Flight Extraction and Phase Identification for Large Automatic Dependent SurveillanceBroadcast Datasets

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

1 Introduction

Automatic Dependent Surveillance - Broadcast (ADS-B) [1] [2] is widely implemented in modern commercial aircraft and will become mandatory equipment in 2020. Flight state information such as position, velocity, and vertical rate are broadcast by tens of thousand aircraft around the world constantly using on-board ADS-B transponders. These data are identified by a 24-bit ICAO address, are unencrypted, and can be received and decoded with simple ground station set-ups. This large amount of open data brings a huge potential for ATM research.

Most studies that rely on aircraft flight data (historical or real-time), require knowledge on the flight phase of each aircraft at a given time. [3, 4, 5, 6, 7] However, when dealing with large datasets such as from ADS-B, which can contain many tens of thousands of flights, exceptions to deterministic definitions of flight phases are inevitable, due to large variances in climb rate, altitude, velocity, or a combination of these. In this case, instead of using deterministic logic to process and extract flight data based on flight conventions, robust and versatile identification algorithms are required. In this paper, a twofold method is proposed and tested: 1) A machine learning clustering step that can handle large amounts of scattered ADS-B data to extract continuous flights. 2) A flight phase identification step that can segment flight data of any type of aircraft and trajectory by different flight phases.

2 Flight Extraction from Large ADS-B Datasets

2.1 Data Fields

ADS-B information collected from ground stations is usually loosely stored as scattered data points representing states of all aircraft at different timestamps. Regardless of the choice of data store, the data schema usually consists of following elements listed in Table 1.

Table 1: Features of ADS-B flight data

Field	Type	Value range	Unit
ICAO address	string	-	-
Callsign	string	-	-
Time stamp	integer	-	s
Latitude	float	[-180, 180]	deg
Longitude	float	[-90, 90]	deg
Altitude	float	[0, 40000]	ft
Speed	float	[0, 500]	kts
Heading	float	[0, 360]	deg

 DOI: 10.2514/1.I010520

For the current study, a non-relational database, MongoDB, is used to store the ADS-B and flight data. It is a well-developed open-source data architecture frequently used for document-based big data processing. [8]

2.2 Pre-processing

In general, several pre-processing steps are required before applying machine learning. 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.

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 each feature $X = \{x_0, x_1, \cdots, x_n\}$ into a common range $[0, s_{max}]$, where all values can be converted to $X' = \{x'_0, x'_1, \cdots, x'_n\}$ as:

$$x'_i = \frac{x_i - \min(X)}{\max(X) - \min(X)} \times s_{max} \tag{1}$$

A numerical label encoder is used for text features such as ICAO addresses. However, converting these text features into numerical features implies a finite distance between any two different labels, which can affect clustering. In order for the algorithm to distinguish data from different aircraft, the scaling factor assigned to this feature, therefore, needs to be significantly larger compared to other features.

2.3 Clustering

When extracting continuous flights from a scattered ADS-B dataset, using the features in Table 1, two variables play a major role. These are the aircraft identification (ICAO address) and the timestamp. This is due to the fact that a single flight can only be carried out by a single aircraft and that each aircraft commonly carries out multiple flights, even during the same day with several stops in between.

Based on these characteristics, a straightforward approach to extracting flights would be to implement a procedure that filters all data belonging to each single aircraft, sort the data by time, and decompose them sequentially. However this would greatly increase the requirements on computational power, and would decrease the efficiency when dealing with very large datasets. The use of unsupervised machine learning, also known as clustering, on a large ADS-B dataset as a whole can have two significant benefits: 1) it can increase the efficiency when dealing with many aircraft simultaneously, 2) it is able to handle outliers caused by irregularities in flight data.

Clustering (or cluster analysis) groups data into subsets (clusters) based on the differences of the features among data points. Several well-known algorithms (K-Means, DBSCAN, BIRCH, Mean-Shift, etc) are available, each with their own advantages for solving particular feature sizes and geometries.

