech and radar data to predict controller commands has been implemented using Automatic Speech Recognition. The method considers communication with ATC and its impact on relevant trajectories; research is still ongoing to improve the accuracy of speech recognition models applied to ATC communications [2]. We consider here a complementary approach, based only on flight tracks, without a systematic transcription of ATC communications.

ML methods are an excellent tool for detection and prediction on datasets displaying the more variance in the more limited scope. In order to get the most of such methods, we focus here on routes between city pairs, i.e. trajectories flying from city A to city B. We assume that most flights are executed according to their flight plans, where interactions with ATC would most likely be limited to standard clearances. By comparing how aircraft fly this route all year long, we assume that most ATC actions result in a deviation on the trajectory:

we therefore focus on deviations from the route drawn by most flights to infer ATC actions and try to understand their causes.

Filed flight plans (DDR files), ADS-B (possibly augmented with multilateration) trajectories, weather conditions (METAR) and even radio recordings (www.liveatc.net) are sources of information that can be used to assess the intent behind each ATC order, whether it resembles more a classic clearance, a deconfliction order placed to avoid a loss of separation, or a reaction to more unusual events. Different types of detected orders could support different types of simulations, requiring classical ATC actions or more specific actions related to more specific situations like thunderstorms, rerouting, runway change at destination airfield.

We present in the following an autoencoder-based anomaly detection method able to detect unusual trajectory patterns and to quantify the degree of abnormality of each outlier. Section II presents a literature review of the state of the art of ML techniques applied to aircraft trajectories. Section III focuses on the mathematics behind mentioned techniques. Section IV describes the studied dataset, including one year of traffic between three city pairs, mostly based on the OpenSky Network [3] platform; then explains how to prepare and format the data to train the autoencoder. In Section V, we attempt to put some trajectories detected as unusual into context and suggest plausible interpretations explaining why controllers could have processed the traffic that way. Section VI puts the results in perspective, showing how the method could be used both for our primary goal and for estimating probabilities of rare events in safety analyses.

II. RELATED WORK AND LITERATURE REVIEW

Research addressing ML techniques to tackle ATM problems is rather recent compared to their equivalent modelling and optimization problems, often solved by mixed-integer linear programming [4], constraint programming [5] or meta-heuristics [6]. ML techniques have then been used to learn ATC workload from past operations [7], to predict traffic flows above California [8] or to predict diversions in freight transportation [9].

ML brings a new perspective to ATM related problems: unsupervised machine learning focuses on detecting and describing how aircraft behave [10], [11]; supervised machine learning applies to labeled dataset and focuses on predicting aircraft trajectories [8], [9], [12]. Reinforcement learning addresses more traditional optimisation problems [13] where the search space may be explored through interactions with a simulator or based on a history of situations.

Unsupervised machine learning techniques have produced convincing results in terms of trajectory clustering [14], [15], with applications to flow prediction, anomaly detection and safety analyses. In particular, Principal Component Analysis has proved useful in its functional form to detect variational patterns with applications to final approaches [10] and climb profiles [11]. Multi-kernel anomaly detection (MKAD) [16] is arguably the current state of the art in the field of flight track anomaly detection, with convincing use cases on US

terminal maneuvering areas [17]. The method combines kernel functions (a measure of similarity) on discrete and continuous features in order to isolate a predefined percentage of anomalous trajectories.

We present in this paper a similar approach based on autoencoders, a particular kind of neural networks which recently made a name in anomaly detection. They have already been used to find breakpoints in time series [18] and to predict realistic transitions in sector configurations [19]. Autoencoders are comparable to compression methods. They are trained to reconstruct, i.e. compress then decompress, data (trajectories) passed in input. In practice, they learn to reconstruct most trajectories in a training dataset and fail to reconstruct the more atypical ones. Anomaly detection is based on the distance, the *reconstruction error*, between real trajectories and their reconstructed copies. The most notable difference with MKAD lies in the fact that no assumption needs to be made on the proportion of anomalies present in the dataset. Detected anomalies may be related to peculiar ATC situations and investigating them may also help refining safety models designed to assess risks and estimate probabilities of rare events.

III. AUTOENCODER-BASED ANOMALY DETECTION

A. Machine Learning Techniques for Anomaly Detection

Machine Learning techniques presented in the literature review address the unsupervised learning problem. This task infers the hidden structure behind unlabeled data. Instead of working on data with information known a priori about the context, including hazardous situations, we focus on a set of unlabeled trajectories, try to grasp their structure and eventually find outlying elements.

Principal component analysis (PCA) explains the variance in data by projecting trajectories on axes holding the more variance, determined by the eigenvalue of the covariance matrix of the samples. Variance ratio diagrams plot how much each component holds of the total variance. The user must then choose an appropriate number of components that describes the data best. Projecting the data on a limited number of components may reveal some structure in the data, sometimes call for some clustering analysis in order to understand and later detect particular behaviours and atypical elements falling outside emerging structures. t-distributed Stochastic Neighbour Embedding (t-SNE) is a similar dimension reduction and visualisation technique addressing non-linear transformations and which can improve the performance of classification methods.

