

Article

Air Traffic Trajectory Operation Mode Mining Based on Clustering

Xinmin Tang *, Yusheng Liu and Kefan Chen

Civil Aviation College, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China; lys1998612@nuaa.edu.cn (Y.L.); ckf@nuaa.edu.cn (K.C.)

* Correspondence: tangxinmin@nuaa.edu.cn

Abstract: When processing track sequences, it is time-consuming and difficult to separate clusters with substantial density variations to deal with the problem of classic clustering methods mining common flight patterns in airspace. To overcome these issues, this research proposes a clustering-based technique for mining air traffic trajectory operation patterns. The track data are first decoded and rebuilt using a motion-based track training approach; next, a compression based on a deep autoencoder (OFAE) is provided, allowing the model to deal with the high-dimensional trajectory vector containing derived information. The compressed trajectory data are made as compact and dense as feasible using the L21 norm constraint, which reduces the operation time and improves the performance of the clustering process. The compressed trajectory is then analyzed using a fast-clustering algorithm based on density peaks (DPCA). To save time, a more refined distance measurement technique is added into the model in order to achieve the usual aircraft operation mode in the terminal area. The accuracy of trajectory prediction can be improved by using the generated unitized and high-class similarity trajectory data.

Keywords: air transportation; autoencoder; density peak clustering; operation mode mining; terminal area



Citation: Tang, X.; Liu, Y.; Chen, K. Air Traffic Trajectory Operation Mode Mining Based on Clustering. *Appl. Sci.* **2022**, *12*, 5944. <https://doi.org/10.3390/app12125944>

Academic Editor: Carla Raffaelli

Received: 8 May 2022

Accepted: 8 June 2022

Published: 10 June 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Aircraft usually have fixed flight patterns in the terminal area, which is based on flight procedures, fixed patterns, and other factors. Therefore, the mining of typical flight patterns in the airspace uses unsupervised algorithms. Clustering has long-been a hot research topic [1–5]. Gariel [6] transcoded ADS-B data and extracted trajectory feature points for DBSCAN (Density-Based Spatial Clustering of Applications with Noise) clustering analysis, and Wang [7] used a hierarchical clustering algorithm to identify the sample categories of trajectories and the abnormal trajectories. Wang [8] analyzed the characteristics of the track data, filtered the abnormal data, and extracted the track feature points by using the fuzzy clustering method, while Zeng et al. [9] demonstrated the advantages of mining traffic flow patterns in terminal areas using a deep autoencoder and a Gaussian mixture model with intuitive display and separability, and shortened clustering time and improved clustering efficiency. Bolic et al. [10] obtained through data mining analysis that the flight trajectory largely depends on the airline's strategy, flight type, and operating costs. Corrado [11] proposed an air traffic flow identification method based on HDBSCAN clustering with a weighted Euclidean distance function to more accurately identify arrival trajectories. The anomaly detection aspect makes a new distinction between spatial and energy anomalies in ADS-B data and provides key insights into the relationship between these two types of anomalies.

However, when the traditional distance measurement DTW (dynamic time warping) or the DBSCAN [12,13] model are used in the above literature, it takes a long time, the cost is high, and it is difficult to distinguish clusters with large density differences. Some tactical traffic flows with low trajectory density due to control or regional reasons are

often classified as a high-density cluster of nearest neighbors by DBSCAN to reduce the clustering effect. Therefore, here we explore the compression method of high-dimensional trajectory features and the appropriate clustering algorithm.

The traffic flow in the terminal area is highly nonlinear, and the dimensionality reduction process of traditional principal component analysis (PCA) is essentially a simple linear transformation of the original features, which cannot capture the nonlinear dependence between the data. In dealing with small samples and nonlinearity in face recognition, Fu [14] combined L21-norm and kernel functions and introduced L21-norm penalty terms to effectively improve the recognition rate. Liu [15] proposed a new orthogonal sparse linear discriminant analysis algorithm, which first constructs the K th nearest neighbor graph to retain the local discriminant information of the sampled points. Then, the L21-norm constraint on the projection matrix is used as a loss function to enhance the robustness to anomalous data points. Wen [16] proposed a robust sparse LDA algorithm to obtain the feature with the maximum discriminant through the L21-norm, combine the orthogonal and sparse matrices to maximize the retention of the original data representation, and enhance the robustness to noise.

Therefore, the autoencoder (AE) is used to learn the nonlinear track features, but if the unconstrained conventional stack AE is directly used to compress the data, the result is not ideal. The number of nodes and hidden layers needs frequent parameter adjustment, which is time-consuming and cannot always be guaranteed to find effective parameter groups. Therefore, this paper uses an L21-norm constrained autoencoder to learn the low-dimensional manifold of high-dimensional track data, compress the track data, and limit the model overfitting so that the clustering algorithm can use low-dimensional data for operation. Then, a fast clustering algorithm based on density peaks (DPCA) is used to mine the typical operation mode of aircraft in the terminal area through cluster analysis and obtain the track data with a unit and high similarity between classes.

2. Data Analysis and Processing

2.1. Data Analysis

The trajectory position information of aircraft presents a high degree of autocorrelation in time. To obtain the spatio-temporal characteristics and movement patterns of the trajectory in the terminal area, it is necessary to explore and analyze a large amount of historical operation data.

To facilitate the uniform management of data sending and receiving in the A-SMGCS system, the European Organization for the Safety of Air Navigation has developed a corresponding data interaction format—Asterix, which allows both communicating parties to codify and decode aircraft trajectory data messages according to this format. The data used in this paper is in Asterix category-062 format, hereinafter referred to as cat-062.

The fields contained in the cat-062 message are shown in Table 1.

Table 1. Overview of the cat-062 text field.

| TFRN | Data Item | Information | Length |
|------|-----------|---------------------------------------|--------|
| 1 | 1062/010 | Data Source Identifier | 2 |
| 2 | - | Spare | - |
| 3 | 1062/015 | Service Identification | 1 |
| 4 | 1062/070 | Time of Track Information | 3 |
| 5 | 1062/105 | Calculated Track Position (WGS-84) | 8 |
| 6 | 1062/100 | Calculated Track Position (Cartesian) | 6 |
| 7 | 1062/185 | Calculated Track Velocity (Cartesian) | 4 |
| FX | - | Field Extension Indicator | - |
| 8 | 1062/210 | Calculated Acceleration (Cartesian) | 2 |
| 9 | 1062/060 | Track Mode 3/A Code | 2 |
| 10 | 1062/245 | Target Identification | 7 |
| 11 | 1062/380 | Aircraft-Derived Data | 1+ |

Table 1. Cont.

| TFRN | Data Item | Information | Length |
|------|-----------|---------------------------|--------|
| 12 | 1062/040 | Track Number | 2 |
| 13 | 1062/080 | Track Status | 1+ |
| 14 | 1062/290 | System Track Update Ages | 1+ |
| FX | - | Field Extension Indicator | - |

In this paper, the real trajectory of the terminal area of Guangzhou was selected, the data were collected from the Eurocat-X system, and the trajectory information was encoded in the cat-062 specification and packaged into a pcap file for communication and transmission within the network. A pcap file is a common datagram storage format consisting of a pcap header and timestamped packets, each consisting of a packet header and a packet data. The file format is shown in Figure 1.