In this study, DBSCAN (density-based spatial clustering of applications with noise) proposed by Ester [10] was selected, because of its ability of handling unknown number of clusters and outliers efficiently. DBSCAN is a density-based clustering method, which separates data into areas of high and low density. DBSCAN uses two fundamental parameters: Eps and $MinPts$. Three types of data points are classified: core points, reachable points, and outliers. 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. As expressed in [10], clusters are are formed as follows:

1. If more than $MinPts$ points are within a distance of Eps to $\boldsymbol{p}$, then $\boldsymbol{p}$ is considered as a core point. These points are all defined as directly density-reachable from $\boldsymbol{p}$.

2. A point $\boldsymbol{q}$ is reachable from $\boldsymbol{p}$ if a directly density-reachable path $d_1, d_2, \cdots, d_n$ exist, where d_1 and d_n are $\boldsymbol{p}$ and $\boldsymbol{q}$

3. From all above points, a cluster is formed.

Although it is not explicitly expressed in the original DBSCAN paper [10], data points that are not density-reachable are considered as outliers. The ability to identify outliers offers a considerable advantage in processing ADS-B data, insomuch as it is preferable to systematically exclude trajectories with low data quality. This a key advantage over other types of clustering methods.

Fig. 1 gives an example of the results of the DBSCAN method on a small test dataset. From the first to last plot, increasing *Eps* leads to a 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. Performance benchmarking and parameter tuning is presented in section 4.1.

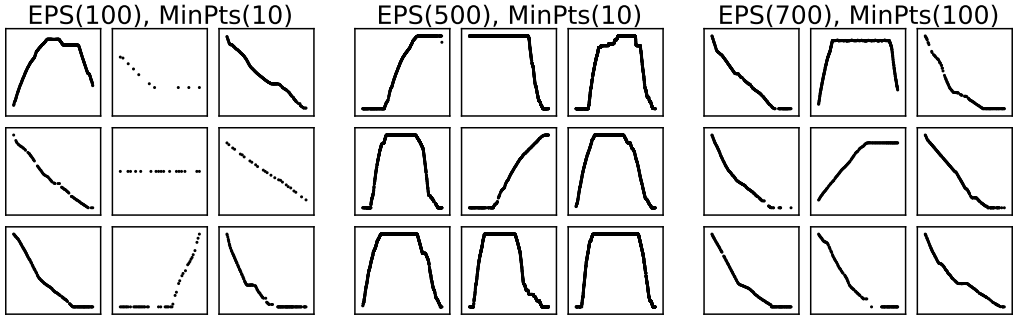

Figure 1: Clustering with DBSCAN

3 Flight Phase Identification

The outcome of the clustering process provides a set of continuous flights, representing either full or partial trajectories. In order to segment a flight into different phases, previous clustering methods may still be used to create sub-clusters based on the characteristics of time-series data [11]. However, two problems arise when applying clustering.

1) Each data point is relatively close to its neighbors based on the Euclidean distance between timestamps, altitudes, velocities, and positions. The classic clustering method cannot produce sub-clusters with a sufficient level of consistency.

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

These two problems can be solved by applying fuzzy logic on the time series data. Fuzzy logic, also known as fuzzy sets theory [12], has been introduced to express real-world objects or concepts where no precise definition of criteria 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.

Most of the membership functions are defined as Gaussian function (denoted as $\mathcal{G}$), where the mean μ and standard deviation σ reflect the reasonable value and range of uncertainty:

$$\mathcal{G}(x;\mu,\sigma) = \exp\left(\frac{-(x-\mu)^2}{2\sigma^2}\right) \tag{2}$$

Other membership function used are Z-shaped membership functions (denoted $\mathcal{Z}$) and S-shaped membership functions denoted $\mathcal{S}$, which are defined as follows:

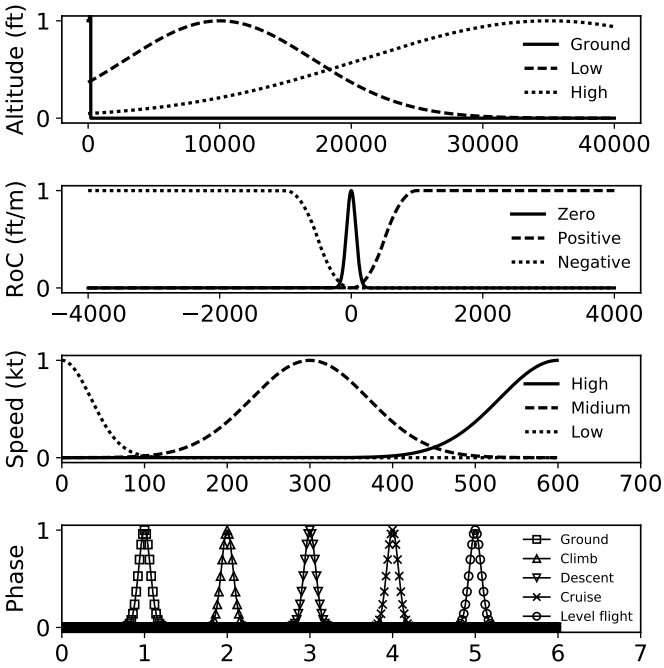

Figure 2: Membership functions

$$\mathcal{Z}(x;a,b) = \begin{cases} 1, & x \leq a \\ 1 - 2\left(\frac{x-a}{b-a}\right)^2, & a \leq x \leq \frac{a+b}{2} \\ 2\left(\frac{x-b}{b-a}\right)^2, & \frac{a+b}{2} \leq x \leq b \\ 0, & x \geq b \end{cases} \tag{3}$$

$$\mathcal{S}(x;a,b) = \begin{cases} 0, & x \leq a \\ 2\left(\frac{x-a}{b-a}\right)^2, & a \leq x \leq \frac{a+b}{2} \\ 1 - 2\left(\frac{x-b}{b-a}\right)^2, & \frac{a+b}{2} \leq x \leq b \\ 1, & x \geq b \end{cases} \tag{4}$$

Here, a and b (or b and a) are the high and low extremes of the sloped part of the function curve.

Examples of these three types of function are shown in Fig. 2, where all membership functions are illustrated as defined in Eq. 5. Each membership function is constructed with appropriate values of previously described μ and σ or a and b. H, V, RoC, and P represent altitude, speed, rate-of-climb, and flight phase respectively.

$$
\begin{aligned}
H_{gnd}(\eta) &= \mathcal{Z}(\eta, 0, 200) && \text{(5a)}\\
H_{lo}(\eta) &= \mathcal{G}(\eta, 10000, 10000) && \text{(5b)}\\
H_{hi}(\eta) &= \mathcal{G}(\eta, 35000, 20000) && \text{(5c)}\\
RoC_0(\tau) &= \mathcal{G}(\tau, 0, 100) && \text{(5d)}\\
RoC_+(\tau) &= \mathcal{S}(\tau, 10, 1000) && \text{(5e)}\\
RoC_-(\tau) &= \mathcal{Z}(\tau, -1000, -10) && \text{(5f)}\\
V_{lo}(v) &= \mathcal{G}(v, 0, 50) && \text{(5g)}\\
V_{mid}(v) &= \mathcal{G}(v, 300, 100) && \text{(5h)}\\
V_{hi}(v) &= \mathcal{G}(v, 600, 100) && \text{(5i)}\\
P_{gnd}(p) &= \mathcal{G}(p, 1, 0.2) && \text{(5j)}\\
P_{clb}(p) &= \mathcal{G}(p, 2, 0.2) && \text{(5k)}\\
P_{cru}(p) &= \mathcal{G}(p, 3, 0.2) && \text{(5l)}\\
P_{des}(p) &= \mathcal{G}(p, 4, 0.2) && \text{(5m)}\\
P_{lvl}(p) &= \mathcal{G}(p, 5, 0.2) && \text{(5n)}
\end{aligned}
$$

