An adaptive hash-based text deduplication for ADS-B data-dependent trajectory clustering problem

Tan Tran
ICT Department
John von Neumann Institute
Ho Chi Minh, Vietnam
tan.tran2015@ict.jvn.edu.vn

Quan Duong
ICT Department
John von Neumann Institute
Ho Chi Minh, Vietnam
quan.duong2015@ict.jvn.edu.vn

Duc-Thinh Pham
Air Traffic Management Research Institute
School of Mechanical and Aerospace Engineering
Nanyang Technological University
Singapore, Singapore
dtpham@ntu.edu.sg

An Mai
ICT Department
John von Neumann Institute
Ho Chi Minh, Vietnam
an.mai@jvn.edu.vn

***Abstract*—The Automatic Dependent Surveillance-Broadcast (ADS-B) protocol is equipped in aircraft as an alternative to secondary radar. This emerging technology produces such a prospective type of data to effectively broadcast the aircraft's status (location, velocity, etc.,) in a specific area, which is very useful in air traffic management (ATM). However, there is still a limited number of advanced studies from machine learning/data mining perspectives relying on this kind of data in ATM research. On the other hand, Locality Sensitive Hashing (LSH) is a data mining technique often used to find similar items in the data with high-dimension properties. It is thus relatively suitable for handling with trajectories data to group similar flight paths. From these factors, we reveal in this paper an adaptive LSH-based algorithm, used in near-duplicated documents detection, for the problem of clustering the nearest trajectories by representing the trajectories as a bag-of-words used popularly in text mining. To illustrate our proposed method, an experiment is designed and carried out in thirty successive days, employing the raw ADS-B data collected from *FlightAware* for the case of Changi International Airport, Singapore. The evaluation based on Silhouette score shows promising results of measuring the clustering performance.**

***Index Terms*—ADS-B, Trajectory Clustering, Air Traffic Control, Locality Sensitive Hashing**

I. Introduction

The rapid growth of population across the world leads to extremely high demand on traveling by aviation, which makes the number of flights increases day by day rapidly and more quickly than system capacity. In order to increase the performance of air space usage, there were many researches related to increasing the air traffic system's capacity in the past [1], focusing mostly on the possible solutions for efficient and effective management of system capacity. These studies can be categorized into two levels: system and airport. There are two main topics: air traffic flow management and airspace research at system level, while airport capacity, airport facility utilization, aircraft terminal operations as well as aircraft ground operations research are the main focus of the airport level. Other works, which cover a broad range of activities, can be figured out from redesigning the national airspace for sustainability [2] to air traffic control modernization [3]. However, there is a lack of research leveraging the use of machine learning based techniques, and one of the most prominent approaches is aircraft trajectory analysis which has been widely studied in recent years. In which, the trajectory clustering analysis can be shown powerful and adaptive well with massive and complex trajectory data [4].

From the considered trajectory data perspectives, so far the main data type of aircraft trajectories is GPS generated from radars and owned by different individual organizations or countries which makes the data sharing for research become extremely cumbersome. Therefore, new generation of surveillance technique called ADS-B (Automatic Dependent Surveillance - Broadcast) was introduced, and most of the modern aircraft are required to be equipped by 2020 in United States, and the equipment was compulsory for some sample aircraft in Europe from 2017 [5]. Originally, ADS-B was developed to help the information sharing between national airspace systems become more efficient and transparent. Since it is a kind of protocol broadcasting the aircraft's location, it thus can be received by individual aircraft anywhere and anytime, and does not depend on national geographic. There are few researches on the infrastructure and the exploratory analysis regarding to this kind of data, for example, [6] discussed about security aspect of its implementation in the next generation of air transportation system. Or in [7], the authors proposed ADS-Bsec, which tackles the ADS-B limitations about the

Proceedings of the 13^{th} International Conference on Computing and Communication Technology (RIVF 2019), Danang, Vietnam.

missing of integrity or authenticity by using a key management infrastructure. Although it is a full of promise data generation technique, there is not too many publications regarding to the use case of this data in air traffic control and ADS-B data-dependent trajectories clustering was not clearly applied or extended recently. Therefore, it uncovers many interesting issues from the research topics of this field based on ADS-B data analysis perspectives.

