A Data-Driven Framework for Modelling Complexity in Terminal Manoeuvring Area

Zhi Jun Lim, Imen Dhief, Duc-Thinh Pham, Sameer Alam
Air Traffic Management Research Institute
Nanyang Technological University
Singapore
{zhijun.lim, imen.dhief, dtpham, sameeralam}@ntu.edu.sg

Daniel Delahaye
OPTIM Lab
Ecole Nationale de l'Aviation Civile
Toulouse, France
daniel@recherche.enac.fr

***Abstract*—This paper presents an objective, data-driven framework for quantifying air traffic complexity in the Terminal Manoeuvring Area (TMA) using historical ADS-B data from Singapore TMA. The motivation for developing this framework stems from the limitations of traditional subjective measures, which are often influenced by individual perceptions and can vary significantly between air traffic controllers. Subjective measures may also fail to capture real-time operational demands, especially in complex, high-density environments such as Singapore TMA. By focusing on operational outcomes—specifically vectoring and holding patterns—the framework provides a more accurate reflection of real-time complexity. Principal Component Analysis (PCA) and k-means clustering are employed to classify complexity levels based on trajectory features such as arc lengths, curvatures, and holding durations. The results show that total arc lengths and curvatures are significant complexity factors, with extensive vectoring contributing more to TMA complexity than holding patterns. The significance of this work lies in its data-driven and objective approach to measuring air traffic complexity, offering a more accurate reflection of real-time demands compared to traditional subjective methods. Quantitative evaluations across multiple real-world scenarios validate the framework's effectiveness, showing that TMA complexity is more strongly associated with vectoring intensity and holding patterns than with flight density alone. This current framework can be extended to incorporate vertical profiles of arrival and departure flights and develop predictive models with practical, actionable lookahead times for real-time air traffic management.**

***Keywords*—Air Traffic Management, Terminal Manoeuvring Area, Complexity, Data-driven, Unsupervised Machine Learning, Clustering**

I. Introduction

The Terminal Manoeuvring Area (TMA), which serves as the critical transition between the en-route sectors and aerodrome control zone [1], is a vital and complex airspace in air traffic management (ATM). This complexity is influenced by many factors, including airways structure of Standard Terminal Arrival Routes (STARs) and Standard Instrument Departures (SIDs), traffic density, the dynamic coordination between multiple arriving and departing flights, the mix of aircraft types in the traffic and meteorological conditions [2], [3]. In terms of the definition of complexity in air traffic control (ATC) literature, Meckiff et al. [4] described it as the level of challenge a traffic situation poses to an air traffic controller (ATCO) while Mogford et al. [5] referred ATC complexity as "the effect on the controller of the airspace and the air traffic flying within it".

The concept of complexity is of paramount importance in the Air Traffic Control (ATC) domain, as it is a key driver of air traffic controllers' workload, which in turn directly influences airspace capacity [6]–[8]. Figure 1 shows the relationship between ATC complexity and controller workload. The factors contributing to ATC complexity can be broadly classified into two categories: airspace geometry, and traffic demand and distribution. ATC complexity, along with the demands of the interface, equipment, and procedures, contributes to the overall taskload of the controller. This taskload is further modulated by individual performance shaping factors (PSFs), which include personal attributes such as age, experience, and skills, as well as the cognitive strategies employed by the controller, which are shaped by professional training. As a result, different controllers may perceive different levels of workload, even when facing the same taskload. Therefore, while taskload represents the objective demands of a task, workload reflects the subjective perception of these demands during task execution. As such, this work will focus on studying ATC complexity in relation to the additional observable taskload due to increased complexity, rather than the subjective workload experienced by controllers. This approach is chosen because workload is subjective and influenced by various mediating factors which are difficult to quantify and observe directly from the air traffic data.

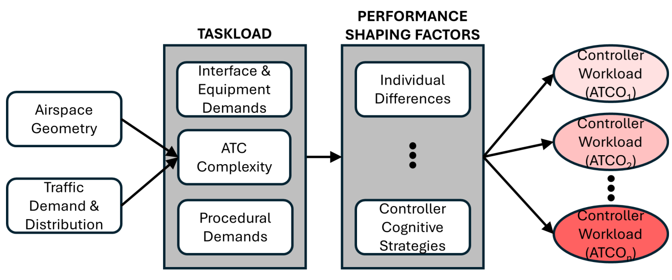


Figure 1. Factors affecting controller workload, adapted from [5], [9], [10].

A. *Related work*

Many studies have focused on complexity indicators for en-route sectors, but only a few have explored them for the TMA.

Traditionally, air traffic complexity has been assessed using subjective evaluations by controllers, such as in Laudeman et al.'s work on Dynamic Density (DD) [11]. This metric, which uses regression analysis to assign weights to factors based on subjective assessments, has been extended to include predictive capabilities [12] and additional complexity factors [7], [13]. However, these models are often sector-specific and may not generalize well to other airspace [14], [15], while subjective assessments introduce variability [16]. Furthermore, en-route models, which consider changes in heading, altitude, and speed as complexity sources, are less applicable to TMA, where such changes are expected and managed by controllers.

Recent years have seen increased research into TMA complexity models. Netjasov et al. [10] introduced a metric to assess complexity in terminal airspace, incorporating static and dynamic factors such as airspace structure, runway capacity, traffic levels, and aircraft types. Applied to London Heathrow, the metric provides valuable insights but may not fully capture real-time variations, limiting its practical application in dynamic settings. In a related development, Deng et al. [17] proposed a complexity estimation framework for RNAV terminal airspace, with components for vectoring, separation, and anomaly complexities. Although the framework aids in detecting operational anomalies, it assumes a linear relationship between components with equal weighting, similar to Netjasov's work, which may not reflect the true interactions between factors [18].

