Trajectory Pattern Identification for Arrivals in Vectored Airspace

Chuhao Deng, Kwangyeon Kim, Hong-Cheol Choi, Inseok Hwang
Aeronautics and Astronautics Engineering
Purdue University
West Lafayette, IN, 47907 USA
{deng113, kim1601, choi642, ihwang}@purdue.edu

2021 IEEE/AIAA 40th Digital Avionics Systems Conference (DASC) | 978-1-6654-3420-1/21/$31.00 ©2021 IEEE | DOI: 10.1109/DASC52595.2021.9594503

***Abstract*—Grouping similar flight trajectories into a cluster, or a pattern, is an important data preprocessing step for data-driven methods in air traffic control as it affects the performance of applications such as separation assurance, anomaly detection and trajectory prediction. Especially in vectored airspace, it is challenging to obtain trajectory patterns as they are deeply embedded in the air traffic data. In this paper, we propose a clustering framework that can identify deeply embedded trajectory patterns in vectored airspace from the historical surveillance data using agglomerative hierarchical clustering and dynamic time warping. The proposed framework is illustrated with the Automatic Dependent Surveillance-Broadcast (ADS-B) data collected in the Incheon International Airport (ICN), South Korea.**

***Index Terms*—trajectory clustering, dynamic time warping, hierarchical clustering**

I. Introduction

In the Air Traffic Management (ATM) system, Air Traffic Controllers (ATCs) assist aircraft operations to ensure the safety and efficiency by monitoring and commanding their movements using radar, computers, or visual references [1, 2]. As the demand and complexity of air traffic increase, maintaining such safety and efficiency of the operations in airspace could lead to an increased workload for ATCs and delays in their decision-making processes. In this regard, it is crucial to develop assistant tools for the decision-making processes, for which significant research efforts have been made.

There are, in general, two kinds of methods for developing the assistant tools: physics-based methods and data-driven methods. Physics-based methods focus on studying the dynamics of aircraft or the airspace system, while data-driven methods focus on studying historical flight data. With the advancement of data collecting and processing technologies, data-driven methods have become more and more popular and have been employed for many applications. For aircraft separation assurance, Hawley et al. [3] proposed a reinforcement learning based algorithm for predicting and mitigating potential loss of separation events. For anomaly detection, Janakiraman and Nielsen [4] proposed an algorithm for anomaly detection in aviation data by implementing the extreme learning machines. Raj et al. [5] proposed an anomaly detection algorithm using temporal logic learning, which generates temporal logic formulas that are easy to be interpreted by human and can be used in real-time monitoring for safety, and they further applied the algorithm to precursor detection for terminal airspace operations [6]. Matthews et al. [7] used scalable data mining algorithm that searches anomalies on large-scale datasets to discover anomalous aviation safety events. Similarly, Li et al. [8] analyzed the flight data to detect abnormal operations using clustering techniques. Other than anomaly detection, data-driven methods have also been used for aircraft trajectory prediction by using probabilistic trajectory models such as Gaussian Mixture Model (GMM) [9, 10] or machine learning models [11], and the identification of traffic flow patterns using supervised learning methods [12].

Since trajectories in different patterns have distinct characteristics, all studies mentioned above rely on trajectory clustering, that is, grouping similar trajectories into a cluster, for data preprocessing, using density-based method, such as Density-Based Spatial Clustering of Application with Noise (DBSCAN) [3, 5, 6, 8, 9, 11, 12], or distance-based method, such as k-means [4, 7, 10]. To obtain better clustering results, many studies proposed new and novel clustering methods. To improve the existing k-means algorithm, Sinaga and Yang [13] proposed an unsupervised k-means clustering algorithm that can automatically find an optimal number of clusters without giving any initialization and parameter selection. Shi et al. [14] proposed an adaptive clustering algorithm based on k-nearest neighbors and density to achieve the similar goal. For clustering of flight trajectories, Peng et al. [15] proposed a trajectory clustering algorithm based on feature representation and selection. However, selecting features from high-dimensional data is time consuming, so Elankavi et al. [16] proposed the Fast Clustering Algorithm that can select features more efficiently by using the minimum spanning tree to remove irrelevant features from the data before clustering. Enriquez [17] presented spectral clustering that, unlike other clustering algorithms, considers not only spatial information, but also temporal values. Liu et al. [18] proposed an improved trajectory clustering method based on fuzzy DBSCAN that implements soft constraints in the conventional DBSCAN method. By using autoencoders, Olive et al. [19] proposed a clustering algorithm that extracts the hidden features of trajectories with autoencoders and then clusters trajectories based on their representations in the low-dimensional latent space.