Motivated by these realities, we propose to apply an adaptation from hash-based method for text deduplication to trajectory clustering and demonstrate its feasibility by carrying out an experiment based on ADS-B data. The paper is organized as follows: In section I, we discussed about the motivation for our study. In section II, we review the most related works to this paper before jumping into the proposed idea in section III. The experimental design and corresponding results taking into account real ADS-B data are placed in section IV. Finally, we come to a conclusion and future works in section V.

II. Related Works

According to Sridhar [8], the ATM system is constructed mainly from two components: Air Traffic Control (ATC) and Traffic Flow Management (TFM), which must be enable for safe and efficient performance in the airspace over the world. The ATC function ensures aircraft's movements must be safe and separated in all cases, while the TFM function controls aircraft in flow patterns for smooth performance of air-traffic. In other to accomplish these characteristics, trajectory clustering plays an important role in ATM which generally group similar trajectories for improving the efficient usability and safety of airspace, especially the trajectory of aircraft toward the airports. In 2014, the project OpenSky [9] was proposed and introduced as a sensor network based on low-cost hardware connected over the Internet in order to enable researchers to conduct experimental studies based on real data ADS-B, the OpenSky covers 720,000 km^2 and enable researchers to analyze billions of ADS-B messages. Based on this source, in 2016, Junzi Sun and the team [10] have leveraged various unsupervised learning techniques and fuzzy logic methods to identify flight phase, in which they re-used popular open-source libraries for dealing with large-scale aircraft data. Continue on this work, they had another publication [11] regarding to the extraction of different aircraft performance parameters from seven distinct flight phases. In order to detect commercial flight baselines, Lexical Link Analysis (LLA), a text mining technique expressing relationships and associations with the given data, was used in [12] to analyze aircraft tracks over a period time, then they were visualized with Google Earth and Maps. Based on the fundamental property of Long short-term memory (LSTM) networks, which do not need to manipulate the entire information but selectively remember patterns for long time duration, the authors in [13] proposed to apply LSTMs based sequence autoencoder to learn interesting features in order for detecting surveillance aircraft taking into account ADS-B data. For a more advanced study in [14], a proposed reinforcement learning model, based on a perception module generated by sensors, was presented with the its application in the field of air navigation relying on ADS-B technology.

There used to have many traditional but still useful researches of trajectory identification problems, such as in a typical research [15], the authors present a particular *partition-and-group* framework for clustering trajectories with the core algorithm called TRACLUS. Named by its nature, it consists of following step: *partitioning* - each trajectory is partitioned into a set of line segments, while for the second step *grouping* - similar line segments are grouped into a cluster determined by a density-based clustering such as DBSCAN, which can discover minority of main trajectories and also detect the outliers. Finally, the authors provided several experiments on various real data to show the effectiveness of their algorithm. Identify and remove outlier trajectories is another story motivated by its impact in practice, this group of authors continued their previous research in [16] to introduce an outliers detection method by first using a step of trajectory partition into a set of line segments, then all outliers line segments are detected in a second detection step. This whole process hence is called *partition-and-detect*.

According to Gariel [17], the standard procedures are mostly used by air traffic controllers to guide aircraft, ensure the safety of the airspace and maximize the runway occupancy. Moreover, the trajectories mined from the recorded radar tracks is also used to monitor conformity of current operations against operations previously identified as standard. The authors subsequently proposed two methods to cluster trajectories and identify flight paths: The first approach is based on the identification of way-points in trajectories, and the remaining approach is based on PCA (Principal Components Analysis) of resampled trajectories.