Delahaye et al. developed a complexity metric for both en-route and TMA sectors based on non-linear [19] and linear [20] dynamical systems, aiming to measure the intrinsic complexity of aircraft trajectories by aligning a vector field with observed positions and speeds. This generates a complexity map for comparing areas of varying complexity. However, to compute complexity using this metric, flight plan trajectories must be used instead of actual flown trajectories. This is because historical flown trajectories incorporate air traffic controllers' actions in resolving conflicts, which can lead to an underestimation of complexity when applying the metric. The use of flight plans, however, presents its own challenges. Flight plans are often proprietary to airlines, especially in Asia, making them difficult to acquire. Additionally, flight plan trajectories may not be suitable for real-time applications, as they do not reflect real-time variations in traffic. These real-time changes are crucial for accurately assessing the dynamic and ever-changing conditions in airspace operations. As a result, relying solely on flight plans to compute complexity may lead to an inaccurate representation of operational demands. To address these limitations, alternative methods are needed that can capture real-time variations and provide a more accurate assessment of airspace complexity.

B. Objective & Motivation

To address the gaps mentioned in the previous works, this paper proposes a framework that leverages historical Automatic Dependent Surveillance-Broadcast (ADS-B) data and airspace structure to learn and identify different levels of air traffic complexity within the TMA. Unlike flight plan data, historical flight trajectory data are readily accessible through open sources like the OpenSky Network [21] for ADS-B data. The framework is designed to be adaptable across various TMAs, providing a consistent and objective assessment of complexity.

The primary motivation for this research is to achieve an objective assessment of air traffic complexity. By utilizing a data-driven metric, the proposed framework provides clear and consistent information that reduces the cognitive load on controllers, allowing them to focus on managing traffic rather than interpreting subjective assessments. Although the current work does not develop a predictive model, the framework lays the groundwork for future research by providing a solid, objective basis for complexity assessment. This foundation will be instrumental in developing predictive methods to anticipate potential high-risk air traffic situations, enabling proactive workload distribution among controllers. Such advancements could reduce the risk of human error and ensure more efficient resource allocation. Additionally, the framework's objective complexity metric supports collaborative decision-making with adjacent sector controllers by providing a unified understanding of the airspace situation. This shared perspective facilitates better coordination and more effective decision-making. Ultimately, integrating this metric with emerging technologies, such as AI-driven decision support systems and automated traffic management tools, will ensure that air traffic management remains adaptive, efficient, and scalable as the aviation industry continues to evolve.

The remainder of this paper is organized as follows: Section II focuses on formulating the problem by establishing the relationship between vectoring, holding patterns, and TMA complexity through a data-driven and machine learning approach. Building on this foundation, Section III details the methodology employed to develop the proposed framework. Section IV then presents an experimental study, showcasing the application of the framework within a specific TMA. Finally, Section V concludes the paper by summarizing key findings and discussing potential avenues for future research aimed at further enhancing the framework and integrating it with emerging technologies.

II. Problem Formulation

Traditionally, airspace complexity has been assessed using factors such as flight density and potential vertical, horizontal, and speed interactions, as outlined in Eurocontrol's complexity metric [22]. However, these metrics focus on the sources of complexity rather than how it manifests operationally. This paper adopts a different approach by quantifying TMA complexity based on features extracted from actual flown trajectories. These features inherently capture the complexity of the airspace at specific moments. Among these, tactical strategies like vectoring [1] and holding patterns [23], commonly used to manage arrival delays, may serve as indicators of increased complexity.

In the TMA, Standard Terminal Arrival Routes (STARs) and Standard Instrument Departures (SIDs) [23] provide structured, predefined sequences of waypoints that guide arriving and departing aircraft along specific routes. These procedures, which include altitude and speed constraints, ensure orderly and efficient traffic flow. However, operational disruptions such as traffic congestion, weather conditions, or runway unavailability often necessitate deviations from these standard routes. In such cases, ATCOs may implement vectoring or holding manoeuvres as illustrated in Figure 2. Vectoring involves providing specific heading instructions to ensure sufficient aircraft separation while holding requires aircraft to fly in circular or racetrack patterns to delay their approach.

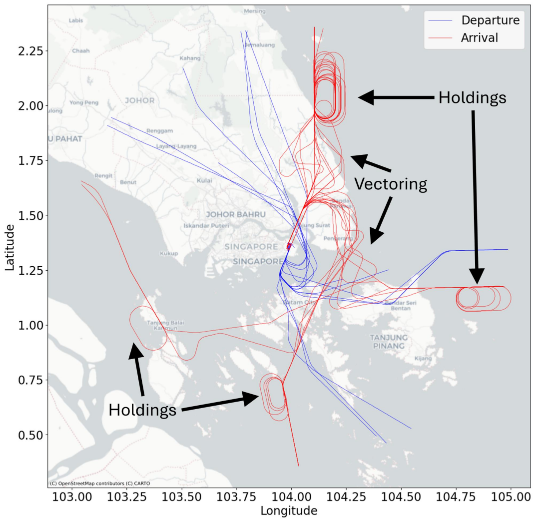


Figure 2. Historical arrival and departure flight trajectories within the Singapore Terminal Manoeuvring Area (TMA) over a one-hour period in 2024. Arrival flights (red) exhibit notable occurrences of vectoring and holding patterns, as indicated, while departure flights (blue) follow more streamlined routes. The presence of these manoeuvres highlights the operational complexity of managing arrival traffic during this timeframe.

This study focuses on arrival flights rather than departures due to the more complex and unpredictable nature of managing inbound traffic. Unlike departures, which follow relatively rigid departure slots and have more controlled sequencing, arrivals are often subject to dynamically changing conditions. Arrival flights must be managed in real-time as they transition from en route to approach, often requiring holding and vectoring to absorb delays and ensure safe separation between aircraft. These deviations may serve as indicators of TMA complexity, as they signal that standard arrival procedures are insufficient to handle the traffic volume or other constraints. In contrast, departures typically face fewer airborne delays and deviations since their timing and sequence can be controlled more easily on the ground. The need for such manoeuvres, especially for arrivals, significantly increases the task load on ATCOs. Therefore, the magnitude and duration of vectoring and holding serve as key measures of the complexity in TMA, particularly for managing arrival traffic.