II. Objectives and Contributions

Clustering is a crucial step in data preprocessing as it can affect the performance of many data-driven methods. Many clustering algorithms have been proposed and applied to the trajectories in the entire terminal airspace of an airport or the airspace that is very close to the runway. However, to the best of our knowledge, none of them focused on identifying various trajectory patterns within one procedure in Standard Terminal Arrival Route (STAR) or Standard Instrument Departure (SID). In terminal airspace operations, aircraft are frequently instructed to deviate from the flight procedures by ATCs to accommodate given traffic situations, e.g., maintaining the separation from neighboring aircraft or taking shortcuts to meet scheduling requirements. Such deviation is called vectoring [20]. In such vectored airspace, where trajectories are often overlapped in the horizontal plane with subtle differences, algorithms like DBSCAN and k-means, which have been widely applied and proven successful in many other tasks, do not perform well since they do not directly measure the dissimilarity between trajectories and consider the time discrepancies in the time-series data. To address these issues, we propose a new framework for trajectory pattern identification based on Dynamic Time Warping (DTW), for calculating the dissimilarity between trajectories with minimal time discrepancies, and agglomerative hierarchical clustering, for clustering the trajectories one by one based on their dissimilarity measure.

The contribution of this paper is to propose a new clustering framework that can be applied to the identification of deeply embedded trajectory patterns in vectored airspace from historical air traffic surveillance data. With all trajectory patterns frequently observed in the data known, ATCs' workload and delays in their decision making process can be reduced, thereby improving safety and efficiency of air traffic operations.

III. Surveillance data in vectored airspace

In Figure 1, the STARs to the Incheon International Airport (ICN) are presented with the entry fixes denoted as yellow dots (REBIT, OLMEN, GUKDO, KARBU, and SEL) and the Initial Approach Fixes (IAF) denoted as green dots (KOTRA, PULUN, TIMON, and DANAN) where the arrival flights begin the approach. In the Aeronautical Information Publication (AIP) data, the sequence of waypoints, or fixes, are defined for each route from an entry fix to an IAF, blue lines in Figure 1, and from an IAF to a runway, red lines in Figure 1. The routes shown in Figure 1 use area navigation (RNAV) that allows aircraft to operate on any path within the coverage of navigation aids, and hence the aircraft are frequently instructed by ATCs to deviate from the routes for the purpose of separation or sequencing. Due to such instructions, called *vectoring*, many flight trajectories in vectored airspace do not precisely follow the routes, which makes any trajectory patterns deeply embedded in the surveillance data, as shown in Figure 2 where the flight trajectories and the routes are shown for the routes GUKDO 1N and GUKDO 1P, both entering

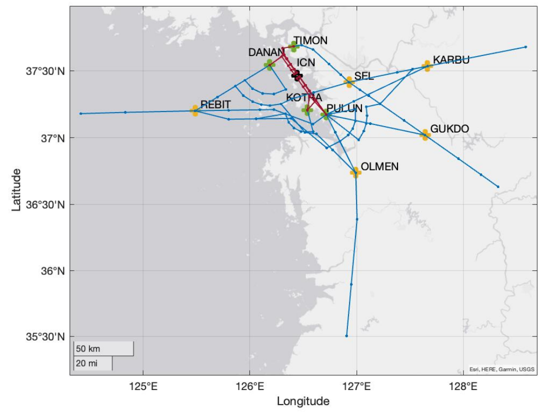


Fig. 1: STARs to the ICN

through GUKDO and approaching to ICN from the North and the South, respectively.

In this paper, we use the flight trajectories from the historical Automatic Dependent Surveillance Broadcast (ADS-B) data for the arrival flights to ICN, collected from January to December in 2019. The ADS-B data record the aircraft's states such as time, position (longitude, latitude, and altitude), and speed (ground speed and vertical rate). We consider the flight trajectories that pass through GUKDO as it is one of the most popular entry fixes in ICN and has many overlapped trajectory patterns, as shown in Figure 2. To identify the trajectory patterns for a given route in STAR, we first separate the trajectories that follow GUKDO 1N (landing from the North) and GUKDO 1P (landing from the South) using the distance from the last recorded position to the position of ICN, as shown in Figure 3. The total numbers of trajectories in GUKDO 1N and GUKDO 1P are 2,967 and 8,355, respectively.

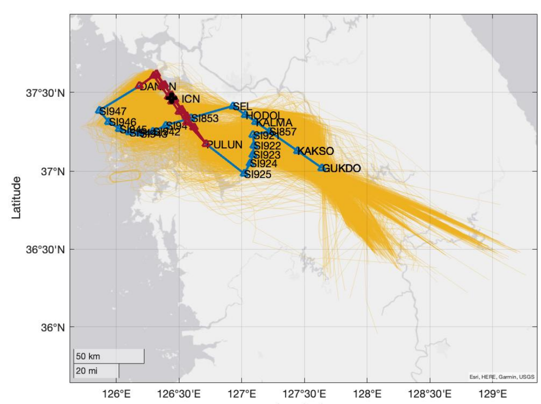
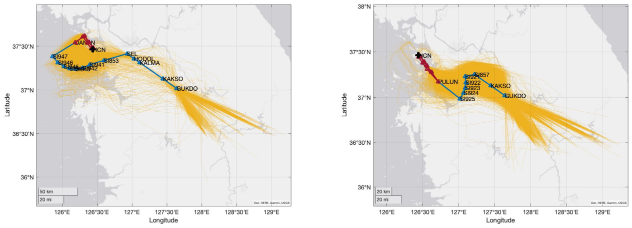