In 2017, trajectory clustering approach with relevant parts was introduced in [18] by Andrienko and the team. They proposed an analytical workflow with a corresponding class of computational, visual, and interactive techniques, whose core is density-based clustering of trajectories focusing on the relevant parts while keeping the integrity of available trajectories. Also, it is necessary to concentrate on the initial or final parts of the flights (take-off or landing phase), or to ignore these parts and consider the variety of the paths from the origins to the destinations, or to deal with all the parts of the flight within a certain area or volume in the airspace. Relevance-aware clustering is the main flow of this paper, in which the authors proposed visualization guidelines for supporting interactive selection in tasks-relevant parts of trajectories then using the density-based algorithms to cluster trajectories. Final results of clustering framework are often represented in visualization step, therefore filtering before clustering was required in this study.

The computational complexity in trajectory clustering also need to be considered in order to obtain efficient clustering performance, The authors suggested in [19] an adaptive trajectory clustering method based on grid and density which can archive better results, when compared with the

TRACLUS, with difference between the adaptive calibration and the optimal is less than 5% and the running time of clustering can reduce approximately 95%. This approach is pretty suitable for handling with large datasets, especially for vehicle trajectories from intelligent systems. Basically, the ideas of this paper is quite simple: All of segments are mapped into the corresponding cells, then it calculates the average distance among the different segments in each grid cell, and the average number of the trajectory segment in each cell. Finally, an DBSCAN-based clustering is employed to carry out the adaptive parameter calibration for achieving effective clustering performance.

However, these traditional approaches require high computational cost for pair-wise distance of trajectories, which directly makes the clustering process very expensive. In 2016, Ivan Sánchez [20] introduced such an impressive methodology of trajectory clustering based on hashing techniques, from applying general family of hash functions, which can map the trajectories to different buckets with an assumption that is all trajectories must have the same length. These ones represents for neighbor points that will fall into same bin with high probability. Therefore, the computational cost of hashing process of trajectories will be in linear time complexity. Their experiments showed that the LSH-FH can achieve competitive accuracy as K-means using DTW distance. And this consequently motivates for the works in this paper.

III. Methodology

In this section, we will describe our proposed trajectory algorithm based on an adaptation of LSH for text deduplication and trajectory simplification via Douglas - Peucker algorithm.

A. Text Deduplication with Locality Sensitive Hashing (LSH)

The traditional approaches to compare, detect, retrieve or even remove near-duplicate documents, used often in web mining, are mostly compute the pairwise similarity of every documents, with average complexity is $O(n^2)$ (see in the book *Mining Massive Datasets - Chapter 3* [21]). To tackle the issue of high complexity computation, LSH (Locality Sensitive Hashing) was developed to achieve an efficient performance.

General Ideas: LSH is based on a family of hash functions which can convert data points to hash signature values, then these signatures will be allocated into buckets which has a property: similar data points will have hash values in the same buckets with high probability, while dissimilar data points will have hash values in different buckets. Here is following steps for documents deduplication from LSH technique perspectives:

- Represent each document by the set of tokens which is the input format for applying similar documents clustering via LSH.
- Define a k-shingle for a document as any substring of length k found within the document. Therefore, two documents could appear to have shingles in common if they are similar.
- The next step is compute the small hashing signature of each document (d) using a hash function H of LSH family. The essential key here is that the H function have the following properties: If $s(document_1, document_2)$ is **high** (**small**), then there is a **high** (**small**) probability that $H(document_1) = H(document_2)$ respectively, where s is a similarity function of two documents.

B. Trajectory Simplification with Douglas - Peucker algorithm

The intuition of **Douglas - Peucker** algorithm is reconstruct a given curve composed from original data points into a similar curve with fewer points. The algorithm defines *"dissimilarity"* based on the maximum distance between the original curve and the simplified curve by the *Hausdorff* distance $\mathcal{H}(P_i, P_j)$ applied between two trajectory curves P_i and P_j. The simplified curve consists of a subset of the set of data points representing the original curve [22]. The pseudo-code is described in Algorithm 1.