The approach presented in this paper is inherently data-driven, leveraging historical trajectory data to quantify complexity based on observable outcomes, such as the frequency and duration of vectoring and holding manoeuvres. These manoeuvres, extracted from large datasets of actual flight paths, serve as real-time indicators of increased complexity. Given the volume and variability of this data, a machine learning approach is well-suited to identify patterns and relationships within the data that would be difficult to discern using traditional methods. By focusing on these tactical strategies, the study provides a clearer representation of the operational challenges involved in managing arrival flights within the TMA. This differs from traditional methods that focus on underlying sources, such as traffic density or weather. Once the TMA complexity can be quantified using this framework, the relationship between TMA complexity and these sources can be explored to develop a prediction model; however, this is beyond the scope of the current study.

To sum up, this paper introduces a framework for quantifying TMA complexity through operational outcomes, specifically vectoring and holding patterns, with a particular focus on arrival flights due to their inherently higher complexity in the TMA. This data-driven approach provides a more direct reflection of the airspace's real-time demands, offering an objective way to measure complexity based on actual observed controller interventions.

III. Methodology

The methodology presented here aims to establish an objective framework for assessing TMA complexity, leveraging historical flight data to quantify operational factors such as vectoring and holding patterns. The methodology framework, shown in Figure 3, comprises four key components: data source, data processing, feature extraction, and machine learning. Each component is essential for developing a comprehensive understanding of TMA complexity.

A. Data Source

The analysis utilizes two primary data sources: historical ADS-B data and airspace structure information.

1) **Historical ADS-B Data**: This dataset provides comprehensive information on aircraft trajectories, including position, altitude, speed, and timestamps. ADS-B data offers detailed insights into real-time flight operations, making it crucial for analyzing operational outcomes such as vectoring and holding patterns. The ADS-B data utilized in this study was sourced from Flightradar24, covering the period from May 15 to July 31, 2024.
2) **Airspace Structure**: Information on the airspace structure is obtained from Aeronautical Information Publications (AIPs). These publications detail the configuration of the TMA, including Standard Terminal Arrival Routes (STARs) and Standard Instrument Departures (SIDs) procedures. This structural information is essential for understanding the predefined routes and constraints that influence flight behaviour within the TMA.

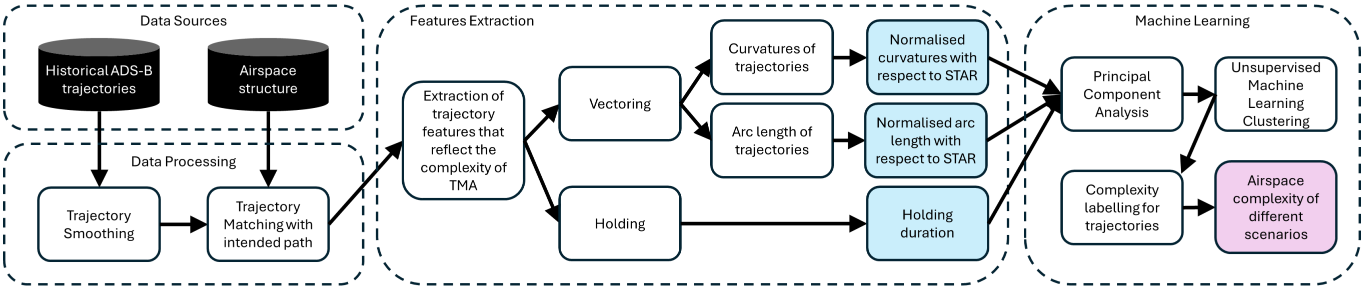


Figure 3. Methodology framework for learning complexity values in the TMA. The data-driven methodology framework consists of four components: data sources, data processing, feature extraction and machine learning, respectively.

B. *Data Processing*

The data processing phase involves several key steps to prepare the trajectory data for analysis. Beyond basic data cleaning, which ensures the dataset's integrity, the process includes trajectory smoothing and trajectory matching:

1) **Trajectory Smoothing with Bézier Curves**: To address the noisy nature of ADS-B data, trajectory smoothing is performed using Bézier curves. Bézier curves are parametric curves defined by control points that influence their shape [24]. They are particularly effective in modelling smooth, continuous paths and are commonly used in graphics and data fitting. By applying Bézier curves to the noisy trajectory data, we achieve a smoother representation of flight paths that reduces the impact of noise and irregularities, providing a more accurate depiction of the intended trajectories.
2) **Dynamic Time Warping (DTW) for Trajectory Matching**: To accurately align arrival trajectories with their corresponding Standard Terminal Arrival Routes (STARs), Dynamic Time Warping (DTW) is employed. DTW is a technique for measuring similarity between two temporal sequences that may differ in speed or timing [25]. It calculates the optimal alignment between sequences by allowing non-linear mappings of time axes. Traditional distance metrics, such as Euclidean distance, may not be suitable for this task because they assume a direct, one-to-one correspondence between points. DTW, on the other hand, can accommodate variations in timing and speed, making it ideal for matching trajectories that deviate from their standard routes. This capability allows DTW to accurately align observed trajectories with predefined STARs, even when there are deviations or variations in timing.

