TU Delft

Delft University of Technology

Large-Scale Flight Phase Identification from ADS-B Data Using Machine Learning Methods

Sun, Junzi; Ellerbroek, Joost; Hoekstra, Jacco

Publication date
2016

Document Version
Accepted author manuscript

Published in
7th International Conference on Research in Air Transportation

Citation (APA)
Sun, J., Ellerbroek, J., & Hoekstra, J. (2016). Large-Scale Flight Phase Identification from ADS-B Data Using Machine Learning Methods. In D. Lovell, & H. Fricke (Eds.), *7th International Conference on Research in Air Transportation: Philadelphia, USA*

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

This work is downloaded from Delft University of Technology.
For technical reasons the number of authors shown on this cover page is limited to a maximum of 10.

Large-Scale Flight Phase Identification from ADS-B Data Using Machine Learning Methods

Junzi Sun, Joost Ellerbroek, Jacco Hoekstra
Control and Simulation, Faculty of Aerospace Engineering
Delft University of Technology
Delft, The Netherlands

***Abstract*—With the increasing availability of ADS-B transponders on commercial aircraft, as well as the rapidly growing deployment of ground stations that provide public access to their data, accessing open aircraft flight data is becoming easier for researchers. Given the large number of operational aircraft, significant amounts of flight data can be decoded from ADS-B messages daily. These large amounts of traffic data can be of benefit in a broad range of ATM investigations that rely on operational data and statistics. This paper approaches the challenge of identifying and categorizing these large amounts of data, by proposing various machine learning and fuzzy logic methods. The objective of this paper is to derive a set of methods and reusable open source libraries for handling the large quantity of aircraft flight data.**

***Keywords*—machine learning, ATM data, big data, fuzzy logic, BlueSky.**

I. Introduction

Automatic Dependent Surveillance - Broadcast (ADS-B) [1][2] is widely implemented in modern commercial aircraft. It uses satellite navigation technology to acquire the position information of the aircraft and broadcasts aircraft tracking information using the 1090 MHz Mode-S transponder. Information is broadcast unencrypted and can be received and decoded by anyone with simple ground station set-ups. Examples of common parameters transmitted through ADS-B are aircraft position, velocity, and identification. Each message can be identified by a 24 bit ICAO address that indicates the source aircraft.

The goal of this paper is to investigate a set of machine learning methods that can be applied to such large amounts of aircraft data, filter noisy information, and extract the relevant properties of a flight. This study is part of a larger project that aims to build open aircraft performance models by applying identification techniques on ADS-B data to estimate the required performance coefficients. The resulting models will be integrated in the open-source ATM simulator BlueSky [3], [4].

One of the principal issues when handling these large amounts of data is the fact that searching, aggregating, and processing data becomes increasingly computationally expensive as the data volume grows. Agile design of the database, therefore, becomes a necessity to facilitate these large amounts of data. Together with machine learning algorithms, it is possible to make calculations on a large scale. When analyzing these data, there are also a number of uncertainties related to aircraft flight information that need to be taken into account. These uncertainties can either be induced by on-board equipment variances or by communication interruptions, and they may lead to errors in the final output of aircraft position and speed data. Therefore, filtering and smoothing algorithms are also proposed in this paper. They are designed to reduce the impact of these uncertainties on the calculations.

As a first step, large volumes of data have to be extracted into individual flights. These can be full or partial flight paths, based on the completeness of the recorded samples. In the current study, unsupervised clustering algorithms are proposed to solve this problem. The DBSCAN and BIRCH methods have been selected to handle large databases with an unknown number of clusters. Due to the diversity of aircraft types, airline procedures, and air traffic control procedures, aircraft tend to have a large range of possible altitudes and speeds in the different flight phases. In order to be able to estimate each phase correctly, fuzzy logic is employed to explore the data of continuous flights.

The remainder of this paper is structured as follows. In section two, the concept of ATM big data is discussed. Statistics from ADS-B data are shown along with the solutions for storage and analysis. In sections three to five, we focus on the fundamentals and implementations of machine learning and fuzzy logic for the entire system. Experiments and results are also shown in each separate section. Finally, section six concludes the research of this paper and points out the future related research of the authors.