One-class support vector machines (SVM) are extensively used for anomaly detection. SVM excel at learning decision boundaries. One-class SVM are designed to learn to separate nominal and anomalous data points based on a pairwise kernel (similarity) matrix and identify statistically significant anomalous examples. The kernel function defining the similarity between samples is crucial to the performance of the algorithm. A commonly chosen kernel is the radial basis function (rbf) but other kernels like the cosine similarity may yield better results depending on the case study. The rbf

calls for a hyperparameter σ that determines the width of the Gaussian distribution over the vector space. In addition, a hyperparameter ν must be provided by the user which corresponds to the maximum fraction of data assumed to be anomalous. Eventually anomalous samples are ordered by their distance to the separating hyperplane: a higher distance may statistically correspond to a higher degree of anomaly. It may happen that some outliers found by a one-class SVM are consistent with those found by clustering techniques applied on projections resulting from a PCA.

Multiple kernel methods are linear, non-linear or data-dependent combinations of several kernels [20]. The simplest and most common method is a weighted combination of kernels. In unsupervised ML problems such as anomaly detection, the resulting kernel is a convex combination of all kernels which are computed over multiple features. Each kernel is weighted based on the a priori knowledge of the expert on the study case. Matthews [17] and Das [16] use multiple kernel methods to detect potential safety anomalies in large databases of flight tracks consisting of discrete and continuous data.

B. Autoencoders

Autoencoders enter the artificial neural networks category. As the name suggests, they consist of two stages: encoding and decoding. A single-layer autoencoder is a kind of neural network consisting of only one hidden layer. Autoencoders aim at finding a common feature basis from the input data. They reduce dimensionality by setting the number of extracted features less than the input. Autoencoder models are usually trained by backpropagation in an unsupervised manner. The underlying optimization problem aims at minimizing the error of the reconstructed results from the original inputs.

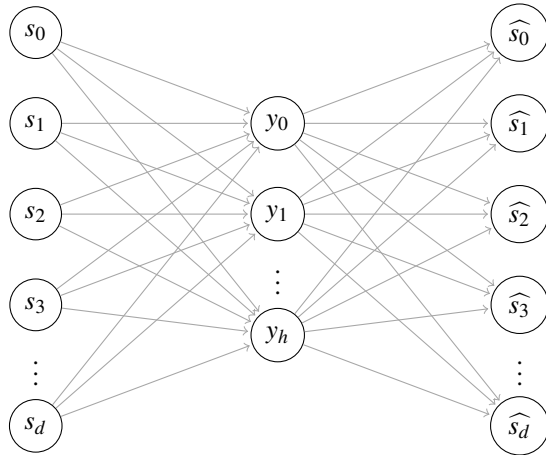


Figure 1. Autoencoder neural network architecture with one layer

The encoding function of an autoencoder (such as the one depicted on Fig. 1) maps the input data $s \in \mathbb{R}^d$ to a hidden representation $y \in \mathbb{R}^h = e(s) = g(w \cdot s + b)$ where $w \in \mathbb{R}^{d \times h}$ and $b \in \mathbb{R}^h$ are respectively the weight matrix and the bias vector and $g(\cdot)$ is a non linear activation function such as the sigmoid or hyperbolic tangent functions. The decoding

function maps the hidden representation back to the original input space according to $\hat{s} = d(y) = g(w' \cdot y + b')$, $g(\cdot)$ being most of the time the same activation function.

The objective of the autoencoder model is to minimize the error of the reconstructed result:

$$(w, b, w', b') = \operatorname{argmin} \ell(s, d(e(s))) \quad (1)$$

where $\ell(u, v)$ is a loss function decided according to the input range, typically the square loss:

$$\ell(u, v) = \|u - v\|^2 \quad (2)$$

or the cross-entropy loss:

$$\ell(u, v) = \sum_i u_i \cdot \log(v_i) + (1 - u_i) \cdot \log(1 - v_i) \quad (3)$$

IV. APPLICATION TO TRAJECTORY ANOMALY DETECTION

A. The Trajectory Dataset

Mode S has become one of the most important technologies in air traffic management as it supports the operation of secondary surveillance radar (SSR), traffic alert and collision avoidance systems (TCAS), and Automatic Dependent Surveillance–Broadcast (ADS-B). In practice, transponders in aircraft are selectively interrogated by sensors (radars) to provide situational awareness through the exchange of binary encoded information.

To be able to selectively interrogate aircraft, transponders aboard aircraft have been assigned a unique 24-bit identifier. The assignment of addresses is done by the national authority where the aircraft is registered. These identifiers are included in all Mode S messages and identify each aircraft.

Aircraft reply to ground sensor requests with messages of different types, called downlink formats (DF). We focus here on DFs 17 and 18 which are not transmitted upon interrogation. They contain all information needed to determine the aircraft's identity, location, and velocity. These squittered information are called Automatic Dependent Surveillance–Broadcast (ADS-B). Extended surveillance data from avionics such as intent- and status information is provided via Comm-B messages (DF 20, 21): such data could be of great help in future work to detect ATC actions on the indicated airspace since ADS-B only contains ground speed.

The OpenSky Network [3] is a crowd-sourced sensor network collecting air traffic data. The collected data used for this study contains only ADS-B data of specific callsigns from January to December 2017. We consider in the following three study cases, associated to three different city-pairs. The data we collected consists of one year worth of data on:

- 1) 28 different callsigns associated to flights from Paris Orly (LFPO) to Toulouse Blagnac (LFBO) airports (3536 trajectories);
- 2) 8 different callsigns associated to flights from Paris Charles-de-Gaulle (LFPG) to Frankfurt (EDDF) airports (2105 trajectories);
- 3) and 4 different callsigns associated to flights from Amsterdam Schiphol (EHAM) to Madrid Barajas (LEMD) airports (977 trajectories).