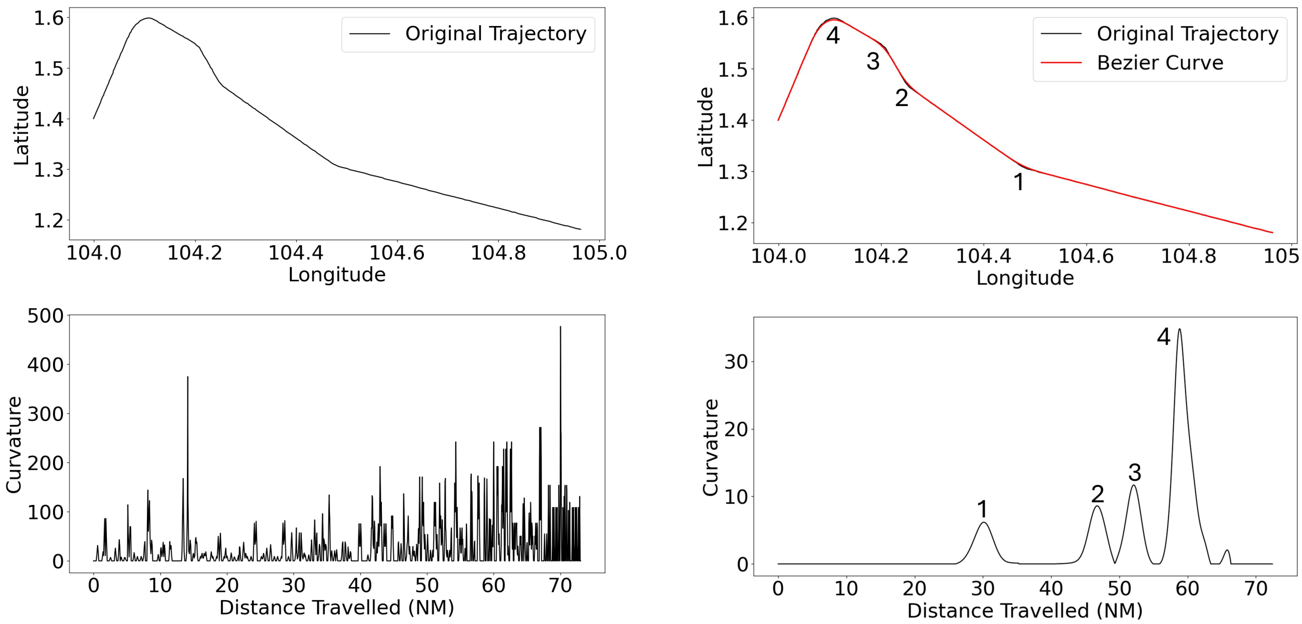
C. *Features Extraction*

In the feature extraction phase, key operational features are derived from the trajectories to quantify TMA complexity. The primary features include vectoring manoeuvres and holding patterns:

1) **Vectoring**: Vectoring is represented by two main features: curvatures and arc lengths. Higher curvatures and longer arc lengths are indicative of more significant vectoring manoeuvres. The curvature of a trajectory is calculated using the formula:

$$\kappa = \frac{\left| \frac{d^2y}{dt^2}\frac{dx}{dt} - \frac{d^2x}{dt^2}\frac{dy}{dt} \right|}{\left(\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2 \right)^{3/2}} \tag{1}$$

where $x(t)$ and $y(t)$ are the coordinates of the trajectory as functions of time t. Here, $\frac{dx}{dt}$ and $\frac{dy}{dt}$ are the first derivatives of $x(t)$ and $y(t)$ with respect to t, representing the velocity components along the x and y directions. $\frac{d^2x}{dt^2}$ and $\frac{d^2y}{dt^2}$ are the second derivatives, representing the acceleration components. The curvature κ measures how sharply the path bends at each point.
Figure 4 illustrates the curvature values along a sampled trajectory before and after applying trajectory smoothing. The raw curvature values, calculated directly from the original data, exhibit significant noise due to the inherent variability in the ADS-B measurements. In contrast, after applying Bézier curve smoothing, the curvature values become much more stable. The significant peaks in the smoothed curvature values reflect more accurately the bends and turns in the trajectory. This smoothing process effectively filters out the noise, providing a clearer and more precise representation of the trajectory's curvature.
For each trajectory, a total curvature value is obtained by summing the curvature values along the path, excluding the contributions from holding patterns. Similarly, the total arc length is calculated as the sum of the distances between consecutive points, also excluding holding patterns. This exclusion is crucial because the complexity contribution of vectoring and holding patterns is not equivalent. Holding patterns are established procedures with simpler instructions provided by air traffic controllers, making them less complex compared to vectoring. Vectoring involves continuous monitoring and dynamic instruction changes, which adds to its complexity. By isolating the effects of vectoring and holding, the analysis provides a more accurate measure of the trajectory's complexity related to vectoring and holding, respectively. The features are then normalized based on the total curvature and arc length of the as-



(a) (b)

Figure 4. (a) Original flight trajectory along with the corresponding curvature values at each point of the trajectory. (b) Curve fitting using a Bézier curve, showing the corresponding curvature values along the fitted Bézier curve. Notably, the significant peaks in the curvature values align with the actual turns in the original trajectory, showing the effectiveness of Bézier curve fitting in smoothing the trajectory.

sociated STAR to ensure comparability across different STARs. This normalization process makes the curvature and arc length values independent of the specific STAR, allowing for a standardized measure of vectoring across different routes.

2) **Holding Patterns**: Holding patterns are characterized by the duration of holding manoeuvres. This feature is extracted directly from the trajectory data by identifying segments where the aircraft is flying in holding patterns and calculating the total holding duration. The method used to detect holding patterns is based on the approach outlined in [26], which accurately identifies the holding patterns from the flight trajectory data.

D. Machine Learning

The machine learning phase consists of four key components: Principal Component Analysis (PCA), unsupervised machine learning, complexity labeling for trajectories, and the computation of TMA complexity for different scenarios.

1) **Principal Component Analysis**: PCA is used to reduce the dimensionality of the feature space while preserving the most significant variance in the data [27]. This aids the clustering process by simplifying the data while maintaining relevant trajectory information. While the dataset consists of only three features, curvature, arc length, and holding duration, PCA is still valuable for improving data quality. It helps to remove potential correlations between features, ensuring the clustering algorithm works with uncorrelated components. Additionally, PCA enhances data interpretation by identifying the principal components that capture the most variance, and it reduces noise, allowing the algorithm to focus on meaningful patterns without being influenced by minor errors or variations.