II. ATM big data

The methods proposed in this paper will be applied to two types of data sources. The first consists of our own ADS-B ground station configuration, which provides a stream of raw ADS-B messages with a coverage of about 400 KM. On average, this receiver provides 10 million ADS-B messages from 3,000 aircraft each day. These raw messages can be decoded into two million entries of position data and five million entries

of velocity data. These two features are aggregated as a post-process of the raw data. On a larger scale, online services such as the flight tracking network FlightRadar24 can be accessed to collect data from thousands of ground stations (approximately 5,000 through analysis of FlightRadar24 data stream), which has the potential to exceed billions of raw messages per day. Although a great portion of these data consist of duplicates, the unique entries of data can exceed hundreds of millions each day. The challenge of making use of those data falls into the domain of big data. This paper proposes tailored machine learning algorithms in an effort to handle such large quantities of ATM big data.

Aircraft flight data are not distributed normally around the globe, not even around a single ground station. ADS-B signal reception requires "line-of-sight." The signal of the transponder is attenuated with increasing distance from the receiver. Figure 1 shows a scatter plot of all positions within 24 hours, through a single antenna situated in Delft, The Netherlands. It can be seen that the message density drops with increasing distance from the ground station, which is caused by loss of signal from the transponder. Also notice in the southwest and northwest of the ground station, two rays of uncovered area are presented. This is due to two tall structures located close by, which block the passing signals from these specific directions.

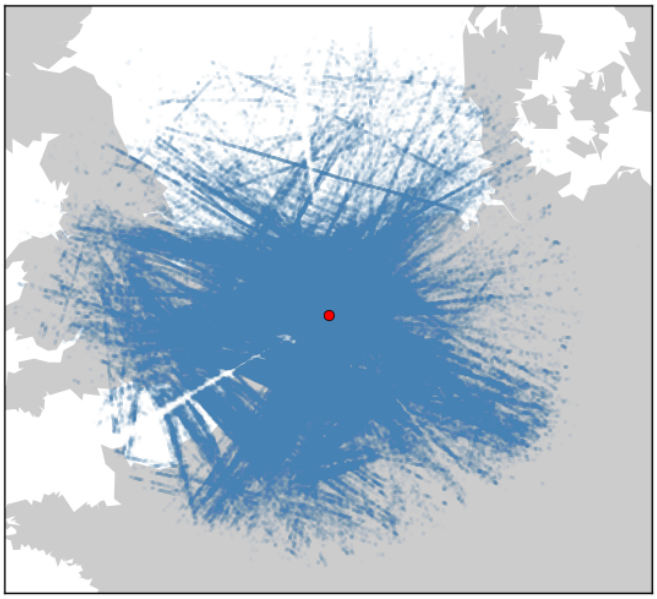

Fig. 1. ADS-B Positions (24h)

Contributors to the FlightRadar24 network are allowed to make use of, as well as process and redistribute [5], a much larger quantity of flight data. These data are gathered from ADS-B receivers around the world. Global ADS-B data can be processed and analyzed similar to local data, only on a much larger scale. Figure 2 shows a global data color map of ADS-B reports over a 24 hour period, where the densities are normalized over a total of 63 million position reports. The graph illustrates that the majority of the air traffic as received by the network is concentrated in Europe, North America, and South Asia.

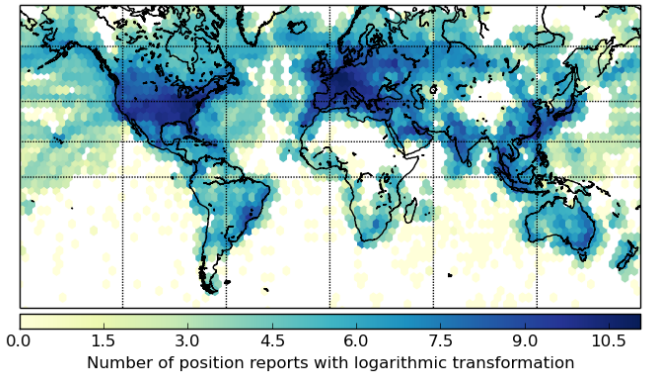

Fig. 2. Global position reports density (24h)

The large amount of data requires the use of a dedicated storage system. Several technologies are available, such as HDF5, SQL, or NoSQL databases [6], [7]. In this paper, a system will be used that best suits the data fields in Table I.

