TRAJECTORY BASED FLIGHT PHASE IDENTIFICATION WITH MACHINE LEARNING FOR DIGITAL TWINS

E. Arts*, A. Kamtsiuris*, H. Meyer*, F. Raddatz*, A. Peters†, S. Wermter†
* DLR Institute for Maintenance, Repair and Overhaul, Hein-Sass-Weg 22, 21129 Hamburg, Germany
† University of Hamburg, Department of Informatics, Knowledge Technology, Vogt-Koelln-Strasse 30, 22527 Hamburg, Germany

Abstract

Analysis of aircraft trajectory data is used in different applications of aviation research. Areas such as Maintenance, Repair and Overhaul (MRO) and Air Traffic Management (ATM) benefit from a more detailed understanding of the trajectory, thus requiring the trajectory to be divided into the different flight phases. Flight phases are mostly computed from the aircraft's internal sensor parameters, which are very sensitive and have scarce availability to the public. This is why identification on publicly available data such as Automatic Dependent Surveillance Broadcast (ADS-B) trajectory data is essential. Some of the flight phases required for these applications are not covered by state-of-the-art flight phase identification on ADS-B trajectory data.
This paper presents a novel machine learning approach for more detailed flight phase identification. We generate a training dataset with supervised simulation data obtained with the X-plane simulator. The model combines K-means clustering with a Long Short-Term Memory (LSTM) network, the former allows the segmentation to capture transitions between phases more closely, and the latter learns the dynamics of a flight. We are able to identify a larger variety of phases compared to state of the art and adhere to the International Civil Aviation Organisation (ICAO) standard.

Keywords

Machine Learning; Neural Networks; Flight Phase; Long Short-Term Memory; K-means; Maintenance, Repair and Overhaul

Acronyms

ADREP Accident Data Reporting.
ADS-B Automatic Dependent Surveillance Broadcast.
APR approach.
ATM Air Traffic Management.

CLI climb.
CRZ cruise.

DST descent.

EPR Engine Pressure Ratio.

GMM Gaussian Mixture Model.

IATA International Air Transport Association.
ICAO International Civil Aviation Organisation.
ICL initial climb.

LDG landing.
LSTM Long Short-Term Memory.

MRO Maintenance, Repair and Overhaul.

NLLL Negative Log Likelihood Loss.
NODE Neural Ordinary Differential Equations.

SVM Support Vector Machine.

TOF take-off.
TXI taxi.

1. INTRODUCTION

An aircraft mission consists of multiple flight phases. Each flight phase has an impact on the aircraft's parameter state. Therefore various stakeholders in aircraft design and after-sales services are interested in understanding the flight missions of the aircraft better to derive a deeper understanding, e.g. of degradation mechanisms. The degradation of an aircraft system can vary throughout a mission, as such, Maintenance, Repair and Overhaul (MRO) institutes need the different flight phases of the trajectory to allow for degradation prognostics [1].
With growing amounts of data collected and distributed, the need for automated classification algorithms rises. Such flight phase classification algorithms can be be integrated in bigger simulation frameworks such as digital twins of assets, which comprise various additional data processing and

doi: 10.25967/550191

simulation functionalities [2].
Air Traffic Management (ATM) uses flight phases in trajectory prediction [3], which is essential for traffic flow prediction at airports and accident avoidance [4]. The analysis and reduction of environmental impact also often uses flight phases [5–7].
The flight phase variable is either part of, or computed from the aircrafts internal data which is very sensitive and has scarce availability to the public. This is not the case for aircraft trajectory data, which is broadcasted through the Automatic Dependent Surveillance Broadcast (ADS-B) technology. Although there are multiple definitions of flight phases, the most commonly used are given by the International Civil Aviation Organisation (ICAO) Accident Data Reporting (ADREP) and International Air Transport Association (IATA) [8, 9] which are rather similar to each other, except for the standing phase.
To allow the use of ADS-B data in a larger variety of contexts Sun et al. [10] present a model that augments the ADS-B trajectory data with flight phases. This model focuses on the processing of large-scale data at the expense of flight phase completeness. As such, it only identifies flight phases of larger grain (taxi, climb, level, cruise, and descent) which do not adhere to any recognised standard. This excludes the use of this model for applications that require more fine-grained phases or an alignment with other data sources that use flight phases, such as accident databases.
In this work we identify the following phases based on the ICAO standard[1]: taxi (TXI), take-off (TOF), initial climb (ICL), climb (CLI), cruise (CRZ), descent (DST), approach (APR), and landing (LDG). The identification is performed through a supervised machine learning architecture to create a model trained on simulation data. The architecture relies on the following steps: preprocessing the data, segmenting the data, classifying each segment. During the preprocessing of the training data, the aircrafts internal parameters are used to label the trajectory data with the correct flight phases, and the data is trimmed to allow for identification on partial flights. For the classification, each flight trajectory is divided into 160 segments relying on K-means clustering, this makes sure that the borders of segments correspond to moments of large amount of change. Every segment is then classified with a Long Short-Term Memory (LSTM) which receives the sequence of segments that make up a flight.
In the following chapters we will first elaborate on the machine learning algorithms used and other models that use machine learning for flight phase estimation (section 2), after which we will describe the details of the labels, data used, and approach used (section 3), report on the results (section 4)), and discuss our findings (section 5).