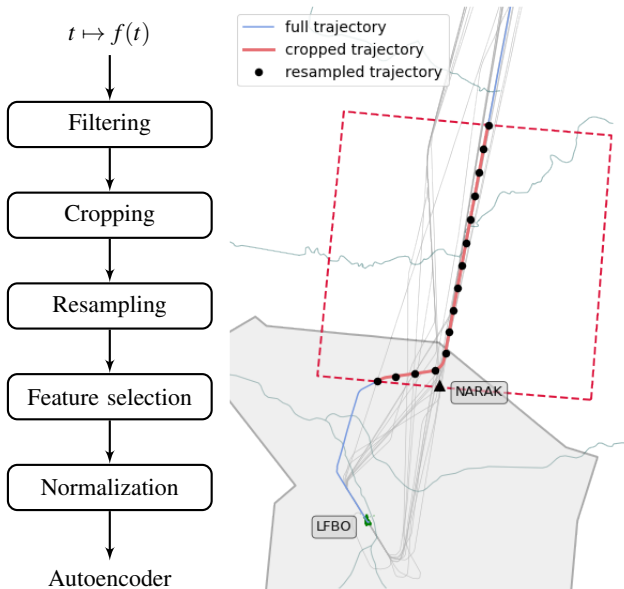


Figure 2. Data preparation: each trajectory is filtered, cropped, resampled; each feature is normalized before feeding the neural network.

		bottom-left coordinates	top-right coordinates
LFPO	LFBO	(44°17' N, 0°57' E)	(45°32' N, 2°45' E)
LFPG	EDDF	(49°9' N, 5°40' E)	(51°1' N, 7°40' E)
EHAM	LEMD	(41°18' N, 2°48' W)	(44° N, 0°37' W)

TABLE I

DEFINITION OF BOUNDING BOXES FOR THE THREE CASE STUDIES

In the following, we will refer to these scenarios respectively as the Toulouse, the Frankfurt and the Madrid scenarios.

B. Data preparation

Trajectories are mathematical objects used to describe the evolution of a moving object. They are described by a state vector with parameters $(x(t), y(t), \dots)$ that evolve in time. In practice, this state vector is only known at some sampled times. For clarity concerns, we name trajectory a sequence of recordings associated to an aircraft. We describe in this section the full chain of data pretreatments applied to each trajectory candidate in our dataset.

Trajectories from the OpenSky Network are unequally sampled and subject to errors happening in different steps of the acquisition chain: errors before the emission of data (imprecision in the positioning, quantification artifacts) and errors in the receiving and decoding of the data by feeders. A cascade of basic median filters is first applied to filter out irrelevant values from each individual trajectory.

For each scenario, the second step consists in defining a bounding box containing the set of trajectories. We chose to consider roughly drawn bounding boxes including most trajectories (definitions in Tab. I) before they enter the terminal maneuvering area (TMA, the greyed area in Fig. 2). We also chose to include in the bounding boxes the first navigational

aid (i.e. a set of geographical coordinates) of the standard arrival (STAR) procedures (NARAK in Fig. 2).

The next step in the data preparation consists in resampling the trajectories in the bounding box we selected. Since autoencoders have a fixed number of inputs (d in the example of Fig. 1), we resample each subset of trajectory cropped to our learning box so as to get only d equally distributed samples. Fig. 2 plots a resampling with $d = 15$ for illustrations' sake but we chose a larger number of samples $d = 150$ for our experiments. This choice is arbitrary and we found that other values of d in the same order of magnitude have no significant impact on the results.

Features are chosen among all data provided in the ADS-B specifications: latitude, longitude, GPS and barometric altitude, track angle, ground speed, vertical speed. Different airspeeds (CAS, IAS, TAS, etc.) are sent by aircraft upon request on DF 20 and 21 but for the sake of clarity, we chose to keep these features out of our dataset for future work beyond the scope of this paper. Controller's actions are most often expressed in terms of altitude (“climb to flight level 310”), track angle (“turn left heading 210”, “route direct NARAK”), and speed (“reduce speed to 160 kts”); since speeds are expressed in IAS and not in ground speed, we focused on track angles and altitude profiles.

Data normalization, also known as *feature scaling*, is widely used in machine learning in order to deal with variations in a wide range of shorter or larger intervals by standardizing the features. Lee [18] shows that feature scaling allow gradient descent algorithms to converge much faster than without it. Various methods can be used to rescale the range of features:

- *min-max normalization* rescales features in $[0, 1]$:

$$\tilde{x} = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (4)$$

has been used for the results presented in the following.

- *standardization* rescales features so that each feature has a zero mean and unit variance:

$$\tilde{x} = \frac{x - \mu(x)}{\sigma(x)} \quad (5)$$

C. Learning protocol

For the sake of clarity, we focus here on an analysis on normalized track angles. The input dimension, i.e. the number of neurons on both input and output layers of our autoencoder, has been set to $d = 150$. The embedding dimension, i.e. the number of neurons on the hidden layer has been set to a lower value of 64. All neurons are defined with a sigmoid activation function. The loss function used is the mean squared error as defined in (2), which compares vectors with their *reconstructions*, i.e. their images through the autoencoder. The learning rate is set to 10^{-2} and the training stops when the loss stops improving by more than 10^{-4} between two epochs. Fig. 3 plots the loss evolution on the Toulouse scenario.