TABLE I
FEATURES OF ADS-B FLIGHT DATA

Field	Type	Unit
ICAO address	string	-
Aircraft model	string	-
Time stamp	integer	s
Latitude	float	deg
Longitude	float	deg
Altitude	float	ft
Heading	float	deg
Speed	float	knt

First, the raw stream of data is converted to JSON format structures, aligned with the schema defined in Table I. Then it is processed by the data storage engine. For the purpose of this research, as well as for better accessibility, document-oriented NoSQL databases are best suited for handling those ATM big data. This type of databases have comprehensive data aggregation methods and MapReduce operations, which makes the processing of data fast and comprehensible. They use common information exchange formats such as JSON to store the raw information and can be scaled up with increasing data storage needs. Another challenge faced by ATM big data is that information can be incomplete, frequently due to lack of inputs. One of the causes for this is the fact that position and velocity are not updated simultaneously. The missing

information may lead to a partial data stream, which does not contain all of the fields defined in Table I. A database engine that is able to handle such unexpected schema-less data frequently is therefore required. For this study, MongoDB was selected. It is a well developed open-source architecture that provides all of the above stated advantages and is also frequently used by researchers and industries from different domains [8].

III. Machine Learning and Data Mining

In order to extract continuous flights and further divide them into segments correlated with flight phases, several parameters (or features) need to be considered. The most significant ones are ICAO address, time stamp, latitude, and speed. Deterministic algorithms can be applied to sort data in different dimensions based on these features. These do, however, pose limitations in terms of efficiency, robustness, and scalability. This section describes a set of machine learning clustering methods that can be used to mitigate these limitations and to efficiently handle large sets of multi-dimensional noisy data.

A. Pre-process

Before data is forwarded to these statistical clustering algorithms, a few pre-processing steps are required. First, any non-numerical data needs to be converted into numerical values. In addition, different features need to be scaled to a reasonable range and missing values need to be computed to complete the dataset. These steps are respectively called data encoding, scaling, and imputation.

Most machine learning algorithms require inputs to be numbers, for example, while calculating Euclidean distance between data points. Data encoding is a process designed to translate text features into their numerical representations, such as ICAO addresses and aircraft types. In this paper, an integer encoder is used for the text features.

While looking at other numerical features, the range of data that were used in this paper varies significantly. Table II shows the reference ranges of each of the features (24 hour data).

TABLE II
Reference ranges

Feature	Data range	Unit
ICAO	[0, ~5000]	-
Time	[0, ~100000]	s
Latitude	[-180, 180]	deg
Longitude	[-90, 90]	deg
Altitude	[0, 40000]	ft
Heading	[0, 360]	deg
Speed	[0, 500]	knt

Large differences in values can lead to a large variation in the relative weights of features while calculating Euclidean distances [9]. A simple method to mitigate this is to scale features of $X = \{x_0, x_1, \cdots, x_n\}$ into a common range $[0, s_{max}]$, where all values can be converted to $X' = \{x'_0, x'_1, \cdots, x'_n\}$ as:

$$x'_i = \frac{x_i - \min(X)}{\max(X) - \min(X)} \times s_{max}$$

Some machining learning methods also require the data be standardized. Each feature then should be scaled based on the mean and standard deviation as follows:

$$x'_i = \frac{x_i - \bar{x}}{\delta_x}$$

where $\bar{x}$ and δ_x are the mean and standard deviation of the data respectively.

B. Dimensionality analysis

In machine learning processes, the dimensionality of the input features also plays a signification role. When dealing with data with multiple (often hundreds of) dimensions, a phenomenon called the Curse of Dimensionality occurs [10]. In higher dimensional data, objects appear to be sparse. Even large differences in one feature bring little changes in overall Euclidean distances, thus making identification and classification less efficient.

From a statistical point view, the sparse data samples in high dimensional data are close to the edge of the sample [11]. Assume N data points, distributed uniformly in an n-dimensional hypersphere centered at origin with a radius of 1. The expected median distance from the origin to the closest point is:

$$E[d_{\min}] = \left(1 - 0.5^{1/N}\right)^{1/n}$$

With n approaching infinity, the expected closest distance $d_{\min}$ becomes 1 even with large data sample number N, where it is almost the radius length of the hypersphere. This illustrates that all the data are distributed at the edge of a hypersphere.

Nevertheless, in this paper, the effect of dimensionality can be neglected due to the relatively small number of features represented in the data.