[1] https://www.skybrary.aero/index.php/Flight_Phase_Taxonomy accessed on 10/08/2021

2. RELATED WORK

Before proceeding with the details of the flight phase identification method developed, we give an introduction to the machine learning algorithms used and other approaches to flight phase identification with machine learning.

2.1. Machine Learning Algorithms Used

Machine learning algorithms are algorithms that learn from experience [11]. For the architecture described in this paper, two well-established machine learning algorithms are used: K-means clustering and LSTM neural networks.
Clustering is an unsupervised learning task and consists of grouping similar data together in the same cluster and putting dissimilar data in different clusters. The K-means algorithm does this by assigning data points to the cluster with the nearest mean. It starts by randomly assigning each data point to one of the K clusters, where K is a predetermined number. It then proceeds by alternating the following two steps until convergence:

- **Assigning** each data point to the cluster with the nearest mean.
- **Update** the clusters' means based on the newly assigned data points that compose them.

The algorithm converges when no points are assigned to different clusters, in comparison to the previous iteration or until the predetermined maximum number of iterations is reached. We use K-means to divide a single flight into different segments in order to assign a flight phase to each of the segments.
These segments are classified using a LSTM [12, 13], a neural network that is known for its good performance on sequences. A neural network is a computational architecture that consists of layers of nodes, connected to each other by weighted edges. These weights are learned during training through the network's feedback. LSTMs are a popular type of recurrent neural networks. Recurrent neural networks have the addition of a memory unit that allows to produce an output of a single data point based also on information acquired from its predecessors in the sequence, this aids the network to learn the temporal relations of a flight.

(a) Trajectory variables used in related work

state variable	Sun et al. [10]	Liu et al. [14]	Kovarik et al. [3] Paglione et al. [15]	this work
Altitude	X	X	X	X
Speed	X			X
Rate of Climb	X	X		X*
XY coordinates			X	
Pitch, Roll and True Heading angle		X		
Power Lever Angle		X		
Engine Fan Speed		X		

* Rate of Climb is computed from altitude

(b) Flight phases identified by related work

flight phase	Sun et al. [10]	Liu et al. [14]	Kovarik et al. [3] Paglione et al. [15]	this work
ground / taxi	X	X	X	X
take-off				X
climb	X	X	X	X*
cruise	X	X**	X**	X
level	X	X**	X**	*
descent	X	X	X	X*
turn			X***	
landing				X

* climb is further divided into initial climb and climb, descent is divided into descent and approach, level is considered part of climb or descent
** cruise and level are considered the same phase
*** in a separate model

TAB 1. Comparison of variables used and phases identified by related work and this work.

2.2. Flight Phase Identification

As flight phases are defined on board the aircraft by the pilot or by internal system parameters [16] they are mostly unavailable on a large scale because of the confidentiality of these sources. This leads to the need to identify them externally either online or offline. Recent research shows that machine learning approaches outperform previous statistical modelling approaches for this task [3]. Many models, however, only provide a reduced number of phases, as can be seen in TAB 1b. The aim of this work is to provide an identification tool that classifies the phases in ADS-B trajectory data. Of the publications referenced in tables 1a and 1b, the approach by Sun et al. [10] is the only one that uses ADS-B data. Their model divides the flights into segments of fixed lengths and classifies them using fuzzy logic [17]: fuzzy logic allows to introduce partial truth in Boolean logic. As such, the segments are classified using ranges of values for the altitude (H), speed (V), and rate of climb (RoC) given for each phase, defined as follows:

- $Ground = H_{ground} \wedge V_{low} \wedge RoC_{zero}$
- $Climb = H_{low} \wedge V_{medium} \wedge RoC_{positive}$
- $Cruise = H_{high} \wedge V_{high} \wedge RoC_{zero}$
- $Descent = H_{low} \wedge V_{medium} \wedge RoC_{negative}$
- $Level\ flight = H_{low} \wedge V_{medium} \wedge RoC_{zero}$

A different model for flight phase identification with machine learning has been presented by Liu et al., using a Gaussian Mixture Model (GMM) clustering [14]. This unsupervised model offers nearly the same phases as the previously discussed approach by Sun et al., but rather than trajectory data, this model also requires aircraft internal data such as: pitch angle, roll angle, true heading angle, power lever angle, and engine fan speed.