As a result of our training process, we get a *reconstruction error*, i.e. a measure of the difference between a given trajectory and its autoencoded representation. Fig. 4 plots

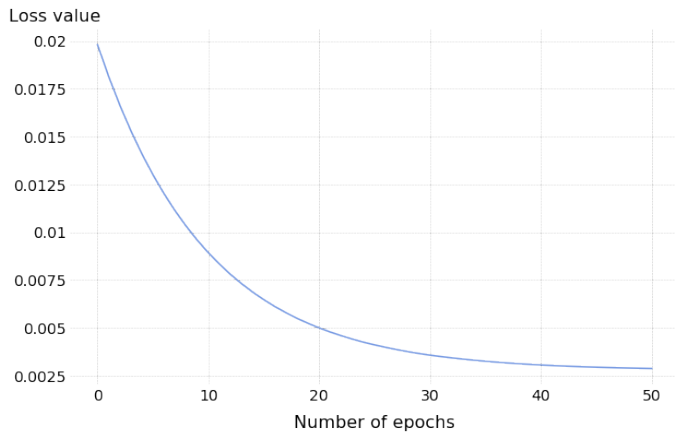


Figure 3. Evolution of the loss during the training process; example of the Toulouse scenario.

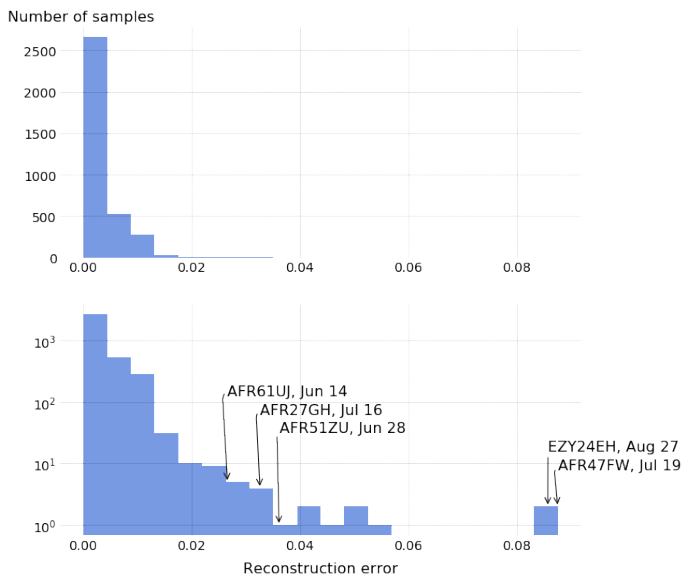


Figure 4. Distribution of reconstruction errors on the Toulouse scenario. The bottom distribution replicates the top distribution on a semi-log y-axis.

the distribution of these reconstruction errors. The model is trained to minimize the sum of all reconstruction errors, so the distribution is centered around zero.

The same distribution is plotted on a logarithmic y-axis to emphasize the few specific trajectories with higher reconstruction errors: we study in the following section the contextual situations associated to such specific trajectories pointed on the distribution. We focused first on trajectories with the highest reconstruction errors, then on a few situations with lower reconstruction errors, closer to the tail of the bell-shape distribution.

V. ANALYSIS OF ANOMALOUS TRAJECTORIES

The training phases on the three different scenarios yields reconstruction errors distributed as a half bell-shaped distribution centered on zero; few trajectories are ranked with higher values. (Fig. 4, 5 and 6) We focus our attention on trajectories

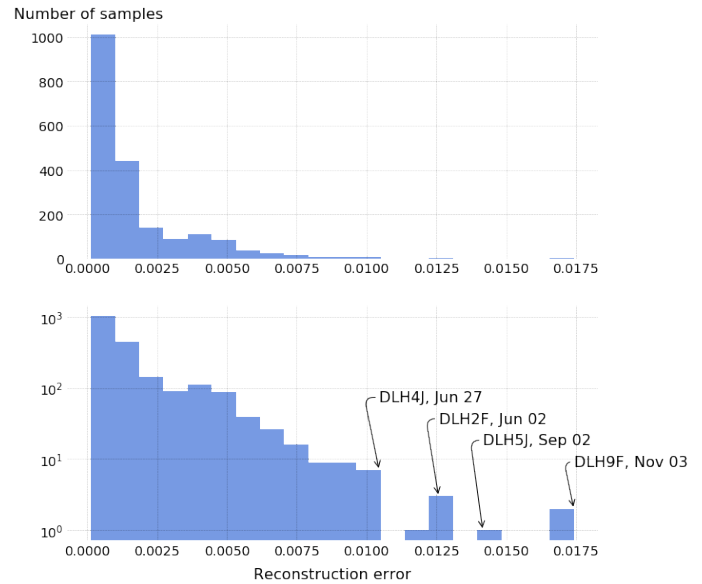


Figure 5. Distribution of reconstruction errors on the Frankfurt scenario. The bottom distribution replicates the top distribution on a semi-log y-axis.