2) **Unsupervised Machine Learning Clustering**: Machine learning is adopted in this framework for uncovering hidden patterns in complex datasets without explicit labels. In the absence of ground truth for TMA complexity, unsupervised clustering algorithms group trajectories by feature similarities, offering valuable insights into different complexity levels.
Clustering algorithms are applied to the PCA-transformed components rather than the raw data to leverage the reduced dimensionality while preserving essential variance. Various clustering algorithms, including k-means, agglomerative hierarchical clustering, and Gaussian Mixture Model (GMM), are evaluated for performance. Given the absence of ground truth labels for TMA complexity, unsupervised machine learning clustering is employed to identify patterns and group trajectories based on their features without predefined categories. To assess the quality of the clusters and determine the optimal number of clusters, metrics such as the Silhouette Score and Davies-Bouldin Index are used.
The **Silhouette Score** evaluates clustering quality by measuring how similar each data point is to its own cluster compared to other clusters. For a data point i, the Silhouette Score $S(i)$ is calculated as:

$$S(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))} \tag{2}$$

where $a(i)$ is the average distance from i to all other points within the same cluster, and $b(i)$ is the minimum average distance from i to all points in the nearest neighboring cluster. The Silhouette Score ranges from -1 to +1, with higher values indicating better clustering. The **Davies-Bouldin Index** assesses cluster separation and intra-cluster cohesion. It is calculated as:

$$DB = \frac{1}{k}\sum_{i=1}^{k}\max_{i\neq j}\left(\frac{s(i)+s(j)}{d(i,j)}\right) \tag{3}$$

where $s(i)$ is the average distance between all points in cluster i and the centroid of cluster i, and $d(i,j)$ is the distance between the centroids of clusters i and j. Lower values of the Davies-Bouldin Index indicate better clustering quality.

A **Combined Score** is used to optimize clustering performance:

$$\text{Combined Score} = S - DB \tag{4}$$

where S is the average Silhouette Score, and DB is the Davies-Bouldin Index. This combined score balances the need for well-clustered data points (high Silhouette Score) with the need for well-separated clusters (low Davies-Bouldin Index). Maximizing this combined score helps identify the optimal number of clusters, reflecting the best representation of the data's underlying structure.

3) **Complexity Labeling for Trajectories**: Once the clusters are determined, each cluster is analyzed based on the features that contribute to trajectory complexity, such as curvature, arc length, and holding duration. The clusters are then ranked in terms of complexity, from lowest to highest, based on these feature values. Clusters with lower rankings correspond to trajectories that exhibit simpler patterns, while higher-ranked clusters indicate more complex trajectories. Thus, the complexity of each trajectory is defined by the rank of its associated cluster, providing a straightforward way to interpret trajectory complexity.
4) **TMA complexity of different scenarios**: For a scenario with n flights, the overall complexity is calculated as a weighted sum of the complexities of all n trajectories. Each cluster is assigned a weight based on its rank. Let there be k clusters, and the weights $\{w_1, w_2, \ldots, w_k\}$ correspond to the ranks of the clusters. The complexity C of a scenario is given by the formula:

$$C = \sum_{i=1}^{n} w_{\text{cluster}(i)} \tag{5}$$

where n is the total number of flights in the scenario, cluster(i) refers to the cluster of i-th trajectory, and $w_{\text{cluster}(i)}$ is the weight assigned to the cluster based on its rank, selected from the list $\{w_1, w_2, \ldots, w_k\}$.

IV. Experimental Study

The experimental study aims to validate the proposed framework's ability to classify and quantify TMA complexity using a data-driven approach. The study focuses on the Singapore TMA and evaluates the model using ADS-B data spanning from May 15 to July 31, 2024.

A. Principal Component Analysis

The PCA analysis was conducted to reduce the dimensionality while preserving significant variance. Principal Component 1 (PC1) explains 57% of the variance and is mainly influenced by total arc lengths (loading of 0.70) and total curvatures (loading of 0.69), highlighting their role in the variance. In contrast, holding duration has a lower loading of 0.19, indicating a lesser impact on PC1. On the other hand, Principal Component 2 (PC2) accounts for 33% of the variance and is primarily driven by holding duration (loading of 0.98), with minimal contributions from total arc lengths (-0.10) and total curvatures (-0.17). The scatter plot of PC1 versus PC2 (Figure 5) visually represents the distribution of the data across these two principal components.

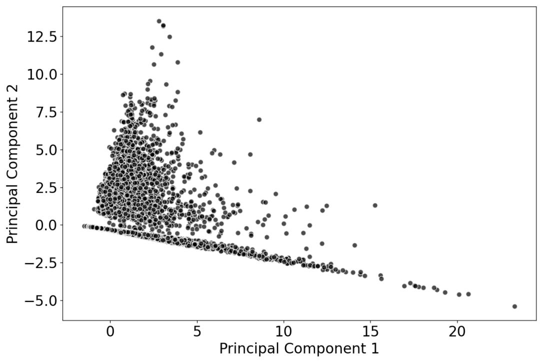


Figure 5. Scatter plot of Principal Component 1 and Principal Component 2 from the Principal Component Analysis (PCA) of three normalized variables: arc lengths, curvatures, and holding duration.

B. Unsupervised Machine Learning Clustering

The performance of the three clustering algorithms for different numbers of clusters was evaluated using the combined score as illustrated in Figure 6. Among the evaluated algorithms, k-means achieved the highest combined score at $k = 4$, indicating the optimal balance of cluster quality. Consequently, k-means with $k = 4$ was selected as the preferred clustering configuration. Figure 7 displays the clustering results of the data using the selected k-means algorithm.

C. Complexity Labeling for Trajectories

The ranking of different clusters is explained based on their complexity characteristics. The clusters are ranked from 1 to 4, with 1 indicating the lowest complexity and 4 indicating the highest complexity. These rankings reflect the relative complexity of each cluster, with green representing the simplest trajectories and purple representing the most complex.

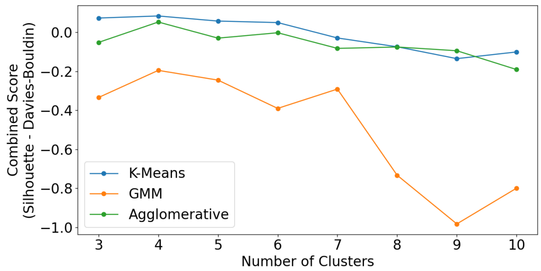


Figure 6. Combined scores for different numbers of clusters using K-Means, Gaussian Mixture Models (GMM) and Agglomerative clustering algorithms. The combined score is calculated as the silhouette score minus the Davies-Bouldin index. The highest combined score is obtained using the K-Means clustering algorithm at k=4, indicating the optimal number of clusters.