There are also approaches that aim at predicting the phase of flight as part of trajectory prediction. Kovarik et al. [3] compared 3 machine learning models to a simple regression model proposed by Paglione et al. [15]: Support Vector Machine (SVM), LSTM, and Neural Ordinary Differential Equations (NODE). They found that the LSTM model performed best among the 3 analysed. The LSTM model consists of two separate networks that predict the next horizontal or vertical flight phase one step ahead. The horizontal flight phases consist of straight and turn, however, the vertical flight phases consist of ascending, descending, and level flights. These flight phases were predicted from XY coordinates and altitude.

In this work we combine the approach of clustering and LSTM. Firstly, similar values are grouped together, as similar values often belong to the same flight phase. In contrast to previous work, more clusters than the number of flight phases are identified and kept continuous in time to give them to a LSTM. The LSTM learns the sequential dependencies of these clusters and identifies flight phases with potentially multiple clusters. The details of this implementation are provided in the next section.

3. METHODS

At the foundation of our architecture lies the supervised learning paradigm. With the help of internal aircraft parameters we create a labeled training dataset of trajectories labeled with the ICAO defined flight phases. The model uses an algorithm called K-means segmentation to transform the trajectory data into fixed length sequential data. These sequences, that represent single flights, are then classified by a LSTM.

3.1. Flight Phases

The ICAO ADREP flight phase taxonomy is adopted to allow alignment with a widespread standard [9]. We focus on commercial powered fixed-wing aircraft flight phases, since this covers a big portion of performed flights. Additional sub-phases are part of the ICAO flight phase taxonomy, but are not included at this stage of our work. Our approach uses the visual flight rule definitions and limit criteria of flight phases. The standing phase is not included in this work and some sub-phases were combined. TAB 2 shows ICAO's primary and secondary phases of flights and the derived flight phases used in this work.

3.2. The Data

To obtain sufficiently large training and testing datasets, flights are generated using the flight sim-

ICAO primary phases	ICAO sub-phases	Phase used for classification method
Standing	-	-
Taxi	...	Taxi (TXI)
Take-off	**Take-off run**	Take-off (TOF)
	Initial Climb	Initial climb (ICL)
En Route	**Climb to Cruise**	climb (CLI)
	Cruise	Cruise (CRZ)
	Change of Cruise Level	
	Descent	Descent (DST)
Approach	...	Approach (APR)
Landing	**Level off-touchdown**	Landing (LDG)
	Landing Roll	

TAB 2. Flight phases addressed by classification method.

ulation software X-Plane[2]. This kind of flight data, compared to ADS-B data, comprises an additional range of parameters from different aircraft systems, such as the landing gears or the engines. Using these parameters, an algorithm is developed to automatically label large numbers of flights.
Furthermore, using trajectory simulation software allows for coverage of various types of aircraft trajectories. This enables tests of different scenarios. On the other hand, there are also differences between simulated data and real data: in our case we noticed an increased smoothness in the variables and a tendency towards shorter flights.
The rules for flight phase identification are derived from the ADREP taxonomy criteria and are shown in TAB 3. Generally, the algorithm relies on a typical order of flight phases, which is described by the relation between the timestamp ts and a timestamp of an arbitrary flight phase $ts(flightphase)$, as shown in TAB 3.
Flight phases on the ground are mainly identified by the $gear\_compr$ parameter, which indicates if there is a normal force on the landing gear. TOF is defined by the maximum Engine Pressure Ratio (EPR) commanded. A lower threshold of 80% is added to avoid sensor noise sensitivity. The phase ends if an altitude of $35\,ft$ is reached or the command to retract the landing gear is given.
CRZ is defined by a rate of climb around zero and an altitude close to the maximum altitude in the flight data given. Furthermore, this phase needs to be longer than a defined lower threshold, in order to exclude short level flight phases during climb and descent.
DST is identified by a negative rate of climb which lasts longer than a defined minimum time span. LDG starts with the flare motion shortly before the landing gear touches the ground and ends either if the aircraft comes to a stop, or leaves the runway, which is indicated by the steering angle $steer\_ang$.