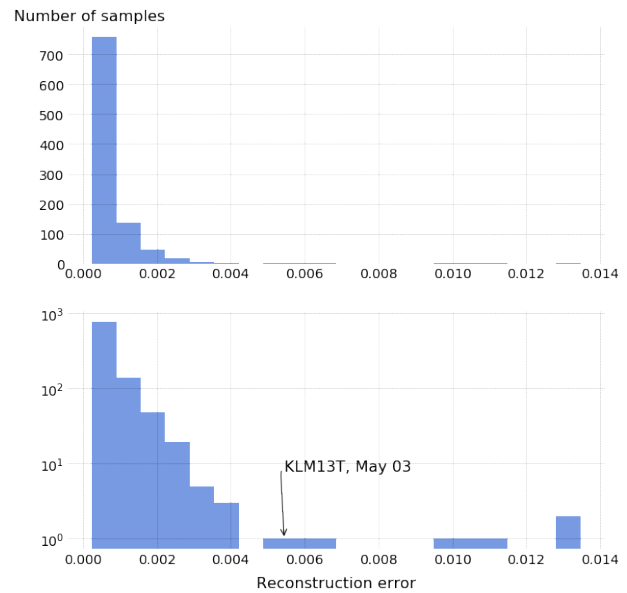


Figure 6. Distribution of reconstruction errors on the Madrid scenario. The bottom distribution replicates the top distribution on a semi-log y-axis.

associated to a higher reconstruction error value since they should be representative of the most unusual trajectories.

The analysis of the context was made after analysing traffic around anomalous detected trajectories. ADS-B may be an incomplete source of data for a thorough analysis of the situation since some aircraft are still not properly equipped. In that case, flight plans and trajectories inferred by multilateration give a hint but situations analysed in this paper have been selected so as to be representative of the wide panel of detected situations and to be explainable with ADS-B tracks openly accessible. A perfect analysis would have also involved confirmation of

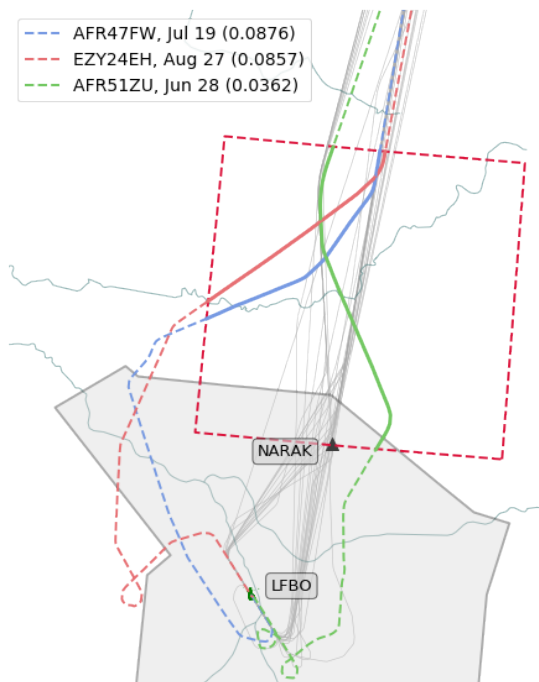


Figure 7. Situations with high reconstruction errors on the Toulouse scenario

our hypotheses with radio recordings; unfortunately, none were available for the chosen scenarios.

In general, we found that very high reconstruction errors are associated to less common situations; explanations are more to be found in the METAR history or regulation history. Conversely, reconstruction errors closer to the tail of the bell-shaped distribution, are more prone to yielding nominal situations which may be explained by ATC orders issued for a deconfliction or sequencing purpose.

A. Highest values of reconstruction errors relate to less common situations

1) *Traffic interruption:* Flight AFR51ZU on June 28th (Fig. 7) yields a high (although not the topmost) reconstruction score and its peculiar route on final approach called the authors' attention. This day was marked with a lot of delays because of weather. METAR on that day is particularly explicit: thunderstorm (TS), presence of cumulonimbus (CB) and a gusting wind forecast (20G35KT).

```
LFBO 281600Z AUTO 20006KT 180V240 9999 TS
FEW033/// SCT047/// BKN060/// ///CB 20/14
Q1004 TEMPO 28020G35KT 2000 TSRA=
```

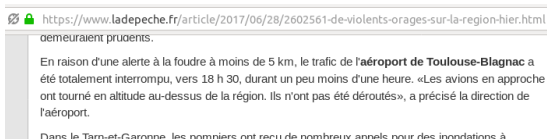


Figure 8. Archives of local news: traffic was interrupted in Toulouse airport around 18:30 local time. No aircraft were diverted during the interruption.

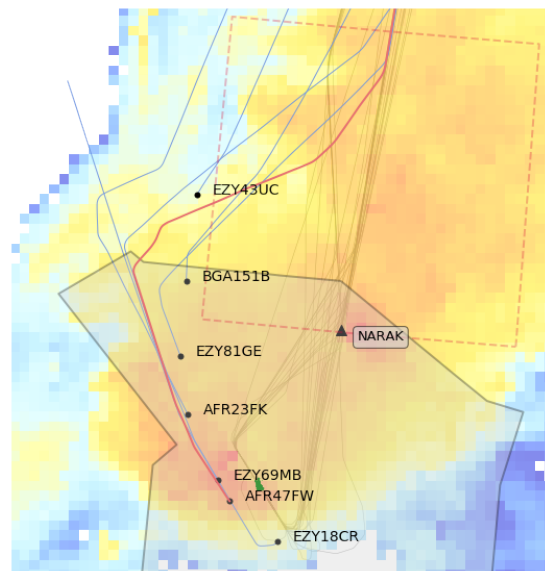


Figure 9. Weather impact on STAR procedures on the Toulouse scenario. The orange background shows the location of cumulonimbus at 21:09 UTC.

Local news (Fig. 8) reported that traffic was interrupted on that day when AFR51ZU was in approach. This interruption may explain the two loops during final approach.