Algorithm 1: $DouglasPeucker(t, \epsilon)$

Inputs:

A given trajectory t is represented by a list of m spatial points P_j^t where $j = 1 : m$. Each point includes latitude and longitude coordinates.
Initialize a small float number $\epsilon > 0$.

Procedure:

for $j = 2$ **to** $m - 1$:
  $d_{max} = \mathcal{H}(P_j^t, Curve(P_1^t, P_m^t))$
if ($d_{max} > \epsilon$):
  // recursively simplify
  $List_1 = DouglasPeucker(t[P_1^t ... P_{index}^t], \epsilon)$
  $List_2 = DouglasPeucker(t[P_{index}^t ... P_m^t], \epsilon)$
  return $\hat{t} = Curve(List_1, List_2)$
else:
  return $\hat{t} = Curve(P_1^t, P_m^t)$

Output:

  Simplified Trajectory $\hat{t}$

For an illustration, we display a sample of original and simplified trajectory coordinates of a flight path in figures 1.

C. Proposed Trajectory Clustering algorithm based on adaptive LSH for text deduplication

To tackle the high-cost issue of trajectory clustering by computing the distance matrix between the trajectories, different from Ivan Sánchez [20] and the team, first we use directly a clustering algorithm such as K-means to determine the numbers of covering clusters of air-space towards the airport. Each trajectory will be allocated into some clusters afterward, and can be represented in fixed-dimensional space. Then we can easily apply hashing techniques described in subsection III-A to detect duplicated trajectories in linear-dependent time without requiring all trajectories having the same length.

Specially, it starts from covering the spatial trajectories by a certain number of group-cells on the map (known as clusters),

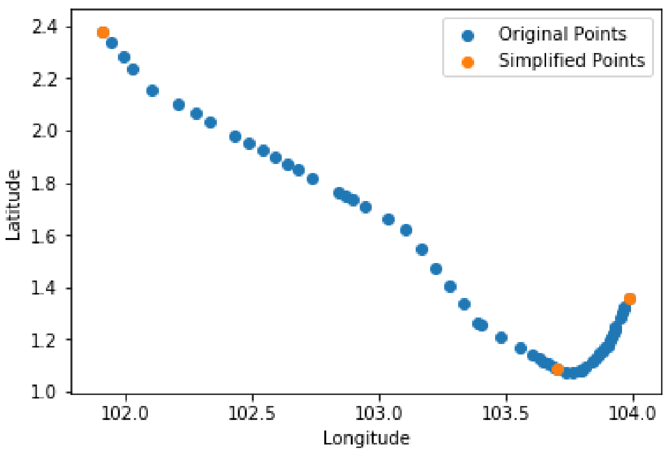

Fig. 1: A sample of original and simplified trajectory coordinates using algorithm 1

which can be considered as document's bag-of-words (a set of tokens). It means that the representation of a trajectory is an array of tokens, for which the hash signature values can be computed time-linearly via LSH. After that, these signatures are stored in buckets, and hence the clustering results could be performed by grouping all trajectories in the same bucket. Here, in order to force the hash signature of these trajectories falling into the same buckets with high probability, we let those trajectories having small distance between them have high probability in a same bucket. This could be carried out by using **Douglas - Peucker** algorithm presented in subsection III-B to simplify the original data points. In detail, our proposed algorithm is summarized in Algorithm 2.

As we can see in figure 2, each data point (including *Latitude* and *Longitude*) belongs to a cluster, for example, in case of $K = 30$, each point will be assigned to a cluster C_i for $i \in \{1, 2, ..., 30\}$. Note that each point must belong to a trajectory flight, so each trajectory now can be represented by a **set of clusters**. From this observation, we can consider each trajectory as a bag-of-word with the vocabulary $\{C_1, C_2, \ldots, C_K\}$.