Fig. 2: Flights that pass through GUKDO

IV. Methodology

In Figure 4, the proposed framework for trajectory pattern identification is shown, which consists of (i) algorithm-based



(a) Trajectories in GUKDO 1N (b) Trajectories in GUKDO 1P

Fig. 3: Trajectories in GUKDO 1N and 1P

clustering and (ii) domain knowledge-based clustering. By taking as input, a set of trajectories that follow a certain STAR, the algorithm-based clustering generates preliminary trajectory patterns that are well-separated, i.e., all trajectories within one pattern follow the same set of waypoints. Then, the domain knowledge-based clustering combines similar patterns based the waypoints that they follow in order to make the final trajectory patterns more interpretable and useful for ATCs. The main challenges in identifying the trajectory patterns in a vectored airspace come from (i) the nature of the air traffic surveillance data that have time discrepancies in the recorded time-series and (ii) the fact that the trajectory patterns are deeply embedded in the data. In this regard, we first explain the core components that can address such challenges, the dynamic time warping (DTW) for (i) and agglomerative hierarchical clustering for (ii), followed by the detail of the entire workflow of the proposed framework.

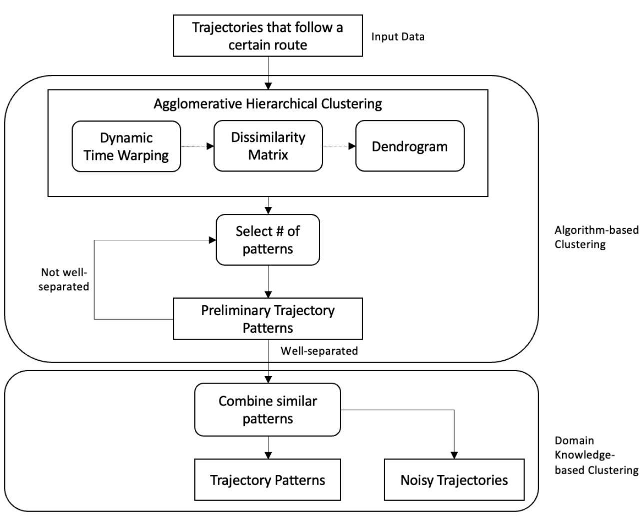


Fig. 4: Proposed trajectory pattern identification framework

A. Dynamic Time Warping

Clustering means grouping similar trajectories together, which requires a measure for their similarities. We define the dissimilarity measure as a numerical measure of how different two trajectories are. Although many methods for computing the dissimilarity measure such as the Euclidean distance exist, most air traffic surveillance data were recorded in real-time at different rates, meaning that there are time discrepancies between the trajectories. In this regards, we use a technique called Dynamic Time Warping (DTW) [21] that can measure similarity between two temporal sequences with time discrepancies. Suppose we have two trajectories, TR^1 and TR^2 with k features and m and n time steps, respectively.

$$TR^1 = \begin{bmatrix} tr^1_{11} & tr^1_{12} & \dots & tr^1_{1m} \\ \vdots & \vdots & \ddots & \vdots \\ tr^1_{k1} & tr^1_{k2} & \dots & tr^1_{km} \end{bmatrix}$$

$$TR^2 = \begin{bmatrix} tr^2_{11} & tr^2_{12} & \dots & tr^2_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ tr^2_{k1} & tr^2_{k2} & \dots & tr^2_{kn} \end{bmatrix}$$

For a feature $(\cdot) \in \{1, \cdots, k\}$, we denote the distance between TR^1 at time step i, $tr^1_{(\cdot)i}$, and TR^2 at time step j, $tr^2_{(\cdot)j}$, as d_{ij}, for $i \in \{1, \cdots, m\}$ and $j \in \{1, \cdots, n\}$. Then, a distance matrix with every combination of i and j can be constructed as:

$$\begin{bmatrix} d_{1n} & d_{2n} & & \dots & & d_{mn} \\ & \ddots & & & \iddots & \\ \vdots & & & d_{ij} & & \vdots \\ & \iddots & & & \ddots & \\ d_{12} & d_{22} & & & & \\ d_{11} & d_{21} & & \dots & & d_{m1} \end{bmatrix}$$

With the distance matrix built, a $path$, or warping function, that starts from d_{11} and ends at d_{mn} can be drawn as shown in Figure 5 where the time steps of TR^1 and TR^2 are presented in the horizontal and vertical axes, respectively.