C. Clustering

Clustering or cluster analysis is an unsupervised learning process that groups data into subsets (clusters) based on the difference of the features. Several well-known algorithms (K-Means, DBSCAN, BIRCH, Mean-Shift, etc) are available in the literature [12], each with their own advantages for solving particular feature sizes and geometries.

The simplest clustering concept is the centroid-based method. Another popular method is called K-Means [13], which divides data samples into segments based on the Euclidean distance of each sample to the centroid of a cluster.

Given a dataset $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}$, with each sample a d-dimensional vector, the approach of the K-Means algorithm is to split all data into $k(k < n)$ segments $\{S_1, S_2, \cdots, S_k\}$. A clustering solution can be found using a two step process of centroid assignment, which is updated until the sum of all distances within each cluster has been minimized:

$$\arg\min_{\mathbf{S}} \sum_{i=1}^{k} \sum_{\mathbf{x} \in S_i} \|\mathbf{x} - \mathbf{c}_i\|^2$$

where, $\{\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_k\}$, are the centroids of all clusters.

K-Means is a direct algorithm and fairly computationally efficient. The disadvantage of this method is the pre-defined k number of clusters. ATM data very often has an undefined number of segments due to the different flight frequencies and operations, which requires the clustering method to be able to adapt the number of clusters depending on the data itself; at the same time, it should be able to handle a large number of clusters. Two algorithms have been selected based on this requirement, DBSCAN and BIRCH.

DBSCAN (density-based spatial clustering of applications with noise) is a density-based clustering method which separates data into areas of high and low density. DBSCAN uses two fundamental parameters: Eps and $MinPts$. Here, Eps is the maximum distance between two data samples for them to still be in the same neighborhood. $MinPts$ is the number of data samples in the neighborhood of a core point. $N_{Eps}(p) = \{q \in D | dist(p, q) \leq Eps\}$ is defined as the Eps-neighborhood of a point p. Clusters are formed when the following conditions are satisfied: [14]

$$p \in N_{Eps}(q)$$

$$|N_{Eps}(q)| \geq MinPts$$

The additional advantage of DBSCAN compared to a centroid-based method is the ability to generate clusters with a required density. It eliminates noise data that is at a lower density than the clusters. This aspect offers a considerable advantage in processing ATM data, insomuch as datasets with low data quality need to be excluded.

The second selected clustering algorithm is BIRCH (balanced iterative reducing and clustering using hierarchies) [15]. This method incrementally constructs a Characteristic Feature (CF) tree from the dataset with two user defined constraint numbers: the threshold (T) and the branching factor (B). An arbitrary clustering algorithm is used to cluster the leaf nodes of the CF tree. It can be considered as multi-level clustering, where a scalable lower level reduces the complexity before the higher-level clustering processing.

Given a multi-dimensional dataset with N data points, CF is defined as $CF = (N, LS, SS)$, where LS is the linear sum $\sum_i^N \mathbf{x}_i$ and SS is the squared sum $\sum_i^n \mathbf{x}_i^2$. When two CF trees (CF_1 and CF_2) are two disjointed clusters, the merging of the two will produce a new CF_M:

$$CF_M = CF_1 + CF_2 = (N_1 + N_2, LS_1 + LS_2, SS_1 + SS_2)$$

Within the TF tree, leaf and non-leaf nodes are constrained by the T and B values. A non-leaf node has at most B number of CF entries. The number of leaf node CF entries satisfies the threshold T. The entire CF tree is built dynamically as new data objects are inserted into the CF. Each leaf node in the final CF tree is a sub-cluster. After that, the high level clustering will generate the final clusters from all leaf nodes based on their CF values, using agglomerating hierarchical clustering.

The BIRCH method scans the entire dataset only once, which results in improved performance on large datasets. It also handles outliers better, compared to the previously discussed K-Means method.

IV. Flight extraction using clustering algorithms

To design and apply the clustering methods, a 24-hour flight dataset is selected from the database. It contains around 12 million raw messages, from which 1.7 million entries of flight data are decoded.

To simplify the features, each entry of data consists of an aircraft location, velocity, identity, and time stamp. The challenge is to cluster these scattered flight data into small sets of continuous flight trajectories. The algorithms need to be able to deal with large unknown numbers of clusters and a reasonable quantity of outliers caused by the noisiness of ADS-B data.