Logically, knowing altitude, speed, and vertical rate without deterministic values, the following relationships can be used to identify the correct flight phase:

$$
\begin{aligned}
&if \quad H_{gnd} \wedge V_{lo} \wedge RoC_0 \quad then \;\; Ground && \text{(6a)}\\
&if \quad H_{lo} \wedge V_{mid} \wedge RoC_+ \quad then \;\; Climb && \text{(6b)}\\
&if \quad H_{hi} \wedge V_{hi} \wedge RoC_0 \quad then \;\; Cruise && \text{(6c)}\\
&if \quad H_{lo} \wedge V_{mid} \wedge RoC_- \quad then \;\; Descent && \text{(6d)}\\
&if \quad H_{lo} \wedge V_{mid} \wedge RoC_0 \quad then \;\; Level\ flight && \text{(6e)}
\end{aligned}
$$

Fuzzy logic takes such relationships between inputs and output to identify the five different flight phases (ground, climb, cruise, descent, and level flight during climb and descent), for a given data point, denoted as (η_i, τ_i, v_i), and all possible discrete flight phase states $\boldsymbol{P}$ $(0 < P_i < 6)$ as shown in the last plot of Fig. 2. Each fuzzy value (numerical representation of phase) can be calculated as follows:

$$
\begin{aligned}
S_{gnd}(\boldsymbol{P}) &= min\left[min\left[H_{gnd}(\eta_i), V_{lo}(v_i), RoC_0(\tau_i)\right], P_{gnd}(\boldsymbol{P})\right] && \text{(7a)}\\
S_{clb}(\boldsymbol{P}) &= min\left[min\left[H_{lo}(\eta_i), V_{mid}(v_i), RoC_+(\tau_i)\right], P_{clb}(\boldsymbol{P})\right] && \text{(7b)}\\
S_{cru}(\boldsymbol{P}) &= min\left[min\left[H_{hi}(\eta_i), V_{hi}(v_i), RoC_0(\tau_i)\right], P_{cru}(\boldsymbol{P})\right] && \text{(7c)}\\
S_{des}(\boldsymbol{P}) &= min\left[min\left[H_{lo}(\eta_i), V_{mid}(v_i), RoC_-(\tau_i)\right], P_{des}(\boldsymbol{P})\right] && \text{(7d)}\\
S_{lvl}(\boldsymbol{P}) &= min\left[min\left[H_{lo}(\eta_i), V_{mid}(v_i), RoC_0(\tau_i)\right], P_{gnd}(\boldsymbol{P})\right] && \text{(7e)}\\
S(\boldsymbol{P}) &= max\left[S_{gnd}(\boldsymbol{P}), S_{clb}(\boldsymbol{P}), S_{cru}(\boldsymbol{P}), S_{des}(\boldsymbol{P}), S_{lvl}(\boldsymbol{P})\right] && \text{(7f)}
\end{aligned}
$$

Here $S(\boldsymbol{P})$ is the combined fuzzy value computed according to the membership logic. The last step is known as *defuzzification*, where the most likely flight phase state $\hat{P}$ can be found as follows:

$$
\hat{P} = round(\arg\max_{P} S(P)) \tag{8}
$$

Here, $\hat{P}$ represents the final output where the highest combined fuzzy value occurs. Finally the numerical flight phase representation can be converted to human readable flight phase text.

To visualize the outcome, a fairly complex flight trajectory is applied with fuzzy logic flight phase identification. As shown in Fig. 3, different flight phases are marked correctly.