2) *Weather impact on STAR procedures:* Traffic around Toulouse airport on July 19th was also impacted by cumulonimbus located on the STAR procedure, around the NARAK beacon. All aircraft (including AFR47FW, see Fig. 9) coming from the North-East were deviated to the West prior to entering the TMA, so as to be sequenced on the usual STAR procedures applicable to aircraft coming from the North-West.

The background on Fig. 9 has been produced with data from the Spinning Enhanced Visible and Infrared Imager (SEVIRI) collected by the Meteosat Second Generation series of satellites. Thermal IR data has been used to estimate the location of cumulonimbus, with red areas indicating the most severe convection.

3) *QFU change:* QFU refers to the magnetic heading of the runway in use. ATC may change the QFU at any time depending on the weather conditions. On August 27th, EZY24EH took a very peculiar route (Fig. 7) as other aircraft were landing from the south on QFU32. After the last flight has landed (16:51), a first aircraft landed at 17:08 on QFU14 and EZY24EH was on hold before landing 3 minutes later.

4) *Avoiding regulations:* Many regulations were in place between Paris and Frankfurt (Reims and Langen ACC) in early evening on June 2nd, 2017, because of cumulonimbus in the area. In particular, DLH2F was impacted by a 15 minute delay before departure because of a regulation filed by Frankfurt arrivals. Figure 11 plots the last filed flight plan (Filed Tactical Flight Model, FTFM) and the Current Tactical Flight Model (CTFM), refined version of the FTFM based on live positions for DLH2F. Such a pattern suggests that DLH2F adapted its route to avoid further regulations in the area.

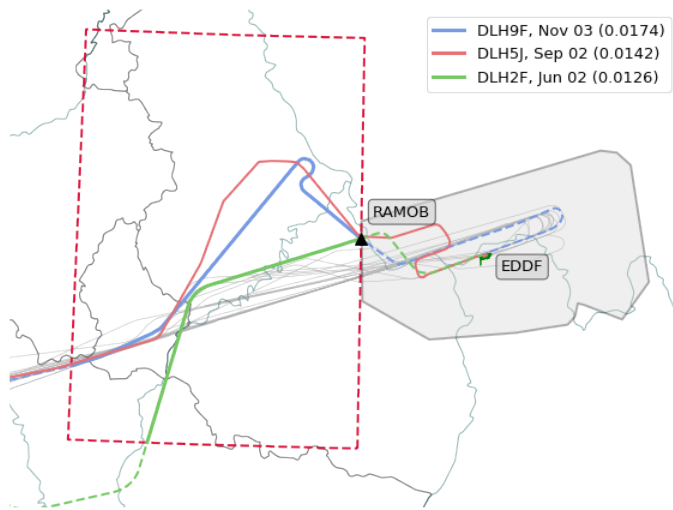


Figure 10. Solutions with high reconstruction errors on the Frankfurt scenario

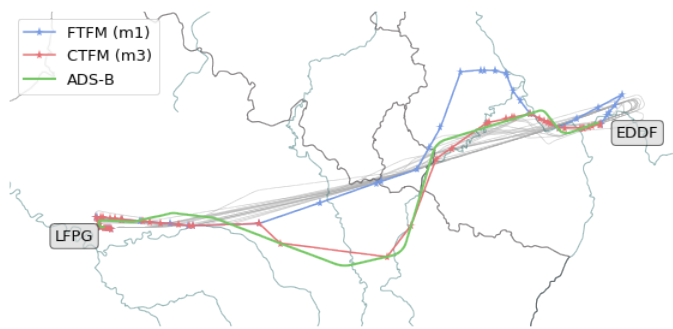


Figure 11. Last filed flight plan and trajectory of DLH2F on June 2nd, 2017, on the Frankfurt scenario

B. High-but-not-highest values of reconstruction errors yield more conventional ATC situations

In contrast, we found more classical and conventional ATC situations close to the right-hand side of the bell-shape of the reconstruction error distributions.

1) *Arrival sequencing:* On Figure 12, AFR27GH has a higher reconstruction error, probably because its trajectory is pushed quite to the East of the NARAK beacon, a less usual pattern for trajectories on the LFPO-LFBO route. As we investigate closer into the situation, it appears that AFR27GH was flying behind EYZ81GE (from Lille, to the North) before being instructed to turn left. As EYZ743L arrived from Lyon (to the North-East), AFR27GH was sequenced behind with an appropriate ATC order. RYR3YM arrives next and is sequenced behind AFR27GH in a similar manner.

2) *Deconfliction:* On Figure 12, DLH4J flies into Frankfurt area. The density of traffic converging on this IAF at this time of the day would probably not explain this shift in trajectory. The explanation could come from TAP571 which took off a bit earlier from runway 18. Without a turn left order to DLH4J, her climb path could have crossed DLH4J's trajectory, probably causing a loss of separation.