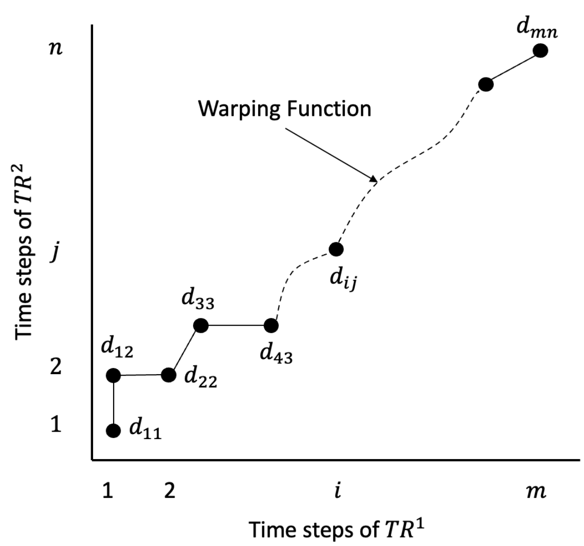


Fig. 5: Warping function in a distance matrix

The warping function is defied as a series of elements in the distance matrix that starts from d_{11} and ends at d_{mn}. By summing up every element in the distance matrix that composes the warping function, the distance between TR^1 and TR^2, is computed. Essentially, DTW is an algorithm that constructs the distance matrix and finds the warping function that delivers the minimal distance value by iteratively connecting the (i,j) element in the distance matrix to the element with the smallest value among the three possible ones, $(i+1,j)$, $(i,j+1)$ and $(i+1,j+1)$, from $(1,1)$ to (m,n) [21], shown in Figure 6. Noted that the length of the warping function does not have to be equal to m or n. As shown in Figure 6, points in the warping function can move horizontally (from d_{ij} to $d_{(i+1)j}$) or vertically (from d_{ij} to $d_{i(j+1)}$), which allows DTW to $align$ the trajectories for each time step to minimize the time discrepancies. If there are no time discrepancies between two trajectories, the warping function coincides with a diagonal line in the distance matrix [21]. In Figure 7, an example of the longitude of two trajectories with 56 and 89 data points being stretched to 114 data points, and their warping function are presented. Some horizontal and vertical lines can be observed in the lower plot in Figure 7(a), meaning that one of the trajectories is being stretched during that time to align with the other trajectory. With the characteristics mentioned above, DTW can compute the dissimilarity between trajectories without the need of resampling, which could lead to distortions of the trajectories, and with the minimal time discrepancies.

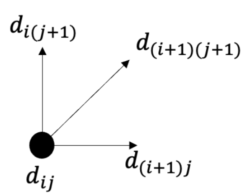


Fig. 6: Three possible directions for each element in the warping function

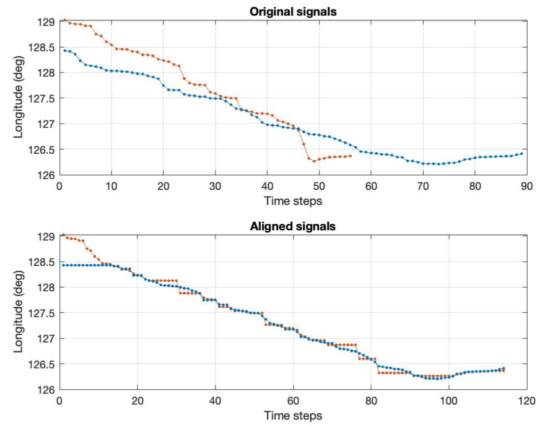
When the dissimilarity measure with minimal time discrepancies is computed, we can build a dissimilarity matrix that accurately represents the dissimilarity measure between trajectories. Let the set of all trajectories be $TR = \{TR^1, TR^2, TR^3, ..., TR^N\}$, where N is the total number of trajectories, and the dissimilarity between two trajectories computed using DTW is $DTW(TR^i, TR^j)$ for all $i,j \in \{1, \cdots, N\}$. Therefore, the dissimilarity matrix is defined as follows, which will be used in the agglomerative hierarchical clustering:

$$\begin{bmatrix} DTW(TR^1, TR^1) & \dots & DTW(TR^1, TR^N) \\ \vdots & \ddots & \vdots \\ DTW(TR^N, TR^1) & \dots & DTW(TR^N, TR^N) \end{bmatrix}$$

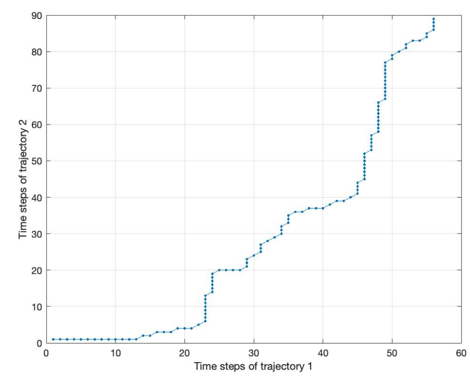
B. Agglomerative Hierarchical Clustering

Agglomerative hierarchical clustering is a bottom-up clustering algorithm that first treats each trajectory as a single cluster, and then uses a linkage method that measures the dissimilarity between the clusters to link the pair of clusters with the minimal dissimilarity. The linking process is repeated until only one large cluster that contains all the trajectories is remained. Common linkage methods are single, complete, average, centroid and Ward's [22]. Each method has its own formula for computing the dissimilarity between pairs of clusters. Let the dissimilarity between two clusters, x and y, be defined as $d(x,y)$, n_x is the number of objects in cluster x, and k_i^x is the i^{th} object in cluster x. Furthermore, the distance between i^{th} and j^{th} objects in cluster x and y is defined as $dist(k_i^x, k_j^y)$. Since the objects are trajectories in this paper, $dist(k_i^x, k_j^y) = DTW(TR^a, TR^b)$, i.e., k_i^x is TR^a and k_j^y is TR^b. The formula for each linkage method to compute $d(x,y)$ is shown in Table I [23–26].