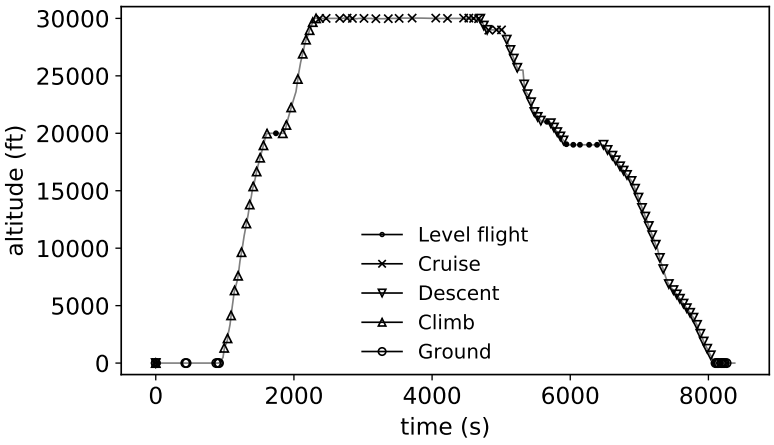

Figure 3: Fuzzy logic segmentation example

4 Validation

4.1 Benchmark of clustering methods

The quality of a clustering outcome not only depends on choosing the right machine learning method, but also on setting the proper parameters. To evaluate the algorithm thoroughly, a grid of parameters is used to benchmark the outcome of DBSCAN clustering algorithms. The testing dataset contains 518 flights extracted from FlightRadar24. With different parameter settings, it is possible to locate the best pair of *Eps* and *MinPts* for DBSCAN. Fig. 4 shows the benchmark results. The two axes of the figure represent the parameters to be tuned. The circular areas represent the number of clusters found by using different parameter pairs. From this figure, the parameters that yield the best performance can be easily identified. By comparing the outcomes with the ground truth on a such small dataset, correct settings for DBSCAN can be found before they are applied on a large-scale dataset.

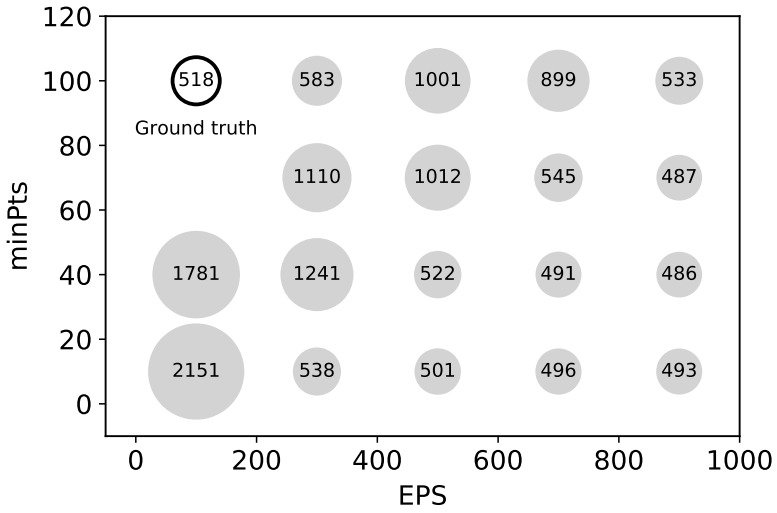

Figure 4: DBSCAN Benchmark

4.2 Examining flight phases

To validate the quality of the flight-phase identification process, two indicators are proposed:

1) The number of phase transitions (N_{Trans}): This indicator is calculated by comparing the phase of two adjacent data points, and summing the number of differences. Statistics of such a parameter on a large number of trajectories are used as a first evaluation.

2) The number of invalid transitions (E_{Trans}): A transition can only occur between certain phase states. The state diagram in Fig. 5 shows the possible transitions. Transitions that are not connected by arrows are considered as invalid transitions. They are counted for each flight.

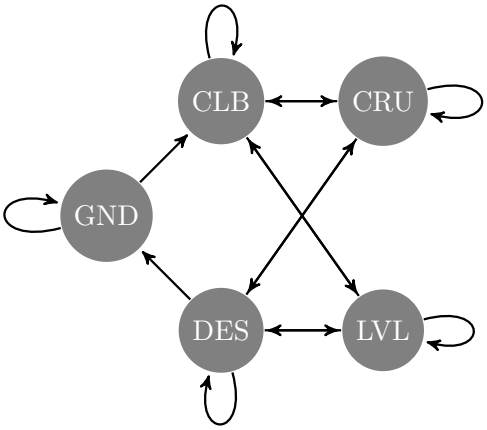

Figure 5: Flight phase state diagram

A second test dataset of 500 complete end-to-end flights is drawn from the database. N_{Trans} and E_{Trans} are calculated for all segmentation labels, shown in Fig. 6. The majority of flights contain around four to eight phase transitions. Most of the flights have zero invalid phase transitions, which holds for more than 95% of all flights. The total number of E_{Trans} is as low as 0.006%, which represents only 38 out of nearly 600,000 data points.