3.3. The Architecture

The architecture developed for the flight phase identification consists of two main steps: the segmentation of the flight, and the classification of the segments after each segment is translated into features.

[2] https://www.x-plane.com/

Flight Phase	Rule
TXI	$(ts < ts(TOF) \vee ts > ts(LDG)) \wedge$ $\wedge gear\_compr \wedge tgt\_epr > 0$
TOF	$alt < 35\,ft \wedge gear\_lvr\_down \wedge cmd\_epr \cdot tgt\_epr > 0.8 \cdot max(tgt\_epr)$
ICL	$ts > ts(TOF) \wedge$ $\wedge 35\,ft < alt < 1000\,ft$
CLI	$ts > ts(ICL) \wedge ts < ts(CRZ)$
CRZ	$-500\,fpm < roc < 500\,fpm \wedge$ $\wedge ((alt > max(alt) - 1000\,ft) \vee (alt > 1000\,ft \wedge (ts_{end} - ts_{begin}) > 360\,s))$
DST	$ts < ts(APR) \wedge$ $\wedge (ts_{begin} - ts_{end}) > 120\,s \wedge roc < -10\,fpm$
APR	$roc < -10\,fpm \wedge alt < 1000\,ft$
LDG	$ts > ts(TOF) \wedge$ $\wedge ts_{begin} - ts(gear\_compr) < 5\,s \ \wedge \ (abs(steer\_ang) < 3\,^\circ \vee spd = 0\,kts)$

TAB 3. The rules applied for the rule based flight phase identification from aircraft variables. Variables used: timestamp (ts), barometric altitude (alt), ground speed (spd), altitude rate (roc), target take-off EPR (tgt_epr), commanded take-off EPR (cmd_epr), landing gear deployment ($gear\_lvr\_down$), force on main landing gear ($gear\_compr$), nose gear steering angle ($steer\_ang$).

For the segmentation a variation of the K-means clustering algorithm [18] is used, while for the classification a LSTM [13] is used with a loss penalty function.

3.3.1. Segmentation

The first step in the architecture is the segmentation of the flight, which consists of dividing it into a fixed number of segments. This is achieved by using a variation of the K-means algorithm, further referred to as K-means segmentation.
K-means segmentation initializes segments by dividing the input into equal parts, after which it allows the edge points of segments to either belong to their current segment or the neighboring one, based on their distance to the segment means and if their current cluster has at least 4 points belonging to it. When two neighboring edges both try to change their cluster of belonging, only the edge that has a bigger difference in distance between the two means is allowed to do so. The hyperparameters of this algorithm are the number of clusters and the maximum number of iterations:

- The number of clusters is found empirically with the right trade-off between size and the minimum error introduced, it is set to 160.
- The maximum number of iterations is set by taking the $95th$ percentile of iterations until convergence which corresponds to 100 iterations.

The result of the algorithm are clusters that represent the segments that are continuous with respect to time and vary in size with the amount of change over time. The aim of this segmentation is to be able to reduce the error introduced by phases overlapping in a single segment. A change of flight phase most likely occurs at a point where there is more change, i.e., where there is a bigger distance between two consecutive data points. In section 4, we compare the use of K-means segmentation with uniform segmentation, which does not perform the K-means iterations and as such divides the flight into equal length segments.

3.3.2. Features

After the flight has been divided into segments, these segments are given to the classification network, this is achieved by extracting features from each segment. As has been mentioned previously, the aircraft state variables given to the model are altitude, speed, and rate of climb, together with time. Other models [3,10] also used the rate of climb and XY coordinates. The latitude and longitude provided in the ADS-B data can be transformed into XY coordinates [19]. However, since both XY coordinates and latitude and longitude are referenced to the earth, for the network to be able to generalize flights from different airports, the origin coordinates are required. ADS-B flights, however, are often incomplete which means that the origin coordinates are not always available. It would be possible to retrieve the coordinates from the origin airport, but this is out of the scope of this work. To circumvent this issue, the latitude and longitude could be used taking their difference over time, yet this would correspond to using the speed, as such, these variables are excluded in this work.
The following features given to the network are computed for each segment provided by K-means segmentation:

1) length of the segment (n)
2) initial altitude (alt_0)
3) final altitude (alt_n)
4) initial speed (spd_0)
5) final speed (spd_n)
6) initial rate of climb (roc_0)
7) final rate of climb (roc_n)

Each of these features is normalised. For the segment length, this consists in subtracting the preset minimum value 4 and dividing by 1800, a manually introduced limit. The altitude- and speed-related features are normalised by dividing each of them by the maximum value of that feature in each flight. For the rate of climb features, this consists in subtracting the minimum value of that feature in the flight it belongs to and subsequently dividing it by the difference of maximum and minimum. The altitude and speed are always positive, and if a flight is incomplete, the minimum altitude or speed might not correspond to the value of these in a complete flight.