Once the linking process is completed, that is, only one cluster with all trajectories is remained, we can form a tree-based representation (or a dendrogram) of the trajectories. An example of a dendrogram with 50 trajectories, constructed using the Ward's linkage method, is shown in Figure 8. In Figure 8, the vertical axis represents the dissimilarity measure and the horizontal axis represents the trajectories. It is shown



(a) Longitude of two trajectories being aligned by DTW



(b) Warping function

Fig. 7: An example of applying DTW to the longitude of two trajectories

TABLE I: Formulas of common linkage methods

Linkage Method	Formula
Single	$min(dist(k_i^x, k_j^y)), i \in \{1, \cdots, n_x\}, j \in \{1, \cdots, n_y\}$
Complete	$max(dist(k_i^x, k_j^y)), i \in \{1, \cdots, n_x\}, j \in \{1, \cdots, n_y\}$
Average	$\frac{1}{n_x n_y} \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} dist(k_i^x, k_j^y)$
Centroid	$\lVert \frac{1}{n_x} \sum_{i=1}^{n_x} k_i^x - \frac{1}{n_y} \sum_{j=1}^{n_y} k_j^y \rVert$
Ward's	$\sqrt{\frac{2 n_x n_y}{(n_x + n_y)}} \lVert \frac{1}{n_x} \sum_{i=1}^{n_x} k_i^x - \frac{1}{n_y} \sum_{j=1}^{n_y} k_j^y \rVert$

that the clusters of trajectories are linked at different heights, meaning they have different dissimilarity values and hence by cutting the dendrogram from different heights, we can obtain any number of clusters, allowing us to choose how detailed we want our clusters to be. In addition, different linkage methods produce different dendrograms, so we can choose a linkage method that works the best with the given data.

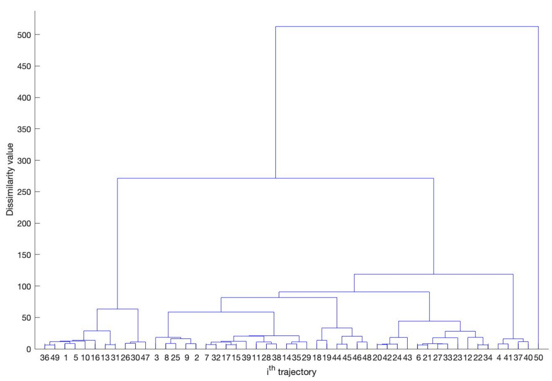


Fig. 8: An example of a dendrogram of 50 trajectories

C. Framework

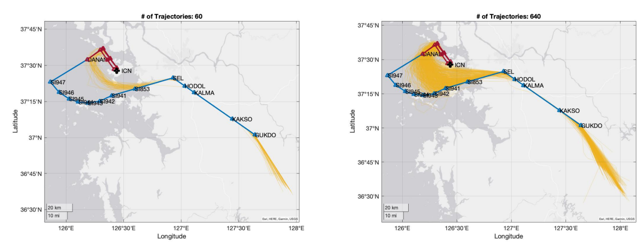
The framework shown in Figure 4 first takes a set of trajectories that follow a certain route as the input data. Then, a dissimilarity matrix is generated using DTW as mentioned in Section IV-A, which is then used to build a dendrogram using a linkage method that works the best with the given data, as explained in Section IV-B. Next, the number of clusters is selected and adjusted until all patterns are well-separated and they form the preliminary trajectory patterns. Once the algorithm-based clustering is complete, we use our domain knowledge to combine similar patterns based on the waypoints that they pass through to form the final trajectory patterns. Noted that there are some patterns that only contain less than 1% of the total number of trajectories and those patterns are ignored as noisy trajectories.

V. Results and Analysis

This section first presents the results from the proposed framework by using the ADS-B data of trajectories follow GUKDO 1N and GUKDO 1P between January and December, 2019. Then, the baseline results from using DBSCAN and k-means are presented for comparison.

A. Identified Trajectory Patterns and Analysis

By implementing the proposed framework, we identify three trajectory patterns in GUKDO 1N (Figure 10) and eight trajectory patterns in GUKDO 1P (Figure 11). In this paper, the criteria for a *good* cluster generated by the framework is based on the domain knowledge instead of a numerical measure. A cluster is considered as good if all trajectories within that cluster follow the same set of waypoints. For example, Figure 9(a) shows a good cluster because all trajectories in that cluster go from GUKDO to SEL and then take a shortcut to DANAN from SI853, while Figure 9(b) shows a bad cluster where the trajectories start taking shortcuts from multiple waypoints, SEL, SI853, SI941 and SI942. In that sense, the results from the proposed framework are considered as good. Note that some identified patterns such as Figures 10(a) and 11(a) contain flights that take shortcuts from multiple waypoints. The reason is because all those waypoints are classified, by the AIP data, as direct-to fixes where pilots are allowed to take shortcuts to the merging point. Therefor, those trajectories can be clustered as a pattern without violating the criteria. The trajectories in GUKDO 1N and GUKDO 1P are used here as an example to demonstrate the framework, but the framework can be readily implemented with other sets of flight trajectories.