IV. Experimental results

A. Data Description

The data used in this section is collected from *FlightAware*, which presents the *latitude*, *longitude*, *altitude*, *velocity* in time series format of aircraft. *FlightAware* is the world's largest flight tracking data company and provides global flight tracking solutions. To demonstrate our proposed algorithm, we carry out an experiment based on ADS-B data of filtered trajectories towards the Changi International Airport, Singapore (International Civil Aviation Organization Code: *WSSS*) with the radius of the considered region is 500 km, and for thirty successive days. Specifically, we only use the *latitude* and *longitude* attributes, extracted from the filtered ADS-B data.

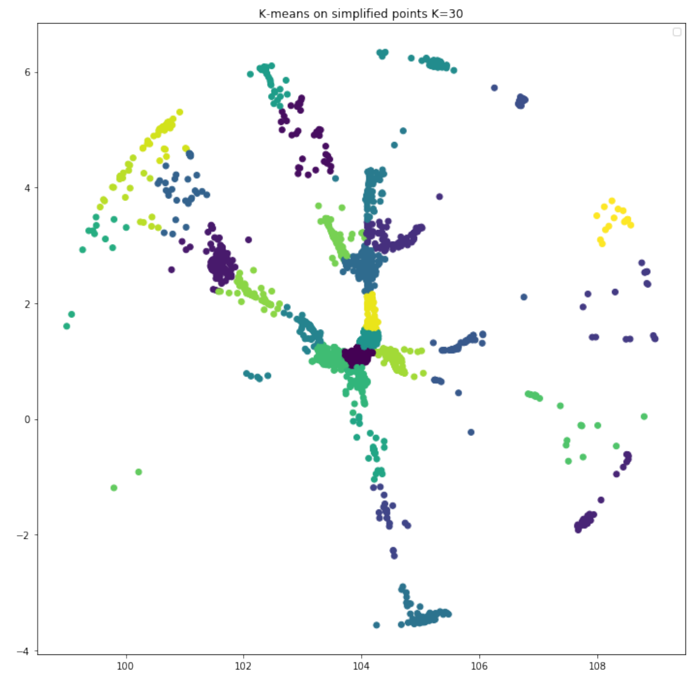

Fig. 2: K-means with $K = 30$ on simplified trajectories of 323 flights toward Changi International Airport, Singapore in 2016-09-29

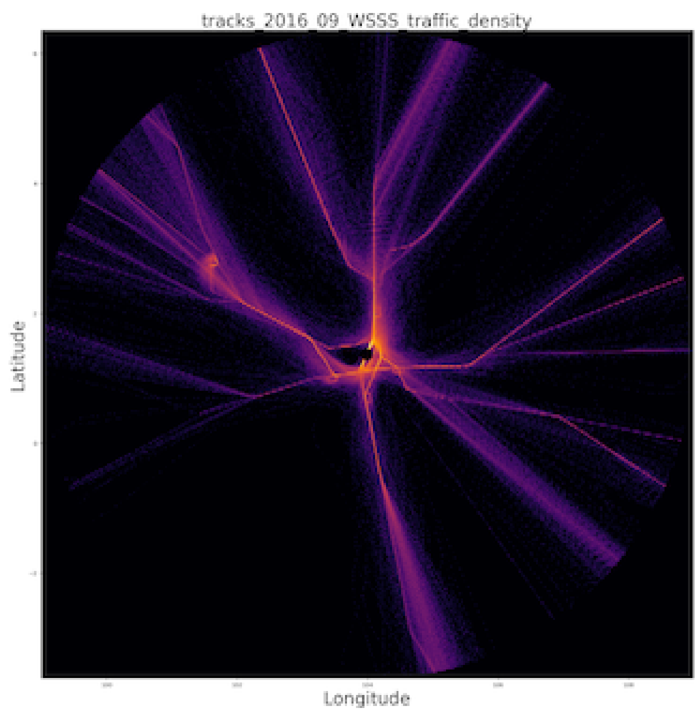

Fig. 3: Example of the density of trajectory coordinates at Changi International Airport, Singapore in 2016-09-29. From this, we can imagine roughly the shapes of arrival flight paths (arrival flows)

Algorithm 2: Adaptive LSH-based Trajectory Clustering

Inputs:

Give a set of N trajectories $t_1, t_2, ..., t_N$. Each t_i is represented by a list of m_i spatial points $P_j^{t_i}$ with $j = 1 : m_i$, and each point $P_j^{t_i}$ includes latitude and longitude coordinates.