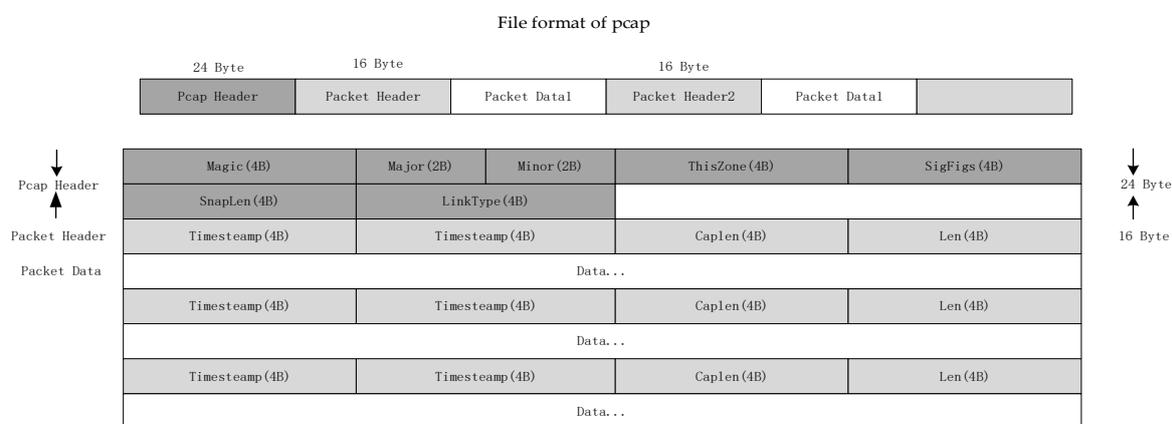


Figure 1. File format of pcap.

The captured communication packets’ content information includes: capture time, source IP, destination IP, port number, etc. Since some information cannot be obtained from the message format, it is necessary to truncate the file header of pcap, use Python to read the version number of the fixed position of the Asterix data stream header, and directly obtain the target cat number to complete the analysis of the Asterix message. The parsed data is the information of all aircraft within the radar receiving range at each time point. It is necessary to re-aggregate the data in the sampling period according to the aircraft. The flight number is obtained by reading field 390 as the unique identification code of the aircraft, and then the aggregated data are saved in CSV in chronological order. The aggregated single original track is shown in Table 2.

It is observed that time information duplication and some key columns are missing from the data, which are marked with NA. Data loss will occur normally after the flight has been running for a period of time, and its effectiveness will be maintained for a long time. Therefore, while removing the repeated time, we extracted the trajectory with complete key information, that is, removed the segmented NA data and retained the occasional NA data in the effective date, because these accidental deficiencies can be compensated for by the trajectory correction algorithm.

After the preliminary cleaning is completed, the trajectory can be drawn to observe the general situation of the data. The large missing or non-conforming trajectory in the dataset that is difficult to repair by the code as a whole can be manually screened, and then the valuable trajectory can be smoothed and denoised.

Table 2. Trajectory data.

| | | | | | | | | | | | |
|----|-----------|----------|----------|----|----|----|----|---|------|----|----|
| 1 | 25,751.42 | 23.38082 | 113.3027 | NA | 0 | −5 | NA | 0 | B763 | NA | NA |
| 2 | 25,751.42 | 23.38082 | 113.3027 | NA | 0 | −5 | NA | 0 | B763 | NA | NA |
| 3 | 25,751.73 | 23.3808 | 113.3027 | NA | 0 | −5 | NA | 0 | B763 | NA | NA |
| 4 | 25,751.73 | 23.3808 | 113.3027 | NA | 0 | −5 | NA | 0 | B763 | NA | NA |
| 5 | 25,751.42 | 23.30876 | 113.3027 | NA | 0 | −5 | NA | 0 | B763 | NA | NA |
| 6 | 25,751.42 | 23.30876 | 113.3027 | NA | 0 | −5 | NA | 0 | B763 | NA | NA |
| 7 | 25,751.73 | 23.30875 | 113.3027 | NA | −1 | −5 | NA | 0 | B763 | NA | NA |
| 8 | 25,751.73 | 23.30875 | 113.3027 | NA | −1 | −5 | NA | 0 | B763 | NA | NA |
| 9 | 25,751.42 | 23.30871 | 113.3027 | NA | −1 | −5 | NA | 0 | B763 | NA | NA |
| 10 | 25,751.42 | 23.30871 | 113.3027 | NA | −1 | −5 | NA | 0 | B763 | NA | NA |
| 11 | 25,751.73 | 23.3087 | 113.3026 | NA | −1 | −5 | NA | 0 | B763 | NA | NA |
| 12 | 25,751.73 | 23.3087 | 113.3026 | NA | −1 | −5 | NA | 0 | B763 | NA | NA |
| 13 | 25,751.42 | 23.30866 | 113.3026 | NA | −1 | −6 | NA | 0 | B763 | NA | NA |
| 14 | 25,751.42 | 23.30866 | 113.3026 | NA | −1 | −6 | NA | 0 | B763 | NA | NA |
| 15 | 25,751.73 | 23.30866 | 113.3026 | NA | −1 | −5 | NA | 0 | B763 | NA | NA |
| 16 | 25,751.73 | 23.30866 | 113.3026 | NA | −1 | −5 | NA | 0 | B763 | NA | NA |

Time, latitude, longitude, altitude, x-direction speed, y-direction speed, climb rate, takeoff and landing status, heading, and wind direction are the columns from left to right. NA is used if the fields do not exist or are missing.

The approximate shape of the invalid track is shown in Figure 2. There are large segments of missing and jumping that cannot be repaired by the reconstruction algorithm, so they are directly eliminated.

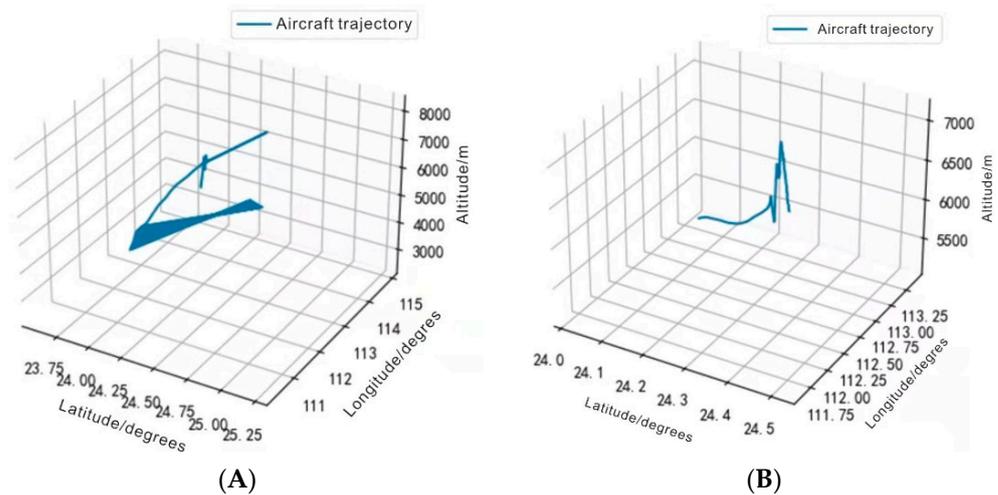


Figure 2. Invalid trajectory. (A) Invalid trajectory at higher altitudes. (B) Invalid trajectory at lower altitudes.

The general shape of the good trajectory is shown in Figure 3, which basically completely depicts the motion state of the aircraft in the whole terminal area. Due to equipment and other reasons, individual points have problems with missing values and are not smooth enough, which can be repaired by a reconstruction algorithm.