3.3.3. Classification

The features extracted from the segments, as explained previously, are used by the classification network that labels each segment with a flight phase. For this purpose, a LSTM is chosen because of its ability to capture longer temporal relations. The LSTM receives the features of the segments as inputs and learns to classify the clusters according to the prevailing label of that cluster. The network is implemented in Pytorch [20] and consists of an input layer, 2 layers of 16 LSTM cells, followed by an activation layer consisting of the logarithm of a softmax function. The output is a value between 0 and 1 for each of the classes, where the output value 1 indicates the segment belongs to that class and the output value 0 indicates that it does not belong to that class. For training, a batch size of 16 and a Negative Log Likelihood Loss (NLLL) are used for stochastic gradient descent. The hyperparameters *number of layers*, *number of hidden units*, and *batch size* have been found with an initial hyperparameter gridsearch. Initial evaluation of the model shows that not every flight phase is identified with the same accuracy, the shorter flight phases present more inaccuracies.
For this reason, a penalty term (equation (1)) is added to the loss function, an approach to class imbalance that has been proven successful in multiple studies [21–23]. This penalty consists of the average of the false negative rate [24] and the false discovery rate [25] of each flight phase, multiplied by an influence factor α.

$$penalty = \alpha \cdot \frac{\sum_{c \in classes} \left(\frac{FP_c}{TP_c + FP_c} + \frac{FN_c}{TP_c + FN_c} \right)}{2 \cdot ||classes||} \tag{1}$$

FP = false positive, TP = true positive, FN = false negative

A hyperparameter gridseach is used to assess the optimal penalty influence value.

3.4. Evaluation Metrics

The evaluation of the different models relies on the concept of precision and recall [26]:

- The **precision** of each class is the number of correctly identified data points belonging to that class ($correct_{class}$) divided by the total number of data points identified as belonging to that class ($identified_{class}$).

$$precision_{class} = \frac{TP_{class}}{TP_{class} + FP_{class}} = \frac{correct_{class}}{identified_{class}} \tag{2}$$

- The **recall** of each class is the number of correctly identified data points belonging to that class ($correct_{class}$) divided by the total number of data points belonging to that class ($truth_{class}$)

$$recall_{class} = \frac{TP_{class}}{TP_{class} + FN_{class}} = \frac{correct_{class}}{truth_{class}} \tag{3}$$

TP = true positive, FP = false positive, FN = false negative

The average of precision and recall will be further referred to as weighted accuracy, while the overall accuracy refers to the ratio of data points correctly identified.
In the next section, the models obtained from training with and without penalty function, and with and with-

		Uniform w/o penalty ($\alpha = 0$)	Uniform w/ penalty ($\alpha = 3$)	K-means w/o penalty ($\alpha = 0$)	K-means w/ penalty ($\alpha = 3$)
Accuracy (%)	Overall	**97.26**	**97.04**	**97.15**	**97.09**
	Weighted	**91.20**	**91.09**	**90.30**	**90.64**
Precision (%)	TXI	99.02	99.08	98.48	98.53
	TOF	90.81	89.38	88.12	88.23
	ICL	76.25	75.02	81.20	81.31
	CLI	97.98	96.74	95.94	95.63
	CRZ	96.59	97.09	98.01	98.19
	DST	99.15	99.29	99.15	99.24
	APR	79.67	76.97	78.31	76.97
	LDG	88.10	87.55	89.46	89.77
	Average	**90.95**	**90.14**	**91.09**	**90.98**
Recall (%)	TXI	99.00	98.99	99.28	99.26
	TOF	83.56	84.28	78.26	79.91
	ICL	82.86	78.73	70.73	72.28
	CLI	96.89	96.75	97.30	97.42
	CRZ	97.25	96.98	97.19	96.84
	DST	98.66	97.69	98.25	97.99
	APR	85.66	92.52	89.57	90.56
	LDG	87.67	90.27	85.57	88.08
	Average	**91.44**	**92.03**	**89.52**	**90.29**

TAB 4. Average results of variations of the model for average of 3 instances of the same model trained independently.

out K-means segmentation are compared using these metrics.

4. RESULTS

In this section, we discuss the results of different variations of the model: using and not using K-means segmentation and using and not using the penalty function.