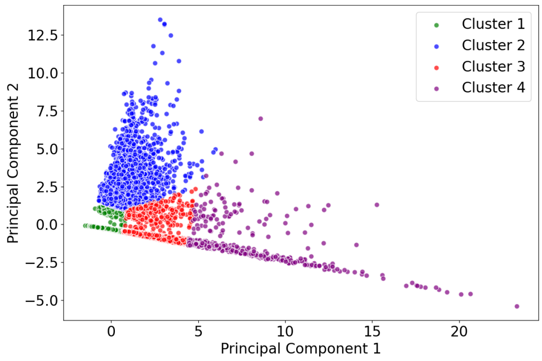


Figure 7. K-means clustering results with k=4 for the principal components.

1) Green Cluster: Represents the lowest complexity. This cluster has the shortest arc length and the lowest curvature, with minimal holding duration, making this the least complex.
2) Blue Cluster: Indicates moderate complexity. It features slightly higher curvature, arc length, and longer holding duration than the green cluster. The increased holding duration accounts for its elevated complexity.
3) Red Cluster: Displays higher complexity, with greater curvature and arc length compared to the blue cluster, though it has a shorter holding duration. The significant vectoring involved raises its complexity rank.
4) Purple Cluster: The highest complexity, marked by the longest arc length and greatest curvature, despite a short holding duration. The extensive vectoring in this cluster justifies its top complexity ranking.

D. TMA complexity of different scenario

In the analysis of TMA complexity, the overall complexity of different scenarios is computed as the weighted sum of the complexities of all trajectories, as previously defined in Equation 4. This approach aggregates the complexity of individual trajectories based on their cluster assignment and the corresponding cluster weights.

The TMA complexity of every 1-hour time periods from 15 May to 31 July 2024 are computed and the violin-plot distribution is shown in the Figure 8. The plot visualizes the spread and distribution of the computed total complexity values. The density of the data is represented by the width of the plot at different points along the complexity scale. From the violin plot, it can be observed that the majority of the complexity values lie between 10 and 40, with a denser concentration around the 20-30 range, suggesting that most time periods during the observed window experienced TMA complexity within this range. The distribution also shows some asymmetry, with a longer tail extending towards higher complexity values, indicating that while higher complexity periods (above 60) are less frequent, they do occur.

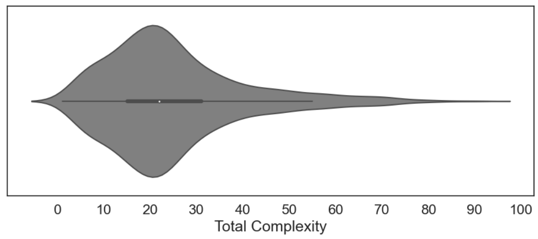


Figure 8. Distribution of total complexities within the airspace, based on 1-hour time periods from 15 May to 31 July 2024, shown using a violin plot.

Figure 9 illustrates the hourly trends in total TMA complexity across different days. The plot shows that complexity is lowest between midnight and early morning (00:00 to 04:00), with a gradual rise beginning around 05:00, peaking between 06:00 and 07:00. The complexity exhibits two distinct peaks in the afternoon: one around 14:00 and another, larger peak between 17:00 and 18:00. Disparities in the peak complexity values are observed across different days. After 20:00, complexity decreases and remains low into the night.

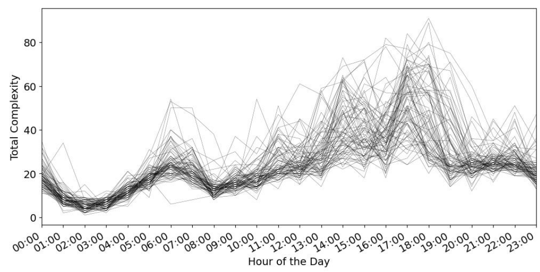
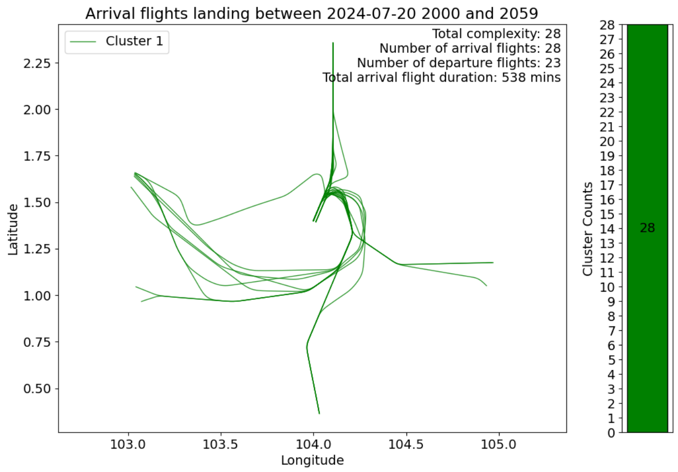


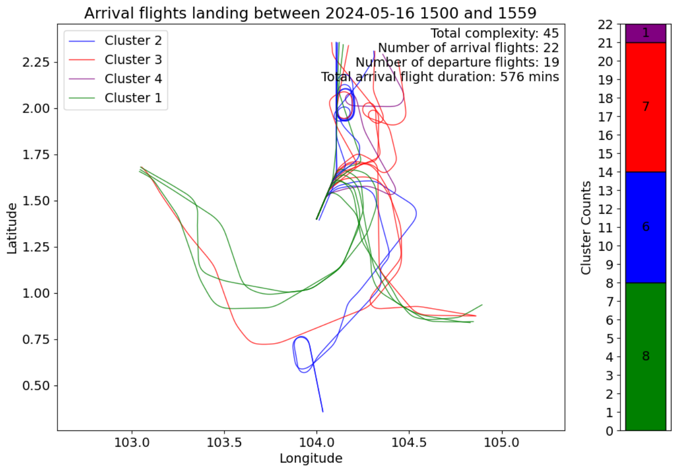
Figure 9. Hourly trends in total TMA complexity across different days, showing variations in complexity levels throughout the day.