For illustrative purposes, only a smaller sample set is plotted to show the results of clustering. The sample set includes 200 random aircraft with approximately 100,000 entries of data. Both BIRCH and DBSAN methods are applied to the sample with different configurations. The multi-dimensional data is represented by two features in the graph, aircraft ID and the time stamps of each data entry, displayed along the x and y axis respectively.

Figure 3 shows the results of BIRCH clustering. Data in the same clusters is linked and represented by the same color. From top to bottom, performance of the method changes, while the threshold value decreases from 100 to five. The smaller the threshold of the CF tree, the smaller a leaf can be. This will result in decreasing cluster size. In the top graph, it can be seen that clusters are formed to contain data from different aircraft, which is far from an optimal result. The middle graph shows that the data are nicely clustered as desired with only a few exceptions. The bottom configuration produces the finest clusters, as well as a higher number of clusters. However, data that should belong together in a single flight trajectory is split into different clusters. Tuning the threshold value is required to find the better balance.

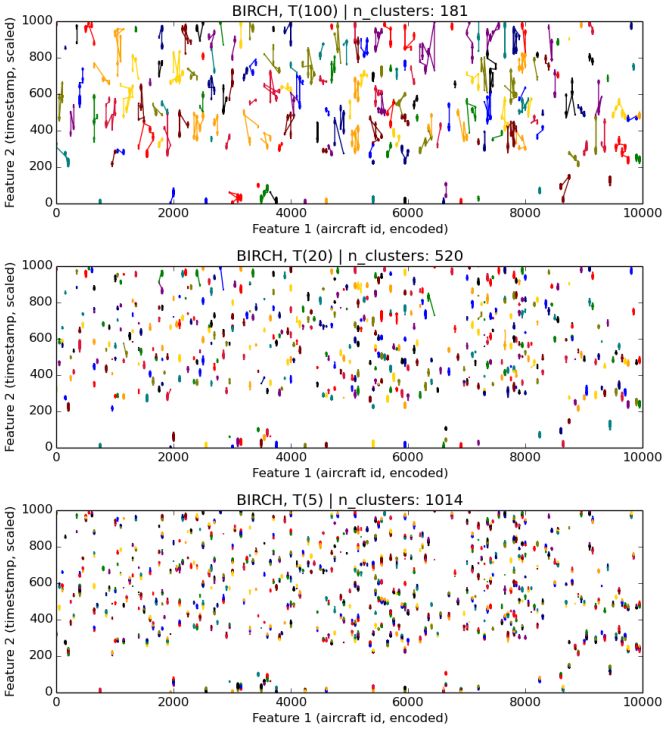

Fig. 3. Clustering with BIRCH method

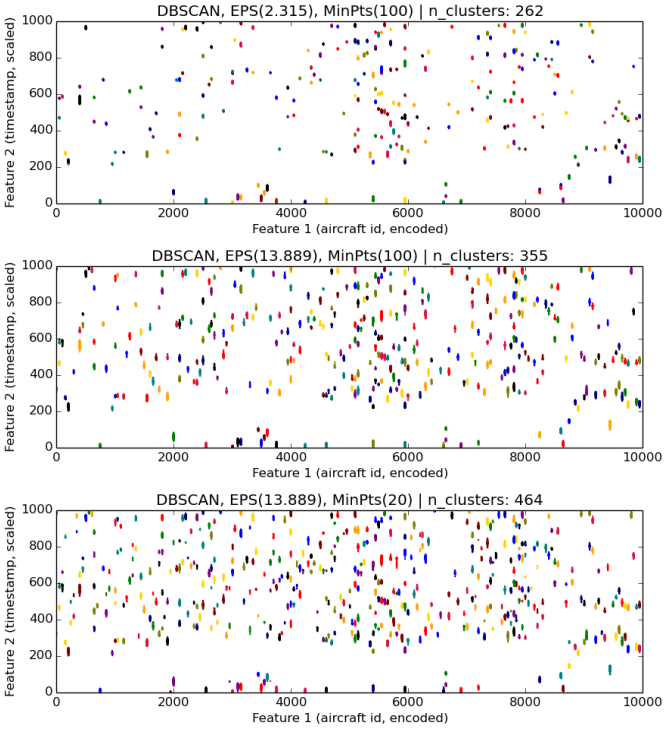

Fig. 4. Clustering with DBSCAN method

The clustering process with the DBSCAN method was applied to the same dataset, in order to evaluate the flight extraction performance. The result is illustrated in Figure 4. The changing parameters are EPS and $MinPts$, which represent the maximum distance of data and minimum number of samples in a single cluster. Increasing EPS leads to larger average cluster size, while increasing $MinPts$ eliminates clusters with a small number of samples. The clustering process can be optimized by tuning the combination of these two variables.