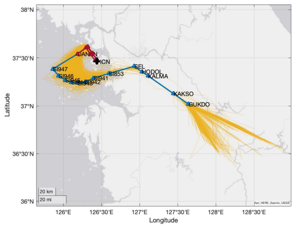
(a) A good cluster (b) A bad cluster

Fig. 9: Examples of good and bad clusters

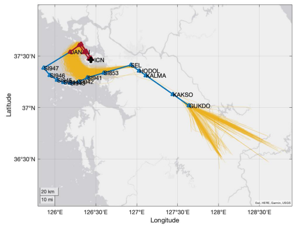
For the identified patterns, we observe that the majority of the flights do not follow the proper order of the route, except Pattern 1 in GUKDO 1N and GUKDO 1P (Figures 10(a) and 11(a)). Figures 12 and 13 show that more than 50% of the trajectories were instructed to deviate from the route by ATCs to accommodate certain traffic situations, which further emphasizes the importance of studying trajectory patterns in vectored airspace in order to develop more accurate and effective assistant tools using data-driven methods.

B. Comparison with Results and Analysis

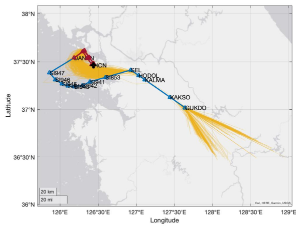
To check if the proposed framework outperforms the existing, widely-used methods, we choose the baseline algorithms to be DBSCAN and k-means (k = 3), which are most commonly used algorithms for trajectory clustering in the literature, and apply them to the data in GUKDO 1N only for illustration. The best results we can get from DBSCAN and k-means are shown in Figures 14 and 15. Note that both DBSCAN and k-means perform too poorly to be presented



(a) Pattern 1 (958 Flights)



(b) Pattern 2 (728 Flights)



(c) Pattern 3 (568 Flights)

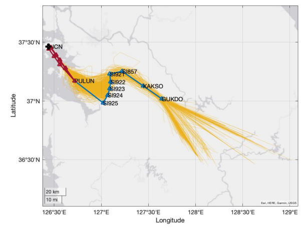
Fig. 10: Identified Trajectory Patterns in GUKDO 1N

for GUKDO 1P, which has much more complex trajectory patterns.

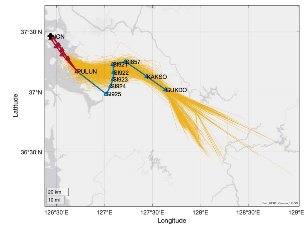
Through practice, we observe that DBSCAN tends to either group most trajectories as outliers or group most trajectories into a single cluster, while k-means tends to divide the entire datasets into k portions with each portion contains multiple trajectory patterns. Based on the criteria mentioned in Section V-A, DBSCAN and k-means do not perform well. The reason for the poor performance is possibly due to the fact that DBSCAN is a density-based algorithm that connects data within a certain range [27] without computing the dissimilarity measure between trajectories, which is important when trajectories are highly overlapped and embedded. K-means, on the other hand, requires the means of k sets of data [28]. However, since the data was recorded at different lengths, it is not possible to compute the means without resampling the data into the same length. By resampling, we loss information and so the hidden trajectory patterns cannot be discovered.

VI. Conclusion

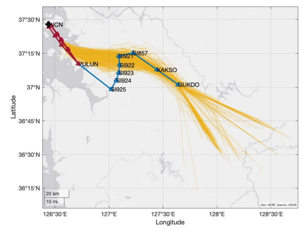
This paper presented a new clustering framework to identify trajectory patterns from the air traffic surveillance data recorded in vectored airspace. To cluster the trajectories that are represented as time-series and whose patterns are deeply embedded in the data, the proposed framework is composed of algorithm-based clustering, relied on Dynamic Time Warping (DTW) and agglomerative hierarchical clustering to obtain the preliminary patterns, and domain knowledge-based clustering where the preliminary patterns are combined based on the waypoints that the trajectories pass through to get the final trajectory patterns. The identified trajectory patterns are crucial information not only for ATCs to speed up their decision-making processes for sequencing or scheduling, but also for various data-driven methods to be able to be implemented



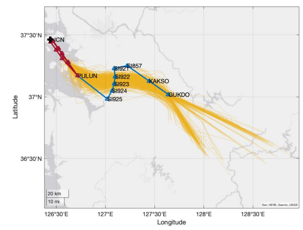
(a) Pattern 1 (697 Flights)



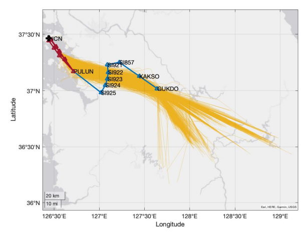
(b) Pattern 2 (919 Flights)