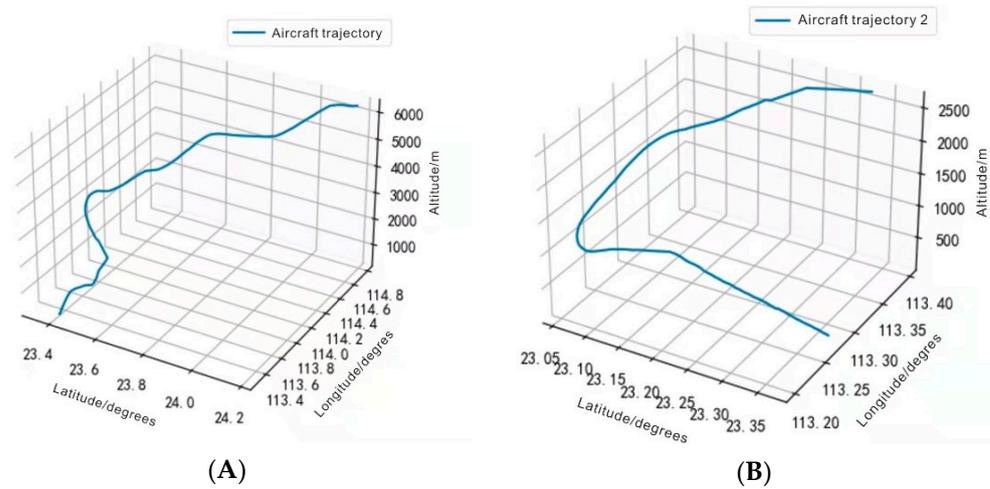


Figure 3. Valid trajectory. (A) Valid trajectory at higher altitudes. (B) Valid trajectory at lower altitudes.

2.2. Trajectory Correction Based on Operational Characteristics

Abnormal time points of trajectory caused by data errors in radar equipment and network transmission, as well as aircraft trajectory under abnormal conditions, will affect the operation law of aircraft under normal conditions of model learning [17,18]. The noise data need to be filtered before generating a training trajectory set. The former are unreal data caused by sampling error, which should be eliminated. To ensure that there is no lack of timescale, it should be filled by numerical prediction. The latter come from real aircraft operations, and the characteristics are not typical. The data belong to outlier samples that cannot be learned by the model, so they should also be eliminated.

For the abnormal time points process, it is usually assumed that the aircraft operates normally in a short period. The speed and heading remain stable. These two variables can determine the reasonable range of three-dimensional coordinates of each time point and mark the time points outside the range as abnormal points.

When detecting time point \vec{x}_{t_0} (a three-dimensional vector of position) in a time series with a time interval of Δt , the current moment is denoted by t_0 , and the nearest time window of length T is chosen, that is, T points before and T points after this point, if the point on one side is selected, and as many points as possible if the number is fewer than T (for example, for the starting point, select T points after it). The series is believed to be full, and the sequence is obtained after selection as:

$$\vec{x}_{t_0-T\Delta t}, \dots, \vec{x}_{t_0-\Delta t}, \vec{x}_{t_0}, \vec{x}_{t_0+\Delta t}, \dots, \vec{x}_{t_0+T\Delta t} \tag{1}$$

The displacement vector on magnitude and direction in detection is confined in the maximum velocity and the boundary value of the heading angle. The detection time t_0 has the following relationship with the previous time point. If this time point is the starting point, the latter time point is used to formulate the relationship:

$$\vec{x}_{t_0} = \vec{x}_{t_0-\Delta t} + \Delta\vec{x} = \vec{x}_{t_0-\Delta t} + \int_{t_0-\Delta t}^{t_0} |\vec{v}_t| \Delta\vec{\theta}_t dt \tag{2}$$

where $\Delta\vec{x}$, \vec{v}_t are the displacement vector and real-time velocity vector of the aircraft, and $\vec{\theta}_t$ is the direction vector of the real-time heading. The detection of the point is the detection of the mode and direction of the pair $\Delta\vec{x}$. Assuming the velocity maxima of the aircraft in

this sequence is, respectively, v_{max} , v_{min} , then the range of displacement distance, $|\vec{\Delta x}|$, can be determined:

$$\left| \int_{t_0-\Delta t}^{t_0} |\vec{v}_{min}| \Delta \vec{\theta}_t dt \right| < \left| \int_{t_0-\Delta t}^{t_0} |\vec{v}_t| \Delta \vec{\theta}_t dt \right| < \left| \int_{t_0-\Delta t}^{t_0} |\vec{v}_{max}| \Delta \vec{\theta}_t dt \right| \tag{3}$$

$$|\vec{v}_{min}| \Delta t < |\vec{\Delta x}| < |\vec{v}_{max}| \Delta t \tag{4}$$

The displacement vector is confined by two components, horizontal angle and vertical angle. If the horizontal angle boundaries of this sequence are α_1, α_2 , and the vertical angle boundaries are β_1, β_2 , then both are within the interval $(-\frac{\pi}{2}, \frac{\pi}{2})$.

If $\frac{\vec{\Delta x}}{|\Delta x|} = \begin{pmatrix} n_1 \\ n_2 \\ n_3 \end{pmatrix}$, the horizontal angle is α_t and the vertical angle is β_t :

After calculating the angle and distance range, Formula (5) is obtained:

$$\begin{cases} |\vec{v}_{min}| \Delta t < |\vec{\Delta x}| < |\vec{v}_{max}| \Delta t \\ \alpha_1 < \sin^{-1} \frac{n_2}{\sqrt{1-n_3^2}} < \alpha_2 \\ \beta_1 < \sin^{-1} n_3 < \beta_2 \end{cases} \tag{5}$$

Thus, the arc space area that \vec{x}_{t_0} may fall into is determined, and higher-quality trajectory data can be obtained by correcting the abnormal points outside the range.

After eliminating the anomalies that deviate from the trajectory, the corresponding predicted values are estimated to fill in the eliminated moment points by the mean values of velocity and heading in this time period, and this predicted value should also be able to pass the anomaly test described above.

For the moment point to be corrected, \vec{x}_{t_0} , the mean values of velocity, horizontal angle, and vertical angle are calculated for each moment point within the proximity time window of its length T . The estimated $\vec{\Delta x}^*$ value of the displacement vector at that moment point can be expressed as:

$$\vec{\Delta x}^* = |\vec{\Delta x}^*| \vec{n}^* \tag{6}$$

where $|\vec{\Delta x}^*|$ is the displacement distance and \vec{n}^* is the displacement direction, which can be expressed by the mean value as:

$$\begin{cases} |\vec{\Delta x}^*| = \bar{v} \Delta t \\ \vec{n}^* = \begin{pmatrix} \cos \bar{\beta} \cos \bar{\alpha} \\ \cos \bar{\beta} \sin \bar{\alpha} \\ \sin \bar{\beta} \end{pmatrix} \end{cases} \tag{7}$$

where \bar{v} , $\bar{\alpha}$, and $\bar{\beta}$ are the mean values of aircraft speed, horizontal angle, and vertical angle, respectively. Then, the coordinates of the current time, $\vec{x}_{t_0}^*$, can be determined according to the coordinates of the previous time point. The corrected coordinates are:

$$\vec{x}_{t_0}^* = \vec{x}_{t_0-\Delta t} + \bar{v} \Delta t \begin{pmatrix} \cos \bar{\beta} \cos \bar{\alpha} \\ \cos \bar{\beta} \sin \bar{\alpha} \\ \sin \bar{\beta} \end{pmatrix} \tag{8}$$