Compared to BIRCH, DBSCAN can exclude some clusters from the result by specifying the $MinPts$ value. This gives control over the final cluster quality for further processing.

Furthermore, both BIRCH and DBSCAN can be tuned to work well with the ATM big data set. Because of their temporal nature, input data can be separated into smaller batches, thus offering the possibility to run the machine learning process on regular workstations with limited memory resources.

V. Flight Phase Identification using Fuzzy Logic

The outcome of the clusters provides us a set of continuous flight data, representing either full or partial trajectories of certain flights. The ability to segment data further into flight phases is important to complete further research on building aircraft performance models.

Previous clustering methods may still be used to create sub-clusters based on the characteristics of time series data [16]. However, two problems arise when applying classic clustering methods.

1) Each entry in a data set is relatively close to its neighbors, based on Euclidean distance of time stamp, altitude, velocity, and position. The clustering method is not able to produce sub-clusters with a certain level of consistency.

2) Due to difference in aircraft types and the divergent flight procedures, flight behavior may vary, which could lead to, for example, aircraft climbing at different rates, flying at different cruise altitudes, and traveling at different speeds, even within the same phase.

These two problems can be solved with fuzzy logic being applied on the time series data. Fuzzy logic, also known as fuzzy sets theory [17], has been introduced to express real-world objects or concepts where no precise definition of criteria for membership exist. It uses membership functions to define the degree of truth for different features. Logic operators AND, OR, and NOT are defined as minimum, maximum, and complement operators. Different output states are activated by certain input operations. In this particular problem, three inputs are used (i.e. altitude, rate of climb, and ground speed)

to determine the flight phase. In Figure 5, the membership functions of the input and output are defined.

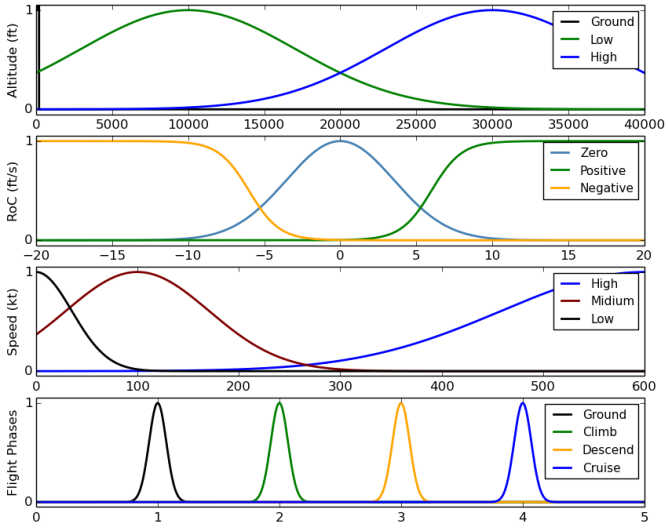

Fig. 5. Membership functions

The logic of the estimator can be described as follows:

$$H_{\text{Ground}} \wedge V_{\text{Low}} \Rightarrow FP_{\text{Ground}}$$
$$H_{\text{Low}} \wedge V_{\text{Medium}} \wedge RoC_{+} \Rightarrow FP_{\text{Climb}}$$
$$H_{\text{High}} \wedge V_{\text{High}} \wedge RoC_{0} \Rightarrow FP_{\text{Cruise}}$$
$$H_{\text{Low}} \wedge V_{\text{Medium}} \wedge RoC_{-} \Rightarrow FP_{\text{Descend}}$$

where H, RoC, V, and FP refer to altitude, rate-of-climbing, ground speed, and flight phase, respectively. The probabilities of all four phases are computed for any input and the most likely phase is considered as its state. However, in case of a extremely low outcome probability, an unknown state is marked. In reality, the data is likely to be corrupted in those cases.

One issue that can influence the performance of the segmentation is data noise, as the input data usually contains noise. Features such as speed and rate-of-climb demonstrate a large variation. For an estimator to be able to determine flight phase more accurately, data is usually filtered (e.g., using a SavitzkyGolay filter [18]) before being processed with fuzzy logic. To reduce the steps necessary for segment identification, the entire time series data are divided into multiple, one-minute time windows of one-minute before the segmentation process starts.