An overview of the results of the variations of the model can be found in TAB 4. For each variation, three instances were independently trained, the results shown are averaged over these.

K-means segmentation is introduced into the architecture to capture the transitions between flight phases more closely with the borders of the segments. In fact, FIG 1 shows a higher density of edge borders, where there are transitions from one phase to the next. If every segment would be correctly classified, these smaller segments around the transitions would reduce the error introduced by two flight phases overlapping within one segment. K-means segmentation reduces the error due to segmentation by more than 20% both on the training and testing data alike. This benefit, however, is not reflected in the combination with the classification, as can be seen in TAB 4. The improvement by the usage of K-means segmentation lies in the increased precision at the expense of a decreased recall, which results in an overall lesser weighted accuracy.

The cause of these results could lie in the fact that when using uniform segmentation, the feature that represents the length of the segment is removed as the network would not be able to use an input that shows the same value for every element in the sequence. The sequence length, on the other hand, is needed when the segments vary in size with each element in the sequence. This iss empirically confirmed by training both variants with and without the segment length feature and comparing the results on the validation dataset. The length of a segment might be a variable that decreases the number of false positives and increases the false negatives for the shorter phases.

The loss penalty function, on the other hand, has a positive impact on the shorter phases that present a relatively lower weighted accuracy (TOF, ICL, APR, and LDG). It slightly improves their weighted accuracy, mostly by increasing the recall, leading to an overall higher recall but lower precision. This shows that when using the loss penalty function, the model assigns more segments to the shorter phases that have an overall lower weighted accuracy.

Overall, however, the best performing model is the one trained without loss penalty that uses uniform segmentation. FIG 2 shows an ADS-B trajectory of a research aircraft with flight phases identified with this model. For this flight, as is often the case for ADS-B, the ground data is incomplete and as such some phases are not present. The main identification error lies in the take-off being prolonged, however the overall performance on this flight is satisfactory with an 98.39% overall accuracy.

5. CONCLUSION

In this work we presented a novel approach to flight phase identification using supervised learning. In contrast to other models, we provide the classification module with information regarding the past trajectory of the flight, by using a LSTM on a flight represented as a sequence.

The application of our approach to simulated data shows promising results for the inclusion of flight phases in trajectory-based identification, that have not been considered so far and that adhere to the ICAO standard. The system is developed for short-haul flights, defined by the IATA as less than 6 hours flight time[3], of commercial powered fixed-wing aircrafts. This is due to the fact that some flight phases might require slightly different rule sets for other types of aircraft or long-haul flights. To allow for comparison, the performance of the best overall performing model is shown on ADS-B data of a research aircraft flight, for which the flight phases are known. Upon comparison of the simulated data (e.g. FIG 1) to real trajectories (e.g. FIG 2) some differences become apparent. The recorded simulation flights have a short length bias, with an average flight duration of 29 minutes. The trajectory values themselves also present some differences in their characteristics, such as smoothness and range. When considering longer flights K-means segmentation might have a bigger impact on the results of the model, as the segments increase in size.

While this architecture is developed for specific conditions and data, we hypothesise that the results will

[3] https://www.iata.org/contentassets/821b593dd8cd4f4aa33b63ab9e35368b/iata-cabin-waste-handbook---final-resized.pdf accessed on 10/08/2021