To sum up, the core idea of the methods of trajectory correction and reconstruction is to deduce the possible position range of the next time step according to the possible speed and heading range of the aircraft at the current time and adjust the reconstruction degree by limiting the position range. The default range is from the maximum value, MAX, to the minimum value, MIN. When the motion feature-based trajectory correction and

reconstruction technique described in the preceding section is implemented in Section 2.1, the reconstruction parameters λ and time window of length T are introduced to facilitate model adjustment. The larger the time window of length T is set, the higher the smoothness during reconstruction. λ is the coefficient limiting the reconstruction range, $0 \leq \lambda \leq 1$, and the larger the λ , the smaller the degree of re-configuration. The relationship between the limit range and λ is as follows. Taking the speed range as an example, it refers to the window mean value, AVR:

$$v_{AVR} + \lambda(v_{AVR} - v_{MIN}) \leq v \leq v_{AVR} + \lambda(v_{MAX} - v_{AVR}) \quad (9)$$

It can be seen that the default range $\lambda = 1$ is used, that is, it is allowed between the maximum values. The smaller the size of λ , the stricter the anomaly detection and the greater the degree of reconstruction. $\lambda = 0$ represents the direct reconstruction of the trajectory according to the mean value in the window.

In Figure 4, the original trajectory is the reconstruction target, and we selected different parameters, λ and T , to display the reconstruction results. For the convenience of observation, the original trajectory is drawn in the form of scattered points.

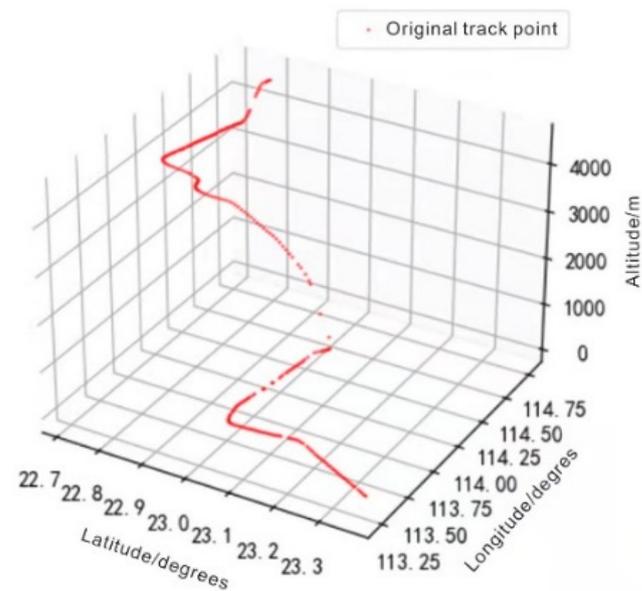


Figure 4. Original trajectory point.

Then, different parameter combinations were trialed for track reconstruction. The following shows the reconstruction effect of parameter combinations $\lambda = 0.5$, $T = 51$ and $\lambda = 0.2$, $T = 91$, which present the situations of lower and higher reconstruction degrees, respectively. The trajectory segment where the aircraft is about to ground was effectively smoothed on the left side of Figure 5A, but there was still significant jitter in the rest of the segment. The excessive reconstruction on the right side of Figure 5B led to a large amount of information loss and the destruction of the real motion state.

After the combination, the final selections, $\lambda = 0.5$, $T = 51$, were reconstructed, and the original track points were drawn on the same canvas. The effect is shown in Figure 6. It can be seen that the jitter of each track segment was smoothed without damaging the aircraft motion situation, which provided a reliable data guarantee for the follow-up work.

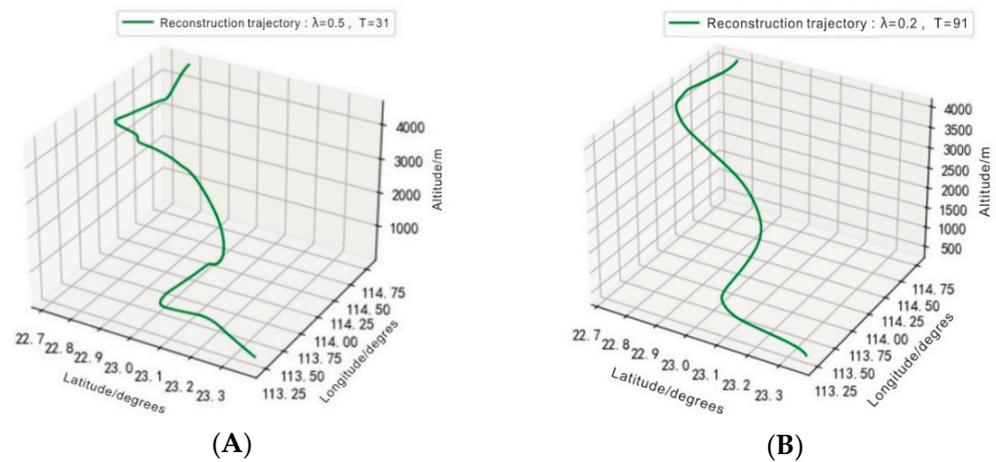


Figure 5. Reconstruction trajectory. (A) Reconstruction trajectory with $\lambda = 0.5$, $T = 51$. (B) Reconstruction trajectory with $\lambda = 0.2$, $T = 91$.

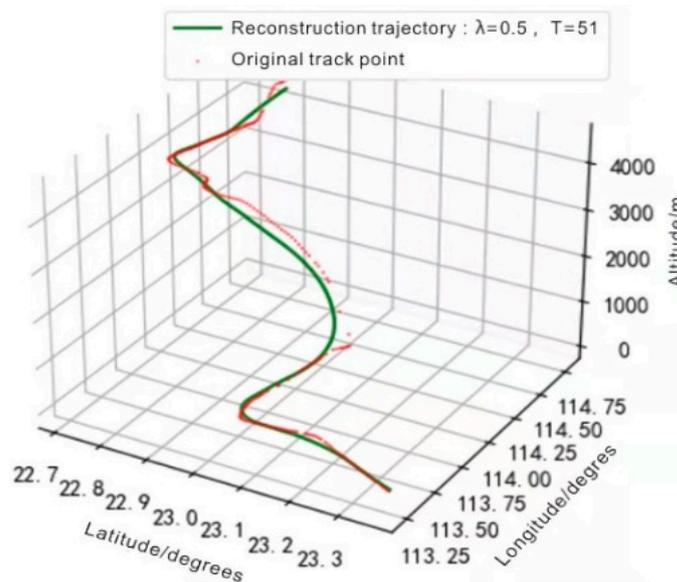


Figure 6. Trajectory comparison.

3. Clustering Analysis-Based Aircraft Operation Pattern Mining

3.1. Trace Feature Compression

3.1.1. Autoencoder of Trajectory Based on L21-Norm Constraints

The L1 regularization and relative entropy constraint method can effectively suppress the overfitting of the autoencoder [19]. The RSPCA (robust principal component analysis) method attempts to learn and express rather than resist the part of the original input that cannot be learned by the model, which is called the deviation. The ability of the autoencoder to restore the rest of the input data is improved by removing the deviation. Inspired by this, this paper applied a similar constraint method to the AE model, removed the abnormal trajectory, extracted the main features of the track data, and further reduced the reconstruction error while constraining the fitting ability of the autoencoder.