The entire segmentation process is presented in Figure 6. Continuous flight data is streamed as input, before it is smoothed and sliced into multiple time windows. All time windows are processed by the fuzzy logic module to identify the exact flight phase. The output consists of a series of labels stating the flight phase of each data entry.

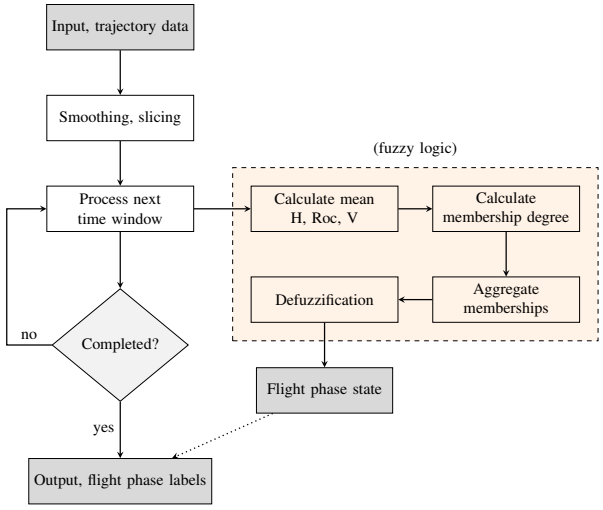

Fig. 6. Flight segmentation with fuzzy logic

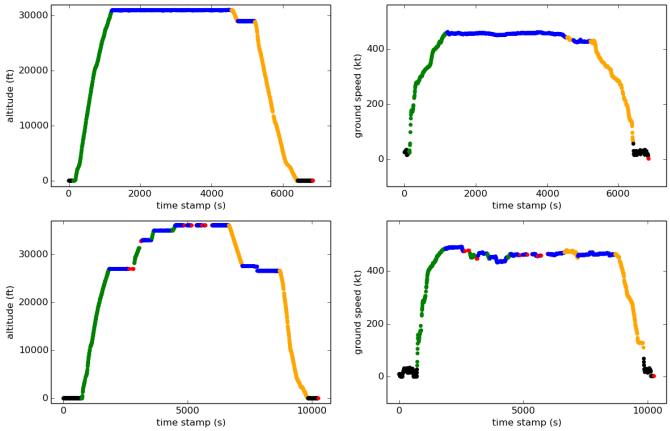

Fig. 7. Fuzzy logic segmentation - example visualization

To validate the method, the output labels are fed in a visualization module with distinct colors for different labels. Example results are shown in Figure 7. Within each figure, altitude and speed are plotted against each data entry time stamp. The colors black, green, blue, and orange are the labels for ground, climb, cruise, and descent accordingly. The red color represents an un-identifiable state due to the incorrectness of data in related time window. The segmentation method shows promising results with the fuzzy logic state estimator.

For those red data points, a simple heuristic method can be used to determine their phase state based on the state of the closest neighbor. Sets A and N represent data with and without labels respectively. Each stateless data point can be labeled as:

$$Ph_N(j) = Ph_A \left[\arg\min_{\mathbf{i}} \|t(j) - t(i)\| \right]$$

where, t is the time stamp and Ph is the flight phase label.

With this, a continuous flight trajectory data can be divided into designated flight phases as we required, thus achieving the goal of this paper. The output data are also stored in the database with original data to be prepared for upcoming research.

VI. Discussion

The two-step process of flight extraction and phase segmentation convert unstructured flight data into clusters of useful subsets of data and enables the further research based on large sets of ATM big data.

An operational system relies on a solid data storage infrastructure. In this paper, MongoDB has been selected as the backend storage system due to its portability and availability. However, more comprehensive data designs such as the Apache Hadoop [19] system can also be used to maintain a large amount of real time data with distributed servers.

One limitation of the segmentation process is that the system currently is not able to separate flight data into further detailed flight phases, such as taxing, take-off, landing, and initial climbing/descending. For these, studies need to be conducted using more deterministic approaches and possibly aggregating other data sources, which goes beyond the scope of this paper.

The output data are being used in different ATM research applications, such as aircraft performance modeling, air traffic analysis and simulation, airspace capacity studies, and continuous descent approach. Both tools and data created in this research have been made public with flexible open-source licenses [20].