To validate the effectiveness of the framework, several scenarios have been plotted in Figure 10 to visually demonstrate its capability to accurately label and differentiate complexity levels across various situations. These visualizations illustrate how the framework assigns and scales complexity values, providing a clear picture of how different scenarios are represented in terms of TMA complexity.

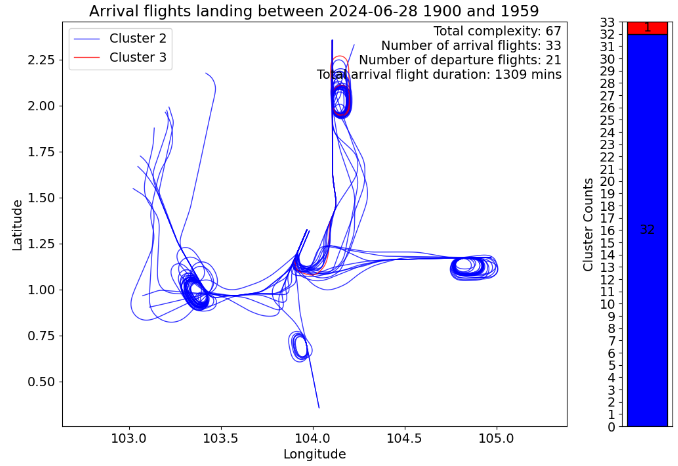
Figure 10a shows a scenario from a 1-hour period with 28 arrival flights in the TMA. All trajectories in this period are assigned to cluster 1, indicating minimal vectoring and holding. This scenario represents the highest number of arrival flights where all trajectories are categorized into cluster 1 across the dataset. Thus, the presence of significant vectoring



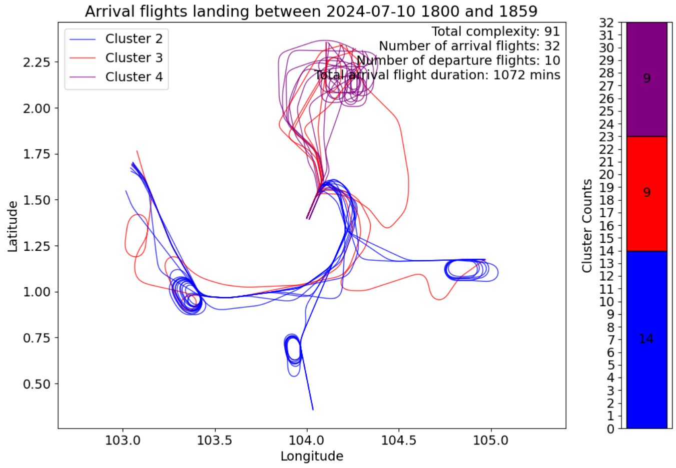
(a) This scenario shows the highest number of arrival flights where all trajectories fall into cluster 1.



(b) This scenario has flight trajectories distributed across all four complexity clusters.



(c) This scenario presents the highest number of arrival flights within a single one-hour period in the dataset.



(d) This scenario represents the highest total complexity recorded across the entire dataset.

Figure 10. Arrival trajectories are colour-coded based on their assigned clusters, with their corresponding total complexities compared across various one-hour periods.

or holding is expected when the number of arrival flights exceeds 28. In contrast, Figure 10b shows a scenario with fewer arrival and departure flights but a higher complexity level of 45, due to increased vectoring and holding. This illustrates that flight density alone does not fully reflect TMA complexity, which may be influenced by other operational factors.

Figure 10c and Figure 10d illustrate TMA complexity during periods when the number of arrival flights is near its maximum for a 1-hour interval. Despite having fewer departure flights, Figure 10d exhibits significantly higher TMA complexity compared to Figure 10c. This is because Figure 10d involves both extensive holding patterns and substantial vectoring, whereas Figure 10c primarily features holding patterns. This observation validates the framework's design, as it demonstrates that vectoring, which is more challenging to manage especially when flight density is high, contributes to higher complexity than holding patterns.

For this dataset, the complexity model produces output values ranging from 1 to 91, which may not be immediately practical for decision-making by air traffic controllers. To make these values more actionable, end-users such as shift supervisors or flow coordinators within the TMA can define thresholds to classify complexity into low, medium, and high levels. Based on the results in Figure 10a, where up to 28 arrival flights occur without significant vectoring or holding, complexity scores below 30 could be categorized as low. Scores between 30 and 60 could represent medium complexity, while those above 61 may be classified as high. This creates a clearer, more intuitive scale tailored to operational needs.

V. Conclusion and Future Work

This study introduces a framework for quantifying air traffic complexity in the TMA using historical ADS-B data from Singapore Changi Airport. The framework focuses on quantifying TMA complexity through operational outcomes, specifically vectoring and holding patterns, with a particular emphasis on arrival flights due to their inherently higher complexity in the

TMA. This data-driven approach offers a direct reflection of the airspace's real-time demands, providing an objective way to measure complexity based on actual observed controller interventions. The framework leverages Principal Component Analysis (PCA) and k-means clustering to identify and classify complexity levels based on trajectory features such as arc lengths, curvatures, and holding durations.

The PCA results highlight that total arc lengths and curvatures are primary contributors to complexity, while holding duration has a lesser impact. K-means clustering with $k = 4$ effectively categorized trajectories into complexity clusters, from low to high. Experimental results validated the framework across multiple real-world scenarios. Comparing time periods with varying arrival flight numbers, we demonstrated that complexity is not solely a function of flight density but is strongly correlated with the degree of vectoring and holding required. Notably, scenarios with extensive vectoring were significantly more complex than those dominated by holding patterns, even with similar flight numbers.

This framework produces a scalable complexity model that generates output values between 1 and 91 for this dataset, offering a comprehensive range for complexity assessment. To enhance practicality for air traffic controllers and operational supervisors, a flexible threshold system is recommended for classifying complexity into low, medium, and high categories based on operational needs. This approach ensures that the complexity model is both actionable and adaptable to varying operational scenarios.