The optimization objectives of the conventional AE model are:

$$\min_{\theta} \| X - D_{\theta}(E_{\theta}(X)) \|_2 \tag{10}$$

where θ is the parameter matrix of the model (including two parts of encoding and decoding), $E_{\theta}(\cdot)$ is the encoder whose parameter matrix is θ , and $D_{\theta}(\cdot)$ is the decoder whose

parameter matrix is θ . The coding network and decoding network of the default model are multi-layer depth.

The input feature X consists of two parts: low-rank information matrix L and sparse deviation matrix S , as follows:

$$X = L + S \tag{11}$$

where L is a low rank with the same dimension of X and represents a low dimensional manifold that can be well-restored in the encoder in the input data, and S is the deviation that is difficult for the model to learn, which is usually caused by the characteristics of the data itself or noise. Traditional AE models are prone to overfitting problems when learning the deviation in the input. Some optimized AE models such as DAE (denoising autoencoder) resist the deviation, S , through noise learning. The denoising autoencoder adds noise to the input training data and makes the autoencoder learn to remove this noise to obtain the real input that is not contaminated with noise, so that the most important features can be extracted, and more robust representations of the input data can be learned. The generalization capability is stronger than that of the general autoencoder. Here, convex relaxation technology was used to learn and express the deviation, so that when the model restores the information matrix, L , it can better extract the main traffic flow in the airspace, non-linearly. In this paper, the coding model using this deviation extraction method is called the filtered autoencoder (OFAE).

The optimization objectives of OFAE are as follows [20]:

$$\begin{cases} \min_{L,S} \rho(L) + \lambda \| S \|_0 \\ \text{s.t. } \| X - L - S \|_F^2 = 0 \end{cases} \tag{12}$$

where λ is the control parameter of medium sparsity, S , $\rho(\cdot)$ is the rank of the matrix, $\| \cdot \|_0$ represents the number of non-zero elements, and $\| \cdot \|_F$ represents F-norm.

Then, the kernel norm and 1 norm were used to replace the rank operator and 0 norms for optimization, to solve the NP-hard problem of the matrix rank operator and 0 norm optimization. The compressed vector can obtain the linear mapping of the original input, and the following optimization objectives were obtained:

$$\begin{cases} \min_{\theta,S} \| L - D_{\theta}(E_{\theta}(L)) \|_2 + \lambda \| S \|_1 \\ \text{s.t. } X - L - S = 0 \end{cases} \tag{13}$$

where θ is the parameter matrix of the model (including two parts of encoding and decoding), $E_{\theta}(\cdot)$ is the encoder whose parameter matrix is θ , and $D_{\theta}(\cdot)$ is the decoder whose parameter matrix is θ .

The model obtained a feasible optimization objective, but further analysis is needed for the task of reconstructing the track data. The AE-processed data are fed into a density-peak-based clustering model. To make similar traces more easily identifiable, AE is needed to remove traces with the anomalous patterns mentioned earlier and to make the reconstructed trace denser. In this paper, we assumed that there are two types of deviations to be separated: one is the deviation of dimensional track features to be compressed, and the other is the deviation of anomalous tracks. By applying the deviation matrix, S , the penalty term is constrained using the L21-norm, and both requirements can be achieved simultaneously. The L21-norm is defined as follows:

$$\| S \|_{2,1} = \sum_{i=1}^n \| s_i \|_2 = \sum_{i=1}^n (\sum_{j=1}^m s_{ij}^2)^{1/2} \tag{14}$$

where s_i is the row vector of the i row of the matrix S , and s_{ij} is the element of the i row and j column of the matrix. L21-norm found 2 norms for the row and 1 norm for the column of the matrix and restricted the sparsity of the row and column at the same time.

When processing track data, the time-series characteristics of a track are tiled into rows, and then the sample set is processed with equal length. All tracks are aggregated by

columns to obtain the tracking matrix. For track data, the constraint of the L21-norm can be expressed as that the time sequence points in the reconstructed track are as compact as possible and the spacing between different tracks is as short as possible. The elimination of rows and columns in the original track matrix during decoding represents the characteristics of removing deviated tracks and compressing redundancy, respectively. Therefore, it is confirmed that the optimization objectives of the track compression model based on OFAE are:

$$\begin{cases} \min_{\theta, S} \| L - D_{\theta}(E_{\theta}(L)) \|_2 + \lambda \| S \|_{21} \\ \text{s.t. } X - L - S = 0 \end{cases} \quad (15)$$

The L2-norm constraint is used for the tiled track feature sequence to make the timing points as compact as possible, and the distance function is defined to calculate the track distance. However, the track characteristics after isometric processing and compression, and the differential correlation characteristics of heading, speed, and other positions will be distorted. Therefore, in the input track characteristics, this variable information needs to be added directly, and the model is difficult to mine directly from the location data.

Since the optimization goal has no clear requirements for the codec function, sigmoid is used as the activation function of neurons. The weight matrix of the decoder selects the transpose of the encoder weight matrix, and the bias term is independent. The encoding and decoding functions are as follows:

$$\begin{cases} E_{\theta}(x) = E_{W,b}(x) = \text{sigmoid}(Wx + b_E) \\ D_{\theta}(x) = D_{W,b}(x) = \text{sigmoid}(W^T E_{W,b}(x) + b_D) \end{cases} \quad (16)$$

where x is the input vector, W is the weight matrix from the input layer to the hidden layer, b_E is the bias matrix from the input layer to the middle layer, and b_D is the bias matrix from the middle layer to the output layer.

3.1.2. ADMM-Based Model Solving Algorithm

ADMM can decompose large-scale equality constrained optimization problems into multiple independent subproblems for distributed optimization. When optimizing a sub-problem, the rest of the optimization function should remain unchanged. The optimization problem of OFAE can be divided into two parts: $\| L - D_{\theta}(E_{\theta}(L)) \|_2$ and $\lambda \| S \|_{21}$.

Since the former optimization solution is differentiable, when the deviation matrix remains unchanged, the classical back-propagation algorithm can be used to train it. The latter optimization is relatively complex, and the near-end gradient method can be used to solve the norm optimization problem.

When dealing with the optimization of $\min(\lambda h(x))$, for the function $h(x)$ that is non-differentiable at the point x , the near-end gradient method looks for an alternative point, u , to make it as small as possible and close to the original non-differentiable point, x . u is called the proximal operator, $prox_{\lambda, h}(x)$, at x . The proximal operator of L21-norm is:

$$prox_{\lambda, l_{2,1}}(s)_i = \begin{cases} s_i - \lambda \frac{s_i}{\|s_i\|_2} & , \|s_i\|_2 > \lambda \\ 0 & , \|s_i\|_2 \leq \lambda \end{cases} \quad (17)$$

where s_i is the row vector of the i row of the matrix S , and λ is the sparsity parameter.

The near-end operator determines the next step of the iteration during L21-norm optimization, and finally obtains the track data after OFAE compression and depolarization, which provides a high-quality dataset with low dimension and low noise for the clustering algorithm. The complete model solving algorithm based on the ADMM framework is shown in Figure 7.