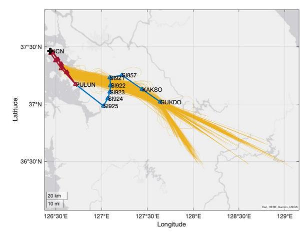
(c) Pattern 3 (591 Flights)



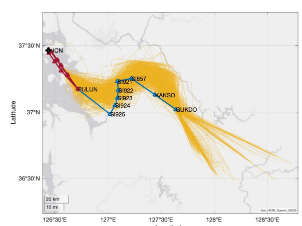
(d) Pattern 4 (499 Flights)



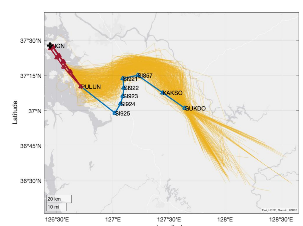
(e) Pattern 5 (1,886 Flights)



(f) Pattern 6 (956 Flights)



(g) Pattern 7 (600 Flights)



(h) Pattern 8 (466 Flights)

Fig. 11: Identified Trajectory Patterns in GUKDO 1P

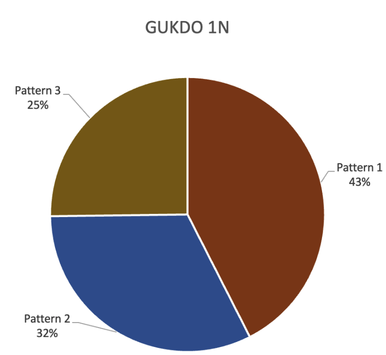


Fig. 12: Percentage of trajectory patterns in GUKDO 1N

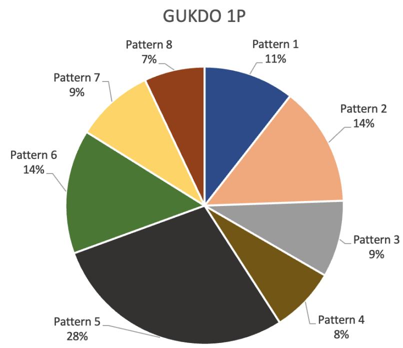
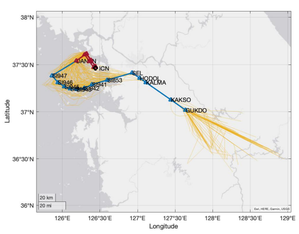
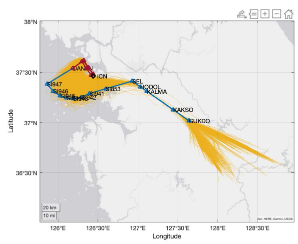


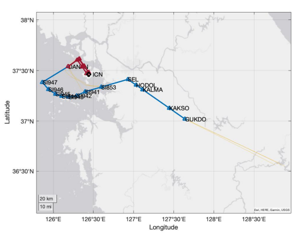
Fig. 13: Percentage of trajectory patterns in GUKDO 1P



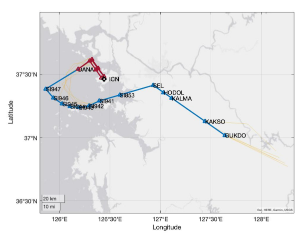
(a) Outliers (131 Flights)



(b) Pattern 1 (2,115 Flights)

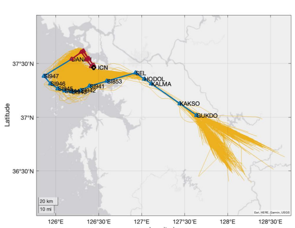


(c) Pattern 2 (3 Flights)

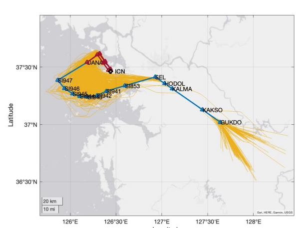


(d) Pattern 3 (5 Flights)

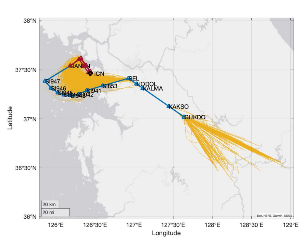
Fig. 14: DBSCAN identified trajectory patterns in GUKDO 1N



(a) Pattern 1 (576 Flights)



(b) Pattern 2 (606 Flights)



(c) Pattern 3 (1,072 Flights)

Fig. 15: k-means identified trajectory patterns in GUKDO 1N

using air traffic surveillance data recorded in vectored airspace. To further demonstrate the proposed framework, an extensive testing with more data will be done in the future. In addition, the identified trajectory patterns will be utilized for trajectory pattern classification and other applications to further validate the results.

ACKNOWLEDGMENT

This work is supported by the Korea Agency for Infrastructure Technology Advancement (KAIA) grant funded by the Ministry of Land, Infrastructure and Transport (Grant 21BDAS-B158275-02). The authors are grateful to Dr. Hak-Tae Lee at Inha University for his valuable comments and support.