The significance of this work lies in its data-driven, objective approach to measuring air traffic complexity, providing a more accurate reflection of real-time demands and laying the groundwork for future prediction models. Future work will expand the framework to include operational outcomes in the vertical profiles of arrival and departure flights, addressing possible interactions such as levelling-off instructions. Additionally, validating complexity values through human-in-the-loop studies and developing a predictive model with practical and actionable lookahead time will further enhance the model's applicability and effectiveness in real-time air traffic management.

Acknowledgment

This research is supported by the National Research Foundation, Singapore, and the Civil Aviation Authority of Singapore, under the Aviation Transformation Programme. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore and the Civil Aviation Authority of Singapore. The authors thank RSAF Senior Air Traffic Controller (Retd.) Luke Neubronner for his feedback and inputs.

References

[1] International Civil Aviation Organization (ICAO), *Procedures for Air Navigation Services – Aircraft Traffic Management (Doc 4444)*, sixteenth ed., 2016.

[2] R. H. Mogford *et al.*, "Application of research techniques for documenting cognitive processes in air traffic control: Sector complexity and decision making," Federal Aviation Administration (FAA), Tech. Rep. DOT/FAA/CT-TN94/3, 1994.

[3] T. Jurinić *et al.*, "Defining terminal airspace air traffic complexity indicators based on air traffic controller tasks," *Aerospace*, vol. 11, no. 5, p. 367, 2024.

[4] C. Meckiff *et al.*, "The tactical load smoother for multi-sector planning," in *Proceedings of the 2nd usa/europe air traffic management research and development seminar*, 1998, pp. 1–12.

[5] R. H. Mogford *et al.*, "The complexity construct in air traffic control: A review and synthesis of the literature," Federal Aviation Administration (FAA), Tech. Rep. DOT/FAA/CT-TN95/22, 1995.

[6] R. Christien *et al.*, "Air traffic complexity indicators & atc sectors classification," in *Proceedings. The 21st Digital Avionics Systems Conference*, vol. 1. IEEE, 2002, pp. 2D3–2D3.

[7] A. Majumdar and W. Y. Ochieng, "Factors affecting air traffic controller workload: Multivariate analysis based on simulation modeling of controller workload," *Transportation Research Record*, vol. 1788, no. 1, pp. 58–69, 2002.

[8] J. Djokic *et al.*, "Air traffic control complexity as workload driver," *Transportation research part C: emerging technologies*, vol. 18, no. 6, pp. 930–936, 2010.

[9] B. Hilburn, "Cognitive complexity in air traffic control: A literature review," *EEC note*, vol. 4, no. 04, pp. 1–80, 2004.

[10] F. Netjasov *et al.*, "Developing a generic metric of terminal airspace traffic complexity," *Transportmetrica*, vol. 7, no. 5, pp. 369–394, 2011.

[11] I. V. Laudeman *et al.*, "Dynamic density: An air traffic management metric," National Aeronautics and Space Administration (NASA), Tech. Rep. NASA/TM-1998-112226, 1998.

[12] B. Sridhar *et al.*, "Airspace complexity and its application in air traffic management," in *Proceedings of the 2nd usa/europe air traffic management research and development seminar*, 1998, pp. 1–6.

[13] P. Kopardekar and S. Magyarits, "Dynamic density: measuring and predicting sector complexity [atc]," in *Proceedings. The 21st Digital Avionics Systems Conference*, vol. 1. IEEE, 2002, pp. 2C4–2C4.

[14] A. Koros *et al.*, "Complexity in air traffic control towers: A field study part 1—complexity factors," Federal Aviation Administration (FAA), Tech. Rep. DOT/FAA/CT-TN03/14, 2003.

[15] B. Kirwan and R. Kennedy, "Investigating complexity factors in uk air traffic management," *Engineering Psychology and Cognitive Ergonomics*, pp. 189–195, 2017.

[16] P. Andraši *et al.*, "Subjective air traffic complexity estimation using artificial neural networks," *Promet-Traffic&Transportation*, vol. 31, no. 4, pp. 377–386, 2019.

[17] D. Chuhao *et al.*, "Area navigation terminal airspace complexity estimation for arrivals," in *Proceedings of the 15th usa/europe air traffic management research and development seminar*, 2023, pp. 1–11.

[18] S. Athènes *et al.*, "Atc complexity and controller workload: Trying to bridge the gap," in *Proceedings of the International Conference on HCI in Aeronautics*, vol. 1. AAAI Press Cambridge, MA, 2002.

[19] D. Delahaye *et al.*, "Air traffic complexity map based on non-linear dynamical systems," *Air traffic control quarterly*, vol. 12, no. 4, pp. 367–388, 2004.

[20] ——, "Air traffic complexity map based on linear dynamical systems," *Aerospace*, vol. 9, no. 5, p. 230, 2022.

[21] M. Schäfer *et al.*, "Bringing up opensky: A large-scale ads-b sensor network for research," in *IPSN-14 Proceedings of the 13th International Symposium on Information Processing in Sensor Networks*. IEEE, 2014, pp. 83–94. [Online]. Available: https://opensky-network.org

[22] EUROCONTROL, "Complexity metrics for ansp benchmarking analysis," EUROCONTROL, Tech. Rep., 2006.

[23] International Civil Aviation Organization (ICAO), *Procedures for Air Navigation Services – Aircraft Operations (PANS – OPS), Doc 8168, Volume 1 - Flight Procedures*, sixth edition ed., 2018.

[24] G. Farin, "Algorithms for rational bézier curves," *Computer-aided design*, vol. 15, no. 2, pp. 73–77, 1983.

[25] M. Müller, "Dynamic time warping," *Information retrieval for music and motion*, pp. 69–84, 2007.

[26] I. Dhief *et al.*, "Speed control strategies for e-aman using holding detection-delay prediction model," in *Proceedings of 10th SESAR Innovation Days*, 2021.

[27] S. Wold *et al.*, "Principal component analysis," *Chemometrics and intelligent laboratory systems*, vol. 2, no. 1-3, pp. 37–52, 1987.