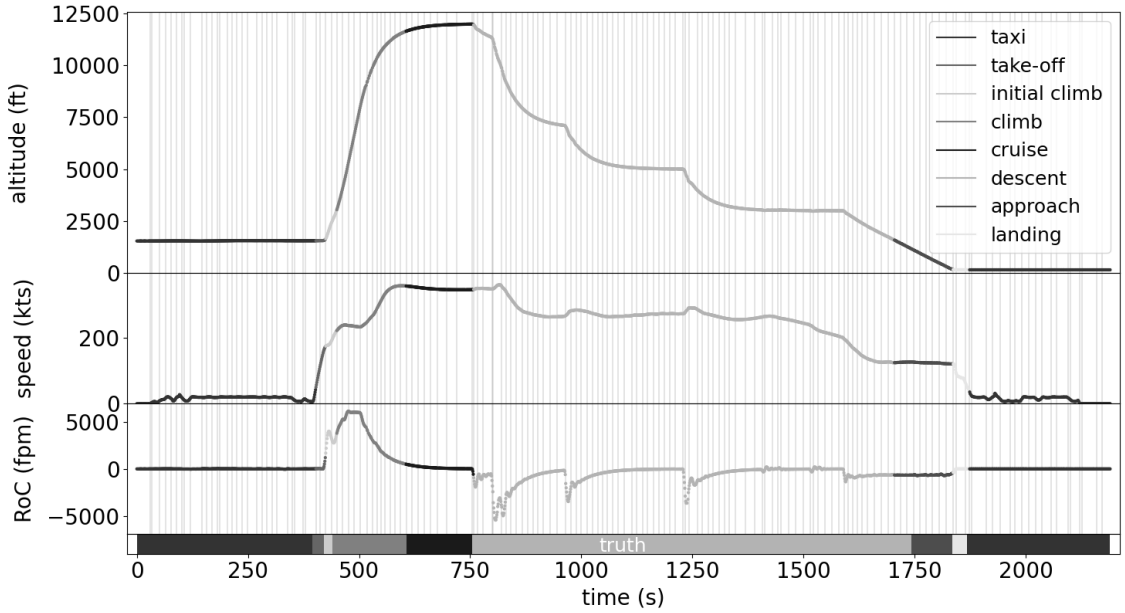

FIG 1. Identification on simulated flight with model using K-means segmentation and penalty for training with penalty influence $\alpha = 3$. Overall accuracy 97.04%. The curves are colored with the identified flight phase, at the bottom the colors of the correct label are shown. The faded background lines indicate the edges between segments.

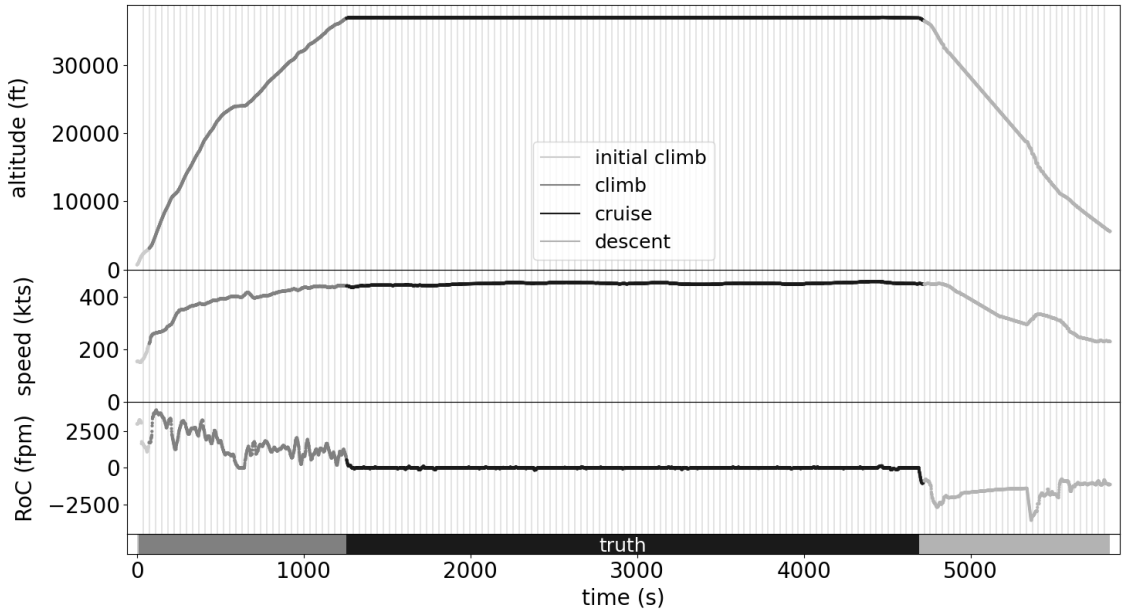

FIG 2. Identification on ADS-B trajectory data of an Airbus A320 research aircraft with model using uniform segmentation and no penalty. Overall accuracy 98.39%. The curves are colored with the identified flight phase, at the bottom the colors of the correct label are shown. The faded background lines indicate the edges between segments.

be similar with the same architecture transferred to similar applications. Such application could include real data for training to make the system more suited to ADS-B data. Further applications could also include different scopes, such as flight phases drawn from different definitions or maneuver detection. To apply the architecture to different data it needs to be retrained and it might be necessary to perform new hyper-parameter tuning. When adapting the architecture to perform different tasks it will additionally also be necessary to provide a new training dataset with the correct labels. In the case of maneuver detection, one might consider training different models that can detect a single maneuver through binary classification and combine them.

Future work will include the exploration of the architecture on different data and tasks to verify remaining hypothesis and widen the scope.

Contact address:

emy.arts@dlr.de

References

[1] R. Meissner, P. Reichel, P. Schlick, C. Keller, and K. Wicke. Engine load prediction during take-off for the v2500 engine. In *5th European Conference of the Prognostics and Health Management Society 2020*, volume 5, Juli 2020. DOI: https://doi.org/10.36001/phme.2020.v5i1.1214.