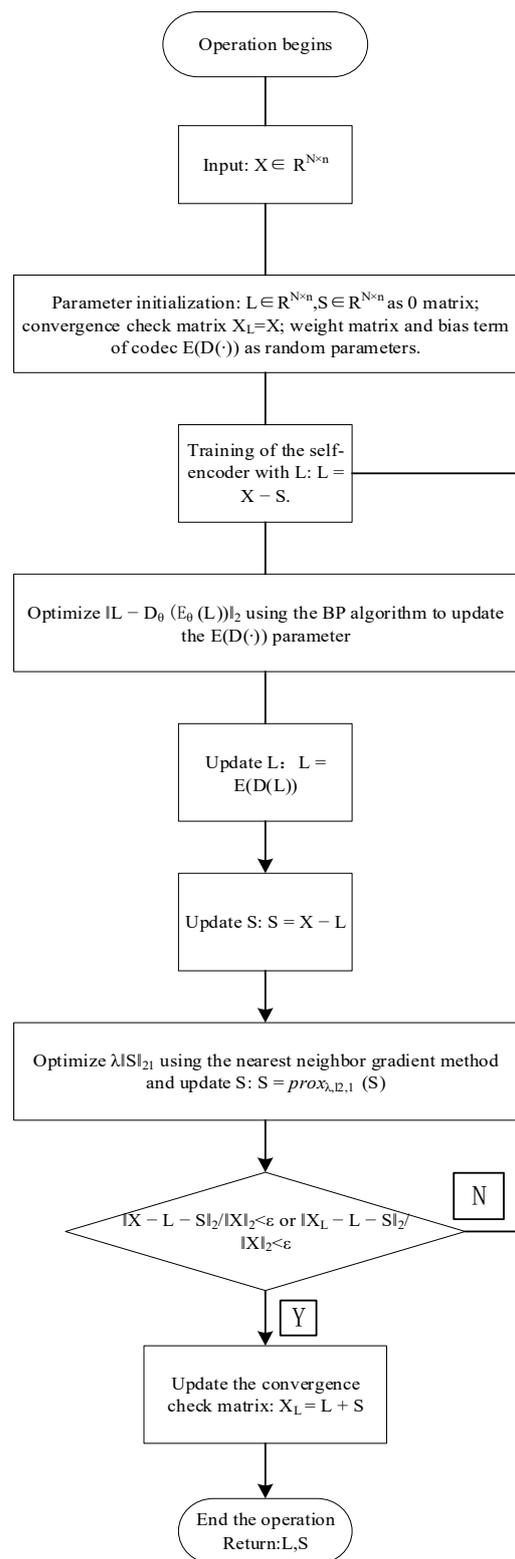


Figure 7. The flow chart of the algorithm based on ADMM.

3.2. Historical Track Clustering

The density peak clustering algorithm is a clustering algorithm proposed in recent years. Compared with the traditional DBSCAN clustering algorithm, it has a faster solution speed, and can extract the points that are most likely to become the class center in the global data through the two indicators of local density and similarity measurement,

which solves the problem that it is difficult for the traditional algorithm to distinguish low-density clusters.

Although DPCA can effectively identify some tactical traffic flows that form a much lower density than normal flight patterns due to control or regional reasons, DPCA cannot identify abnormal data, and abnormal trajectories that deviate significantly from normal data are also directly classified into nearby cluster classes, so this paper used OFAE in combination with DPCA, and removed the deviation points from the original data by OFAE, reconstructing the trajectory. At the same time, the normal trajectory was condensed to improve the local density, which makes up for the defect that DPCA cannot denoise.

After AE was used to compress the track, the time characteristics of the track were erased, leaving only the spatial characteristics. Therefore, the Euclidean distance was used to measure the similarity between the two tracks. The hard threshold was selected as the distance threshold of the model parameter, and the step count function was selected as the density statistical function, $\chi(\cdot)$. Then, the local density of the track i was calculated as follows:

$$\rho_i = \sum_j \chi(d_{ij} - d_c) \tag{18}$$

where $\chi(\cdot)$ is the density statistical function, which specifies the calculation of density when the condition is satisfied, and here the step counting function was chosen. d_{ij} is the distance between the points i and j , and d_c is the distance threshold, which specifies the condition for density counting and can be determined by hard or soft thresholding (e.g., Gaussian kernel function).

Next, the similarity distance, δ_i , of point i was calculated, and the rule is quite simple: if point i is the point with the largest local density, then δ_i is the distance between that point and the point farthest away from it; conversely, the shortest distance between that point and all points with local density higher than itself is calculated. Then, we have:

$$\delta_i = \begin{cases} \min_{j:\rho_j > \rho_i} \exists \rho_j > \rho_i \\ \max & \text{other} \end{cases} \tag{19}$$

where ρ_j and ρ_i are the local densities at points j and i , respectively, min is the shortest distance between point i and all points with a local density higher than i , and max is the distance between point i and the farthest point from i when ρ_i is the local maximum.

As shown in Figure 8, the dataset contains two cluster classes, which are labeled in red and blue in Figure 8A. The local density, ρ_i , and similarity distance, δ_i , were calculated for all data points to obtain Figure 8B.

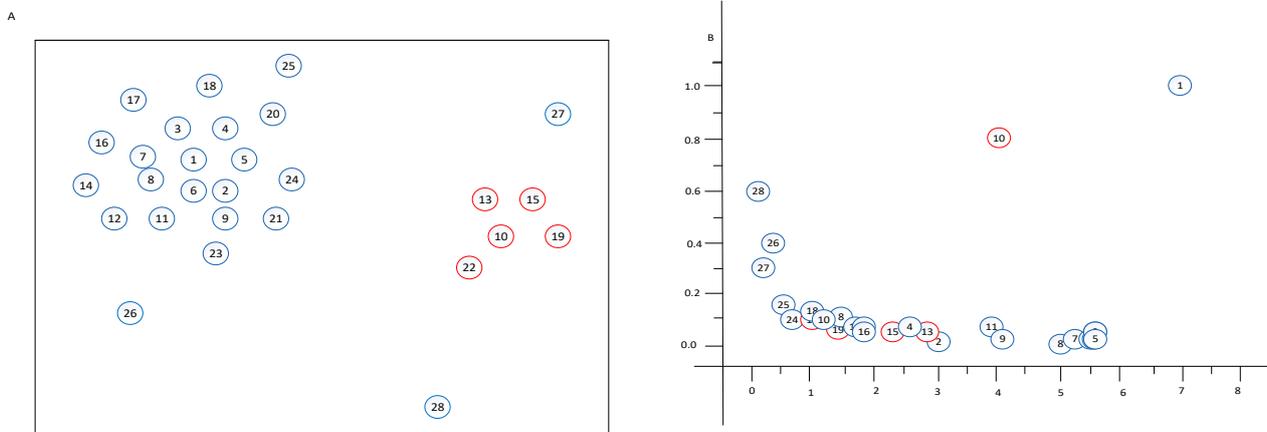


Figure 8. Dataset to be classified. (A) Dataset cluster class. (B) The figure of similarity distance.

In Figure 8B, points 10 and 1 have the maximum local density, ρ_i , and similarity distance, δ_i , which are the cluster centers of the two cluster classes, respectively, so the cluster centers can be determined directly from these two metrics without iterations. However, the more obvious drawback of DPCA is the high empirical component, which relies on the subjective tendency of the user in the problem of cluster center selection. In this example in Figure 8B, the classification result is clear, and points 1 and 10 can be clearly distinguished from other points in Figure 8B. However, in practice, it is often necessary for the user to define a function, $\gamma_i(\rho_i, \delta_i)$, to measure the centroids based on domain experience, balancing the tendency of the two metrics in determining the centroids. A simple choice is to use the form of the product of the two, $\gamma_i = \rho_i \delta_i$, and select those points with the largest γ_i as class centers.

$\chi(\cdot)$ with d_{ij} , specifically:

$$\chi(x) = \begin{cases} 1, & x < 0 \\ 0, & x \geq 0 \end{cases} \tag{20}$$

$$d_{ij} = \|L_i - L_j\|_2 \tag{21}$$

where L_i and L_j are the OFAE-compressed track vectors.