The total number of spatial points: $M = \sum_{n=1}^{N} m_n$.

K is the number of grid cells for grouping all points.

Initialize a small float number $\epsilon > 0$.

Procedure:

```
for i = 1 to N:
    Simplify trajectory: t_i = DouglasPeucker(t_i, ε)
    Reduce m_i to m_i^reduce, with m_i^reduce << m_i
Apply & tune K-means on M spatial points:
→ K clusters: {C_k}_{k=1}^K (K is large enough).
for i = 1 to N:
    for j = 1 to m_i^reduce:
        Assign P_j^{t_i} to one of K clusters.
for i = 1 to N:
    Coverage(t_i) = {C_1^i, ..., C_{K_i}^i}
    with {C_k^i}_{k=1}^{K_i} ⊂ {C_k}_{k=1}^K and 2 < K_i < K.
for i = 1 to N:
    for j = 1 to K_i:
        T(C_j^i) ← Tokenize(C_j^i)
for i = 1 to N:
    Represent t_i by: t_i^doc = List{T(C_1^i), ..., T(C_{K_i}^i)}.
    Compute hash signature: h_i = LSH(t_i^doc).
    Store h_i in B buckets.
```

Output:

```
Indices of all trajectories in B buckets
for b = 1 to B:
    Retrieve a list of trajectories in {t_i^doc}_{i=1}^N for h_i ∈ b
```

B. Results Evaluation

For the interpretation and validation of clustering results, Silhouette score is employed in this study. The main ideas of this measure is scoring how similar data points in their covering cluster when compared to other clusters. The Silhouette value is a real number in a range from -1 to 1, where the best value is 1 showing that the object is well-matched to its covering cluster, while -1 is the worst one, and overlapping clusters will produce near 0 values [23].

We summarize the experimental results, executed in a single machine with Intel core i5 CPU and 8GB of RAM, for thirty successive days in table I. Experimentally, our proposed algorithm runs fast with stable Silhouette scores over time, and an acceptable number of clusters was detected as a data-driven recommendation for air traffic controller in practice. In order to illustrate visually the trajectory clustering result as the output results of our proposed algorithm, we select the one showing highest Silhouette score (the experiment performed in 2016-09-29 at Changi International Airport, Singapore with Silhouette score is **0.771** and number of detected clusters here is **8**) then visualize its density of trajectory coordinates in figure 3 and corresponding trajectory clusters in figure 4.

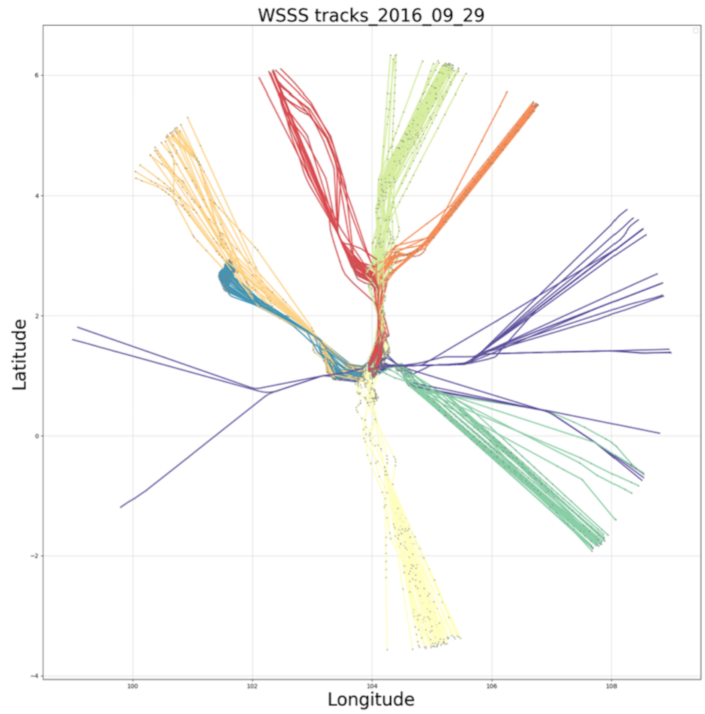

Fig. 4: Trajectory Clustering visualization of arrival flows toward Changi International Airport, Singapore in 2016-09-29, as the result of our proposed algorithm