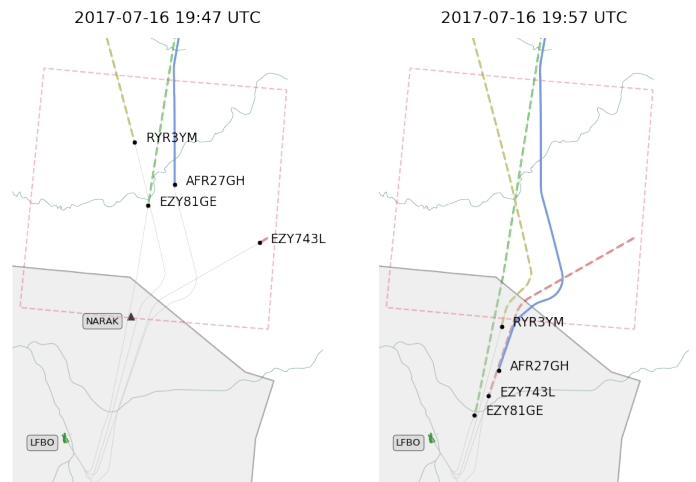


Figure 12. Aircraft are sequenced for landing before entering Toulouse TMA; AFR27GH is vectored behind EYZ81GE and EYZ743L.

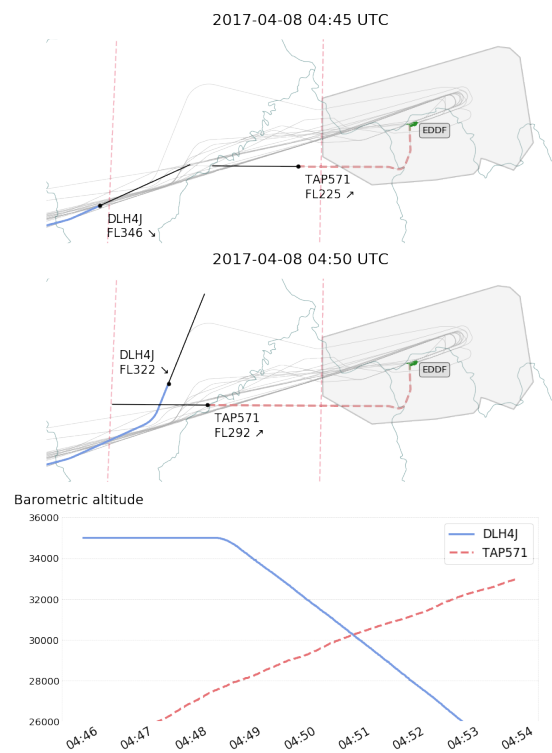


Figure 13. Deconfliction between DLH4J scheduled for landing and TAP571 taking off from Frankfurt airport, runway 18.

Similarly, Figure 14 plots a deconfliction between AFR61UJ landing in Toulouse and ISS3932 bound for New York. The ATC order enabled AFR61UJ to descend through the level of ISS3932 without conflict.

3) *Delaying:* On Figure 15 both KLM13T and IBK3ME come to Madrid, resp. from Amsterdam and Oslo. In this situation, KLM13T comes before and flies two levels below IBK3ME. About 15 minutes before entering the TMA, KLM13T is delayed by a maneuver to the left so that IBK3ME can overtake her.

ACKNOWLEDGMENTS

The authors would like to thank Simon Proud from Oxford University for his help with the satellite image on Fig. 9 and Enrico Spinielli from Eurocontrol for his help in analysing recorded flight plans and regulations related to Fig. 11.

REFERENCES

- [1] A. Srinivasamurthy, P. Motlicek, M. Singh, Y. Oualil, M. Kleinert, H. Ehr, and H. Helmke, "Iterative learning of speech recognition models for air traffic control," in *Proceedings of the International Conference on Spoken Language Processing*, 2018, pp. 3519–3523.
- [2] S. V. Subramanian, P. F. Kostuik, and G. Katz, "Custom IBM Watson speech-to-text model for anomaly detection using ATC-pilot voice communication," in *Proceedings of the Aviation Technology, Integration, and Operations Conference*, 2018, p. 3979.
- [3] M. Schäfer, M. Strohmeier, V. Lenders, I. Martinovic, and M. Wilhelm, "Bringing up OpenSky: A large-scale ADS-B sensor network for research," in *Proceedings of the 13th international symposium on Information processing in sensor networks*, 2014, pp. 83–94.
- [4] S. Cafieri and N. Durand, "Aircraft deconfliction with speed regulation: new models from mixed-integer optimization," *Journal of Global Optimization*, vol. 58, no. 4, pp. 613–629, 2014.
- [5] C. Allignol, N. Barnier, P. Flener, and J. Pearson, "Constraint programming for air traffic management: a survey," *The Knowledge Engineering Review*, vol. 27, no. 3, pp. 361–392, 2012.
- [6] N. Durand, J.-M. Alliot, and J. Noailles, "Automatic aircraft conflict resolution using genetic algorithms," in *Proceedings of the symposium on Applied Computing*. ACM, 1996, pp. 289–298.
- [7] D. Gianazza, "Forecasting workload and airspace configuration with neural networks and tree search methods," *Artificial intelligence*, vol. 174, no. 7-8, pp. pp-530, 2010.
- [8] Y. Lü, Y. Duan, W. Kang, Z. Li, F.-Y. Wang *et al.*, "Traffic flow prediction with big data: A deep learning approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 865–873, 2015.
- [9] C. Di Ciccio, H. Van der Aa, C. Cabanillas, J. Mendling, and J. Prescher, "Detecting flight trajectory anomalies and predicting diversions in freight transportation," *Decision Support Systems*, vol. 88, pp. 1–17, 2016.
- [10] X. Olive and P. Bieber, "Quantitative assessments of runway excursion precursors using Mode S data," in *Proceedings of the International Conference for Research in Air Transportation*, 2018.
- [11] F. Nicol, "Functional principal component analysis of aircraft trajectories," in *Proceedings of the 2nd International Conference on Interdisciplinary Science for Innovative Air Traffic Management*, 2013.
- [12] Y. Liu, M. Hansen, D. J. Lovell, and M. O. Ball, "Predicting aircraft trajectory choice – a nominal route approach," in *Proceedings of the International Conference for Research in Air Transportation*, 2018.
- [13] M. Brittain and P. Wei, "Autonomous aircraft sequencing and separation with hierarchical deep reinforcement learning," in *Proceedings of the International Conference for Research in Air Transportation*, 2018.
- [14] M. Gariel, A. N. Srivastava, and E. Feron, "Trajectory clustering and an application to airspace monitoring," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1511–1524, 2011.
- [15] M. Conde Rocha Murca, R. DeLaura, R. J. Hansman, R. Jordan, T. Reynolds, and H. Balakrishnan, "Trajectory clustering and classification for characterization of air traffic flows," in *Proceedings of the 16th Aviation Technology, Integration, and Operations Conference*, 2016.
- [16] S. Das, B. L. Matthews, A. N. Srivastava, and N. C. Oza, "Multiple kernel learning for heterogeneous anomaly detection: algorithm and aviation safety case study," in *Proceedings of the 16th international conference on Knowledge discovery and data mining*, 2010, pp. 47–56.
- [17] B. Matthews, D. Nielsen, J. Schade, K. Chan, and M. Kiniry, "Automated discovery of flight track anomalies," in *Proceedings of the 33rd Digital Avionics Systems Conference*, 2014.
- [18] W.-H. Lee, J. Ortiz, B. Ko, and R. Lee, "Time series segmentation through automatic feature learning," *arXiv:1801.05394*, 2018.
- [19] T. Dubot, "Predicting sector configuration transitions with autoencoder-based anomaly detection," in *Proceedings of the International Conference for Research in Air Transportation*, 2018.
- [20] M. Gonen and E. Alpaydin, "Multiple kernel learning algorithms," *Journal of Machine Learning Research*, pp. 2211–2268, 2011.