From the above equation, the local density, ρ_i , is the number of tracks in the track set with a distance less than d_c between the i -th track. The local density was calculated for all points in the dataset in turn and sorted in descending order.

We assumed that the dataset contains M tracks in total, and $\{q_n\}_{n=1}^M$ is a subscript sequence of $\{\rho_i\}_{i=1}^M$ in descending order; that is, if $\rho_{q_1} \geq \rho_{q_2} \geq \dots \geq \rho_{q_M}$ is satisfied, q_1 is the subscript of the track with the highest local density, and q_M is the subscript of the point with the smallest density.

Then, we calculated the similarity distance, δ_i , of the track data i . Assuming the rank of i in the local density ranking is n , then $i = q_n$. If i is not the track with the largest local density, then δ_i is the minimum distance between the track and other tracks with higher density than itself, and vice versa is the maximum distance, that is, δ_i :

$$\delta_i = \delta_{q_n} = \begin{cases} \min_{l:l < n} (d_{q_n q_l}) & n \geq 2 \\ \max_{l \geq 2} (d_{q_n q_l}) & n = 1 \end{cases} \tag{22}$$

where q_l is the subscript j of the other track, and to reflect its density relationship with track i , the uniform sequence of subscripts $\{q_n\}_{n=1}^M$ in descending order of density is expressed as q_l and q_n .

We calculated the center measurement parameters $\gamma_i = \rho_i \delta_i$. The points with high local density and long-distance, and greater than the threshold γ_c , can be identified as density peaks or cluster centers. The remaining points are assigned to the same cluster adjacent to their nearest high density without iteration. The flow chart of the algorithm is shown in Figure 9.

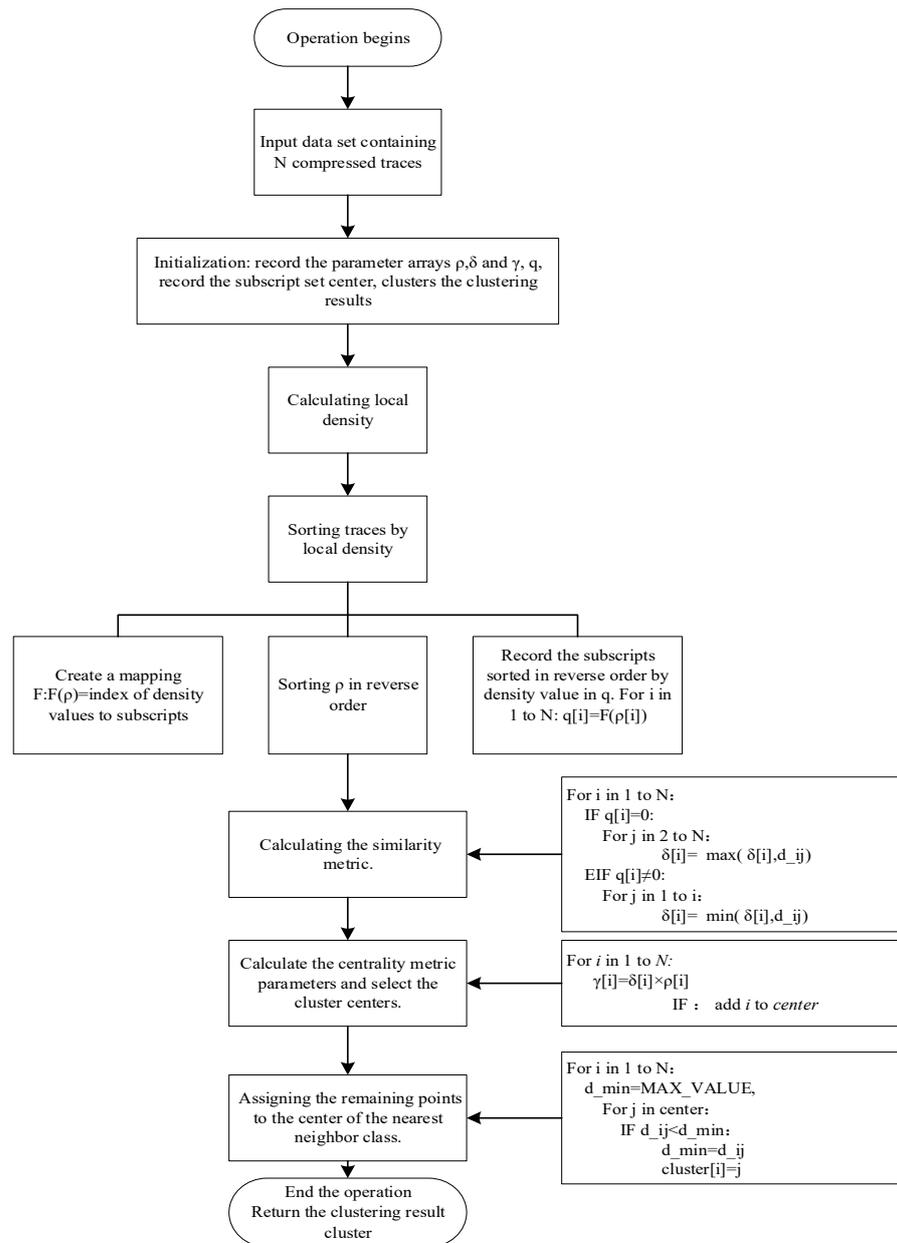


Figure 9. The flow chart of the algorithm.

4. Simulation

4.1. Experimental Design

After modifying and reconstructing the sample set in a single track, AE was used to reconstruct it as a whole. The track correction method described in Sections 2.1 and 2.2 was used to correct the track sample set of Baiyun Airport, and after removing the seriously damaged invalid tracks, there were 1263 remaining tracks in the sample set. When establishing the clustering model, due to the relatively fixed operation mode in the terminal area and strong overall regularity, a large number of samples was not required. At the same time, to verify the feasibility of the algorithm, it was necessary to horizontally compare the models measured by DTW. To avoid time consumption, 10% of the tracks will be randomly selected for the analysis of typical operation modes.

Firstly, the sample set used by OFAE was constructed. The input of the autoencoder is required to be a matrix with fixed dimensions, so the track samples cannot be directly used as input and need to be isometric. The previous reconstruction fills in the vacancy value of the timestamp, which is essentially an up-sampling, but in fact, the too-long sequence

is not friendly to the autoencoder, so it was necessary to carry out down-sampling. The length of the sequence was limited to 100, that is, each track contained 100 track points, and the characteristics of the track points are as follows:

$$P_t = (x, y, z, v_H, v_V, \sin \theta, \cos \theta) \quad (23)$$

where x, y, z are the three-dimensional position coordinates of the aircraft. For the horizontal speed, v_H , and vertical speed, v_V , of the aircraft, the timestamp information of the original track was lost in the process of up- and down-sampling, so the differential term was introduced to describe the instantaneous state of the aircraft. Using θ as the heading, it can describe the instantaneous motion state of the aircraft as the speed. Here, the sine and cosine value was used instead of the angle itself. The sine and cosine value of the heading combined with the speed can directly reflect the position change, but the angle value has no actual physical significance.

The i th track in the clustering sample set is expressed as: $Trail_i = \{P_1, P_2, \dots, P_{100}\}$, with 700 dimensions in total, and then the dimension of the sample matrix was 126×700 . The numerical normalization was carried out according to the column, the information matrix, L , was obtained after use of the autoencoder and compression, and the reconstructed sample set, \hat{X} , was obtained after decoding the information matrix. The comparison effect of the two is shown in Figure 10, in which the green line on the left is the reconstructed track and the blue line on the right is the original trajectory.