REFERENCES

[1] S. Thomas and G. Athens, “Civilian and military air traffic control in the eu,” 2000.

[2] *Air Traffic Controllers*, 15th ed., Bureau of Labor Statistics, Washington, DC: USA, 2014.

[3] M. Hawley, R. Bharadwaj, and V. Venkataraman, “Real-time mitigation of loss of separation events using reinforcement learning,” in *2019 IEEE/AIAA 38th Digital Avionics Systems Conference (DASC)*, 2019, pp. 1–6.

[4] V. M. Janakiraman and D. Nielsen, “Anomaly detection in aviation data using extreme learning machines,” in *2016 International Joint Conference on Neural Networks (IJCNN)*, 2016, pp. 1993–2000.

[5] R. Deshmukh, D. Sun, K. Kim, and I. Hwang, “Temporal logic learning-based anomaly detection in metroplex terminal airspace operations,” *Transportation Research Part C: Emerging Technologies*, vol. 126, p. 103036, 2021.

[6] ——, “Reactive temporal logic-based precursor detection algorithm for terminal airspace operations,” *Journal of Air Transportation*, vol. 28, pp. 155–163, 10 2020.

[7] B. Matthews, S. Das, K. Bhaduri, K. Das, R. Martin, and N. Oza, “Discovering anomalous aviation safety events using scalable data mining algorithms,” *Journal of Aerospace Information Systems*, 10 2013.

[8] L. Li, S. Das, R. J. Hansman, R. Palacios, and A. N. Srivastava, “Analysis of flight data using clustering techniques for detecting abnormal operations,” *Journal of Aerospace Information Systems*, vol. 12, 9 2015.

[9] M. C. Rocha Murça and M. de Oliveira, “A data-driven probabilistic trajectory model for predicting and simulating terminal airspace operations,” in *2020 AIAA/IEEE 39th Digital Avionics Systems Conference (DASC)*, 2020, pp. 1–7.

[10] S. T. Barratt, M. J. Kochenderfer, and S. P. Boyd, “Learning probabilistic trajectory models of aircraft in terminal airspace from position data,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 9, pp. 3536–3545, 2019.

[11] Z. Wang, M. Liang, and D. Delahaye, “Short-term 4d trajectory prediction using machine learning methods,” in *7th SESAR Innov.*, 2017, pp. 1–9.

[12] M. C. R. Murça and R. J. Hansman, "Identification, characterization, and prediction of traffic flow patterns in multi-airport systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 5, pp. 1683–1696, 2019.

[13] K. P. Sinaga and M.-S. Yang, "Unsupervised k-means clustering algorithm," *IEEE Access*, vol. 8, pp. 80 716–80 727, 2020.

[14] B. Shi, L. Han, and H. Yan, "Adaptive clustering algorithm based on knn and density," *Pattern Recognition Letters*, vol. 104, pp. 37–44, 2018.

[15] Z. Peng, J. Zhang, X. Gui, R. Hu, and D. Sui, "A data-driven probabilistic trajectory model for predicting and simulating terminal airspace operations," in *2020 International Conference on Research in Air Transportation*, 2020.

[16] R. Elankavi, R. Kalaiprasath, and R. Udayakumar, "A fast clustering algorithm for high-dimensional data," *International Journal of Civil Engineering and Technology*, vol. 8, pp. 1220–1227, 2017.

[17] M. Enriquez, "Identifying temporally persistent flows in the terminal airspace via spectral clustering," 10 2013.

[18] F. Liu, W. Bi, W. Hao, F. Gao, and J. Tang, "An improved fuzzy trajectory clustering method for exploring urban travel patterns," *Journal of Advanced Transportation*, vol. 2021, pp. 1–13, 2021.

[19] X. Olive, L. Basora, B. Viry, and R. Alligier, "Deep trajectory clustering with autoencoders," 08 2020.

[20] S. Hong and K. Lee, "Trajectory prediction for vectored area navigation arrivals," *Journal of Aerospace Information Systems*, vol. 12, pp. 490–502, 2015.

[21] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 26, no. 1, pp. 43–49, 1978.

[22] B. A. Sabarish, R. Karthi, and T. Gireeshkumar, "Clustering of trajectory data using hierarchical approaches," in *Computational Vision and Bio Inspired Computing*, D. J. Hemanth and S. Smys, Eds. Cham: Springer International Publishing, 2018, pp. 215–226.

[23] J. H. Ward, "Hierarchical grouping to optimize and objective function," *Journal of the American Statistical Association*, vol. 58, 2013.

[24] S. C. Johnson, "Hierarchical clustering schemes," *Psychometrika*, vol. 32, no. 3, pp. 241–254, 1967.

[25] H. H. Ross, "Principles of Numerical Taxonomy," *Systematic Biology*, vol. 13, no. 1-4, pp. 106–108, 03 1964.

[26] D. Defays, "An efficient algorithm for a complete link method," *The Computer Journal*, vol. 20, no. 4, pp. 364–366, 01 1977.

[27] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, ser. KDD'96. AAAI Press, 1996, p. 226–231.

[28] S. Lloyd, "Least squares quantization in pcm," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.