V. Conclusions

In this work, we propose an adaptive LSH-based algorithm, which originally applies for removing duplicated query-documents in web mining, to detect the near trajectory paths and cluster them in a same group. This can be used to simplify the structure of arrival flight paths toward an airport, which is a crucial problem in air traffic control. We then verify this algorithm by using the filtered ADS-B data of Changi International airport for thirty successive days and obtain quite impressive results. Moreover, via this thorough study, we would like to indirectly emphasize the potential of ADS-B data which could lead to many interesting researches regarding to AI applications to air traffic management. Lastly, available source codes of our proposed algorithm and experiments are published in Github[1]. In the future, we will extend our study for more sophisticated use case when there is a large number of arrival flows to the airport simultaneously, which requires the incorporation of route scheduling as well.

Acknowledgment

The authors would like to thank John von Neumann Institute, Viet Nam National University and Tenpoint7 LLC for supporting this work.

[1] https://github.com/tanthml/atc

TABLE I: The clustering results applied for the filtered ADS-B data of arrival flight paths of Changi International Airport on thirty successive days by using our proposed algorithm 2. Here, we obtain $K = 30$.

Date	**No. Flights**	**Silhouette Score**	**No. Clusters**	**No. Original Points**	**No. Simplified Points**	**Running Time** (second)
2016-09-01	316	0.604	7	14240	4576	4
2016-09-02	341	0.642	8	15426	4610	4
2016-09-03	314	0.714	8	14695	3995	4
2016-09-04	345	0.758	8	15999	5116	5
2016-09-05	320	0.728	9	14358	4631	4
2016-09-06	311	0.612	8	14090	4578	4
2016-09-07	320	0.675	7	14491	4272	4
2016-09-08	335	0.704	10	15177	4654	4
2016-09-09	337	0.670	9	15852	4866	4
2016-09-10	307	0.593	8	14279	3958	4
2016-09-11	335	0.540	7	15669	4918	4
2016-09-12	326	0.630	9	15412	4575	4
2016-09-13	299	0.655	10	13571	3804	4
2016-09-14	308	0.713	8	14076	4301	4
2016-09-15	317	0.734	9	14435	4585	4
2016-09-16	341	0.643	8	15590	4652	4
2016-09-17	305	0.686	6	14138	3866	4
2016-09-18	336	0.658	6	15400	4425	4
2016-09-19	331	0.608	7	15439	4644	4
2016-09-20	317	0.674	7	14840	4203	4
2016-09-21	319	0.624	6	14542	4496	4
2016-09-22	315	0.685	7	14316	4086	4
2016-09-23	349	0.534	6	16179	4391	4
2016-09-24	317	0.701	8	14490	4099	4
2016-09-25	332	0.671	7	15207	4250	4
2016-09-26	319	0.660	8	14279	4540	4
2016-09-27	310	0.627	7	13837	4191	4
2016-09-28	299	0.618	8	13226	4061	4
2016-09-29	323	**0.771**	**8**	14313	4463	4
2016-09-30	332	0.710	9	15390	5117	5

REFERENCES

[1] C.-L. Wu and R. E. Caves, “Research review of air traffic management,” *Transport Reviews*, vol. 22, no. 1, pp. 115–132, 2002.

[2] J. Andrews, E. Burke, and J. Thomas, “Redesigning the national airspace system for sustainability,” *Summer Task Force on Air Transportation, Princeton University, Princeton, NJ*, 2004.