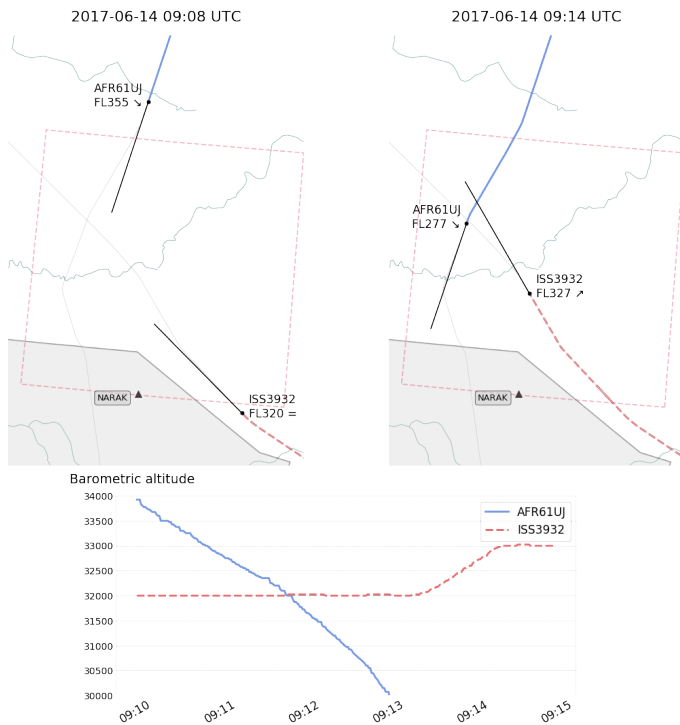


Figure 14. Deconflicting turn right actions on AFR61UJ, landing in Toulouse, and ISS3932, en route on FL300.

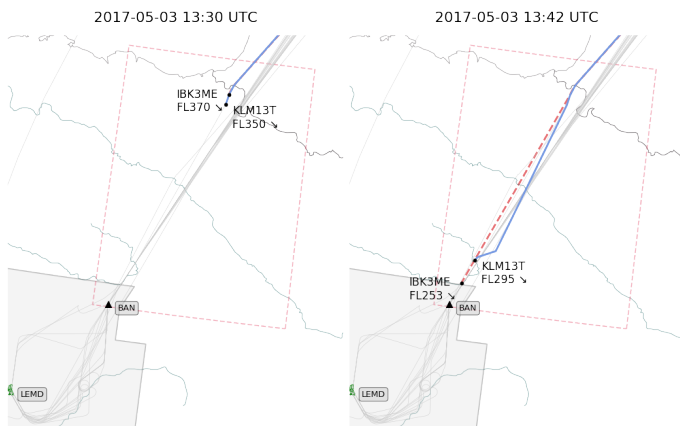


Figure 15. In the Madrid scenario, IBK3ME flies above and behind KLM13T but enters Madrid TMA before her.

VI. CONCLUSIONS

This paper presented an anomaly detection technique well suited for identifying uncommon situations and ATC orders in Mode S data. We believe that this method is well suited to build a dataset of real situations with suggestions of plausible resolution actions. Studying uncommon or late ATC actions could also be a way to estimate ATC workload or to refine safety models and probability estimations associated with incident-prone situations. Future works could consider more evolved structures of networks, then include flow identification and outlier detection techniques inside a control sector so as to prescind from the limitation of the city pair paradigm.