[2] H. Meyer, J. Zimdahl, A. Kamtsiuris, R. Meissner, F. Raddatz, S. Haufe, and M. Bäßler. Development of a digital twin for aviation research. In *Deutscher Luft- und Raumfahrt Kongress 2020*, Bonn, Germany, September 2020. Deutsche Gesellschaft für Luft- und Raumfahrt - Lilienthal-Oberth e.V. DOI: https://doi.org/10.25967/530329.

[3] S. Kovarik, L. Doherty, K. Korah, B. Mulligan, G. Rasool, Y. Mehta, P. Bhavsar, and M. Paglione. Comparative analysis of machine learning and statistical methods for aircraft phase of flight prediction. 2020.

[4] Z. Shi, M. Xu, Q. Pan, B. Yan, and H. Zhang. Lstm-based flight trajectory prediction. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2018.

[5] T. Baklacioglu. Modeling the fuel flow-rate of transport aircraft during flight phases using genetic algorithm-optimized neural networks. volume 49, pages 52–62. Elsevier, 2016.

[6] Y.S. Chati and H. Balakrishnan. Analysis of aircraft fuel burn and emissions in the landing and take off cycle using operational data. In *International Conference on Research in Air Transportation*, 2014.

[7] H. Aydın, Ö. Turan, T. Hikmet Karakoç, and A. Midilli. Exergo-sustainability indicators of a turboprop aircraft for the phases of a flight. volume 58, pages 550–560, 2013. DOI: https://doi.org/10.1016/j.energy.2013.04.076.

[8] C. Uri. Notes on the flight phase taxonomy. 2018.

[9] International Civil Aviation Organization (ICAO). ECCAIRS Aviation 1.3.0.12 Data Definition Standard English. Standard, Montreal, Canada, April 2013.

[10] J. Sun, J. Ellerbroek, and J. Hoekstra. Large-scale flight phase identification from ads-b data using machine learning methods. In *7th International Conference on Research in Air Transportation*, 2016.

[11] D. Michie, D.J. Spiegelhalter, C.C. Taylor, et al. Machine learning. volume 13, pages 1–298. Technometrics, 1994.

[12] D.E. Rumelhart, G.E. Hinton, and R.J. Williams. Learning representations by back-propagating errors. volume 323, pages 533–536. Nature Publishing Group, 1986.

[13] S. Hochreiter and J. Schmidhuber. Long short-term memory. volume 9, pages 1735–1780, 1997. DOI: 10.1162/neco.1997.9.8.1735.

[14] D. Liu, N. Xiao, Y. Zhang, and X. Peng. Unsupervised flight phase recognition with flight data clustering based on gmm. In *2020 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, pages 1–6, 2020. DOI: 10.1109/I2MTC43012.2020.9128596.

[15] M. Paglione and R. Oaks. *Determination of Horizontal and Vertical Phase of Flight in Recorded Air Traffic Data*. DOI: 10.2514/6.2006-6772.

[16] M.J. Carrico. Operational flight phase determination and indication system. Google Patents, 2012. US Patent 8,217,807.

[17] L.A. Zadeh. Fuzzy sets. volume 8, pages 338–353, 1965. DOI: https://doi.org/10.1016/S0019-9958(65)90241-X.

[18] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, pages 281–297, Berkeley, Calif., 1967. University of California Press.

[19] M. Crossley. A guide to coordinate systems in great britain. 1999.

[20] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[21] F. Caliva, C. Iriondo, A. Morales Martinez, S. Majumdar, and V. Pedoia. Distance map loss penalty term for semantic segmentation. 2019.

[22] A. Buccini, O. De la Cruz Cabrera, M. Donatelli, A. Martinelli, and L. Reichel. Large-scale regression with non-convex loss and penalty. volume 157, pages 590–601, 2020. DOI: https://doi.org/10.1016/j.apnum.2020.07.006.

[23] Y. Zhu, X. Shen, and W. Pan. Network-based support vector machine for classification of microarray samples. volume 10, pages 1–11. BioMed Central, 2009.

[24] S.Z. Li and A. Jain. *False Negative Rate*. Springer US, Boston, MA, 2009. ISBN: 978-0-387-73003-5.

[25] Y. Benjamini and Y. Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. volume 57, pages 289–300. Wiley Online Library, 1995.

[26] D.L. Olson and D. Delen. *Advanced data mining techniques*. Springer Science & Business Media, 2008.