[3] G. Gibbons, “Air traffic control modernization: Faa, nextgen, gnss, and avionics equipage,” *Inside GNSS*, vol. 30, no. April, pp. 1–6, 2011.

[4] J. Bian, D. Tian, Y. Tang, and D. Tao, “A survey on trajectory clustering analysis,” *arXiv preprint arXiv:1802.06971*, 2018.

[5] E. A. Lester, “Benefits and incentives for ads-b equipage in the national airspace system,” Ph.D. dissertation, Massachusetts Institute of Technology, 2007.

[6] D. McCallie, J. Butts, and R. Mills, “Security analysis of the ads-b implementation in the next generation air transportation system,” *International Journal of Critical Infrastructure Protection*, vol. 4, no. 2, pp. 78–87, 2011.

[7] T. Kacem, A. Barreto, D. Wijesekera, and P. Costa, “Ads-bsec: A novel framework to secure ads-b,” *ICT Express*, vol. 3, no. 4, pp. 160–163, 2017.

[8] B. Sridhar, K. S. Sheth, and S. Grabbe, “Airspace complexity and its application in air traffic management,” in *2nd USA/Europe Air Traffic Management R&D Seminar*, 1998, pp. 1–6.

[9] 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*. IEEE Press, 2014, pp. 83–94.

[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] J. H. Junzi Sun, Joost Ellerbroek, “Modeling aircraft performance parameters with open ads-b data,” in *Proceedings of the 12th USA/Europe Air Traffic Management Research and Development Seminar*. FAA/EUROCONTROL, 2017.

[12] R. Salcido, A. Kendall, and Y. Zhao, “Analysis of automatic dependent surveillance-broadcast data,” in *AAAI Technical Report*. AAAI, 2017, pp. 225–230.

[13] T. N. Brooks, “Using autoencoders to learn interesting features for detecting surveillance aircraft,” *arXiv preprint arXiv:1809.10333*, 2018.

[14] S. Álvarez de Toledo, A. Anguera, J. M. Barreiro, J. A. Lara, and D. Lizcano, “A reinforcement learning model equipped with sensors for generating perception patterns: Implementation of a simulated air navigation system using ads-b (automatic dependent surveillance-broadcast) technology,” *Sensors*, vol. 17, no. 1, p. 188, 2017.

[15] J.-G. Lee, J. Han, and K.-Y. Whang, “Trajectory clustering: a partition-and-group framework,” in *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*. ACM, 2007, pp. 593–604.

[16] J.-G. Lee, J. Han, and X. Li, “Trajectory outlier detection: A partition-and-detect framework,” in *Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on*. IEEE, 2008, pp. 140–149.

[17] 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.

[18] G. Andrienko, N. Andrienko, G. Fuchs, and J. M. C. Garcia, “Clustering trajectories by relevant parts for air traffic analysis,” *IEEE transactions on visualization and computer graphics*, 2017.

[19] Y. Mao, H. Zhong, H. Qi, P. Ping, and X. Li, “An adaptive trajectory clustering method based on grid and density in mobile pattern analysis,” *Sensors*, vol. 17, no. 9, p. 2013, 2017.

[20] I. Sánchez, Z. M. M. Aye, B. I. P. Rubinstein, and K. Ramamohanarao, “Fast trajectory clustering using hashing methods,” *2016 International Joint Conference on Neural Networks (IJCNN)*, pp. 3689–3696, 2016.

[21] J. Leskovec, A. Rajaraman, and J. D. Ullman, *Mining of massive datasets*. Cambridge university press, 2014.

[22] D. H. Douglas and T. K. Peucker, “Algorithms for the reduction of the number of points required to represent a digitized line or its caricature,” *Classics in Cartography: Reflections on Influential Articles from Cartographica*, pp. 15–28, 2011.

[23] P. J. Rousseeuw, “Silhouettes: a graphical aid to the interpretation and validation of cluster analysis,” *Journal of computational and applied mathematics*, vol. 20, pp. 53–65, 1987.