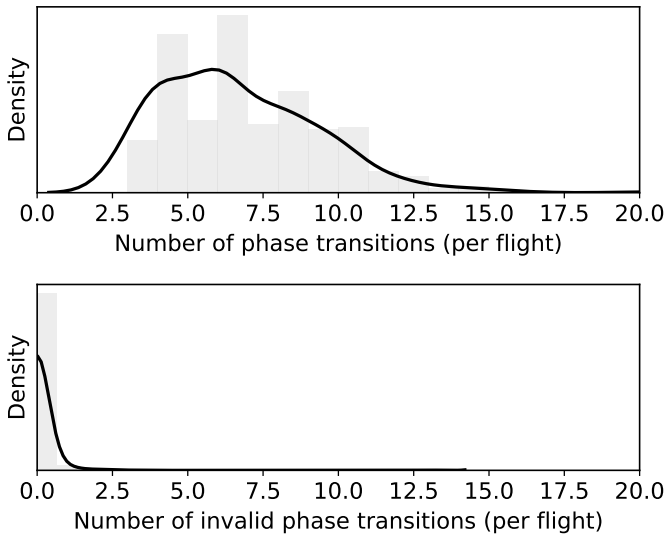

Figure 6: Evaluation of flight phase identification on 500 flights

5 Conclusions

In this paper, a twofold machine learning approach for mining scattered ADS-B data is presented. Methods at two different levels are proposed to extract flights, and segment them in individual

flight phases. The proposed method is robust to aircraft type and flight pattern variations. The core methods used in this approach are unsupervised machine learning (clustering using DBSCAN algorithm) and fuzzy logic identification. The approach was validated to ensure a reasonable and expected result.

It is worth to note that take-off and landing are both identified as ground phase. However with the derivative of velocity (acceleration), one can easily further identify the take-off and landing phase in the flight data. One issue that can influence the performance of the segmentation is noise in the measurement data. Features such as speed and rate-of-climb can demonstrate large fluctuations. One can, for example, use a SavitzkyGolay filter [13] or piece-wise low-order polynomial splines to smooth the data.

In order to reduce the computation time during the fuzzy logic identification along all data points in each flight, a shifting time window can limit the number of iterations significantly, as well as weaken the influences from noisy measurements.

From the results it can be concluded that the twofold machine learning approach proposed in this paper has the potential to enable researchers to handle large amounts of scattered flight data efficiently, and conveniently conduct various ATM studies based on open ADS-B data.

References

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

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

[3] S. Shresta, D. Neskovic, and S. S. Williams, "Analysis of continuous descent benefits and impacts during daytime operations," in *8th USA/Europe Air Traffic Management Research and Development Seminar (ATM2009), Napa, CA*, 2009.

[4] Y. Cao, T. Kotegawa, and J. Post, "Evaluation of continuous descent approach as a standard terminal airspace operation," in *9th USA/Europe Air Traffic Management R&D Seminar*, 2011.

[5] R. Alligier, D. Gianazza, and N. Durand, "Machine Learning and Mass Estimation Methods for Ground-Based Aircraft Climb Prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 6, pp. 1–12, 2015.

[6] J. Sun, J. Ellerbroek, and J. Hoekstra, "Modeling aircraft performance parameters with open ads-b data," in *Twelfth USA/Europe Air Traffic Management Research and Development Seminar*, 2017.

[7] J. Sun, J. Ellerbroek, and J. Hoekstra, "Bayesian inference of aircraft initial mass," in *Twelfth USA/Europe Air Traffic Management Research and Development Seminar*, 2017.

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

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

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

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

[12] L. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, pp. 338–353, jun 1965.

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