VII. Conclusions

In this paper, a machine learning approach to handle ATM flight data is presented. Multiple levels of methods are designed to gather, extract, cluster, and segment large amounts of loosely scattered data into useful continuous flight segments. The system can operate with a large amount of ATM data, which contains an unknown number of flights, aircraft types, locations, and flight patterns. The core methods are unsupervised machine learning (clustering) and fuzzy logic, each solving a different level of the identification problem. The input data are usually noisy, which means that filters sometimes need to be applied beforehand.

The result of this processing system shows good promise in handling ATM flight data. It has been implemented and used daily for us to process data from ADS-B receivers. Due to the built-in arbitrary conditions, it produces robust date output.

In upcoming research, the segmented flight results will be used as bases of data to build medium to low fidelity open aircraft performance models for the open-source BlueSky ATM simulator.

References

[1] ICAO, "Guide on technical and operational considerations for the implementation of ADS-B in the SAM Region (Version 1.2)," , No. May, 2013, pp. 1–61.

[2] ICAO, *Technical Provisions for Mode S Services and Extended Squitter*, No. June, 2009.

[3] Hoekstra, J., "BlueSky Software," http://homepage.tudelft.nl/7p97s/BlueSky/home.html, November, 2015.

[4] Hoekstra, J. and Ellerbroek, J., "BlueSky ATC Simulator Project: an open Data and Open Source Approach," *Proceedings of the 7th International Conference on Research in Air Transportation*, 2016, submitted.

[5] "FlightRadar24, Terms and Conditions, Item 7," http://www.flightradar24.com/terms-and-conditions, November, 2015.

[6] Folk, M., Heber, G., Koziol, Q., Pourmal, E., and Robinson, D., "An overview of the HDF5 technology suite and its applications," *Proceedings of the EDBT/ICDT 2011 Workshop on Array Databases*, ACM, 2011, pp. 36–47.

[7] Stonebraker, M., Cetintemel, U., and Zdonik, S., "The 8 requirements of real-time stream processing," *ACM SIGMOD Record*, Vol. 34, No. 4, 2005, pp. 42–47.

[8] Hoberman, S., *Data Modeling for MongoDB: Building Well-Designed and Supportable MongoDB Databases*, Technics Publications, 2014.

[9] Milligan, G. and Cooper, M., "A study of standardization of variables in cluster analysis," *Journal of Classification*, Vol. 5, No. 2, 1988, pp. 181–204.

[10] Bellman, R. and Corporation, R., *Dynamic Programming*, Rand Corporation research study, Princeton University Press, 1957.

[11] Hastie, T., Tibshirani, R., and Friedman, J., *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition*, Springer Series in Statistics, Springer, 2009.

[12] Witten, I., Frank, E., and Hall, M., *Data Mining: Practical Machine Learning Tools and Techniques: Practical Machine Learning Tools and Techniques*, The Morgan Kaufmann Series in Data Management Systems, Elsevier Science, 2011.

[13] Kanungo, T., Mount, D., Netanyahu, N., Piatko, C., Silverman, R., and a.Y. Wu, "An efficient k-means clustering algorithm: analysis and implementation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, No. 7, 2002, pp. 881–892.

[14] Ester, M., Kriegel, H. P., Sander, J., and Xu, X., "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," *Second International Conference on Knowledge Discovery and Data Mining*, 1996, pp. 226–231.

[15] Zhang, T., Ramakrishnan, R., and Livny, M., "BIRCH: An Efficient Data Clustering Databases Method for Very Large," *ACM SIGMOD International Conference on Management of Data*, Vol. 1, 1996, pp. 103–114.

[16] Fu, T.-c., "A review on time series data mining," *Engineering Applications of Artificial Intelligence*, Vol. 24, No. 1, 2011, pp. 164–181.

[17] Zadeh, L., "Fuzzy sets," *Information and Control*, Vol. 8, No. 3, jun 1965, pp. 338–353.

[18] Savitzky, A. and Golay, M. J., "Smoothing and differentiation of data by simplified least squares procedures." *Analytical chemistry*, Vol. 36, No. 8, 1964, pp. 1627–1639.

[19] White, T., *Hadoop: The definitive guide*, " O'Reilly Media, Inc.", 2012.

[20] Sun, J., "Flight Data Processing Library, Code Repository on Github," https://github.com/junzis/flight-data-processor, Feburary, 2016.