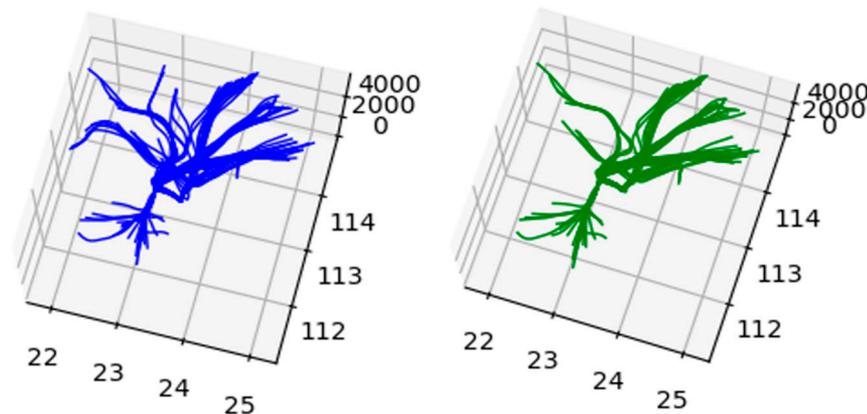


Figure 10. Reconstruction of the sample set comparison.

Observing the general form before and after reconstruction in Figure 10, it can be seen that OFAE has a good anomaly detection ability, which can remove the deviated track that is difficult to classify into a specific mode, and better maintain the overall situation of other normal tracks.

In Figure 11A, there are many scattered abnormal tracks in the red wireframe area of the original track. Although these tracks may also come from the real operation, they cannot be clustered due to their extremely low frequency. They are noise to the clustering model, so they need to be identified and eliminated during reconstruction.

Figure 11B shows the horizontal projection of the reconstructed trajectory set. It can be clearly seen that the scattered trajectory of the red dotted box in the original trajectory is eliminated under the constraint of one norm between groups. At the same time, the constraint of two norms within the group also makes the reconstructed trajectory more condensed in clusters with low density.

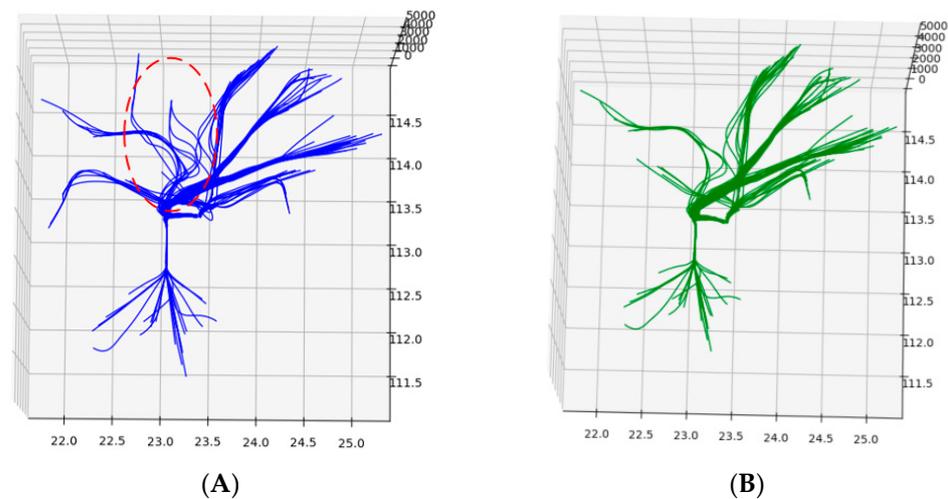


Figure 11. High noise area and denoised sample set. (A) Sample set before reconstruction. (B) Sample set after reconstruction.

4.2. Validation of Density Peak Clustering Effect

The track set reconstructed by OFAE obtains two-way compression within and between different tracks. Similar tracks are more cohesive, and the deviated tracks are removed. The Euclidean distance measurement can be directly used.

Since the data used are unlabeled sequences, the visual manual observation of the clustering effect and the contour coefficient were used as the evaluation indexes of the clustering effect.

The contour coefficient evaluates the clustering effect by two indicators: intra-cluster dissimilarity, a_i , and inter-cluster dissimilarity, b_i . For the sample sequence i , a_i is the distance between the track and other samples in the same cluster. The higher the value, the more similar the track is to the track in the same cluster, and the better the clustering effect is. b_i is the average distance between the track and the tracks of other clusters. The larger the value, the greater the difference between the track and the tracks of other clusters, and the better the clustering effect. The contour coefficient of the track, s_i , is expressed as follows:

$$s_i = \frac{b_i - a_i}{\max\{a_i, b_i\}} \quad (24)$$

The effectiveness of the clustering algorithm can be reasonably evaluated by calculating the average contour coefficient of the entire sample set.

Firstly, we attempted to use the combination algorithm of DTW and DBSCAN to cluster the trajectories, and after trying to combine multiple sets of hyperparameters, we chose the scan radius $\mu = 4.3$ and the minimum number of samples in the cluster $\eta = 9$. The clustering results are shown in Figure 12.

Overall, DBSCAN made a good distinction between the main traffic flows in several areas, but in the area of longitude $>114^\circ$, i.e., the red and yellow clusters, it can be seen that there was not only one traffic flow in these two clusters, but the interior of these two clusters cannot be separated only by parameter adjustment. This is because DBSCAN uses the same scanning radius on the whole sample set. When the distance between clusters of several clusters is quite different, the algorithm makes it easy to classify the clusters close together, whereas the light blue cluster on the left in the Figure 12 is obviously far away from all other clusters. In the process of parameter adjustment, it can be distinguished by almost any combination of parameters. However, due to the close distance between the red and blue clusters, if the scanning radius is too large, these tracks will directly become a cluster; if the radius is reduced, a large number of abnormal tracks will appear. The adjustment of the minimum number of samples in the cluster η is the same, and a large number of abnormal tracks will appear if the clustering requirements are improved. If the

requirements for clustering are reduced, unreasonable multiple small clusters will appear in the main traffic flow. In addition, although DBSCAN has the ability to recognize noise, in the yellow area with many abnormal tracks, the algorithm was unable to identify the potential traffic flow from the abnormalities, but tended to directly classify the tracks in this area into one category.

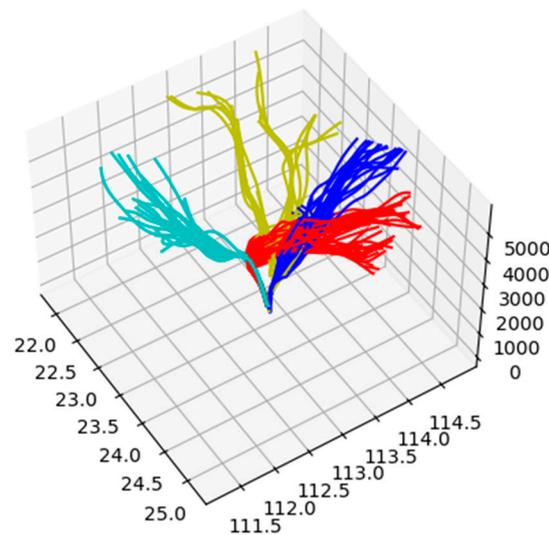


Figure 12. DTW + DBSCAN clustering result.

In summary, the algorithmic shortcomings of DBSCAN mainly lie in the rigidity of the parameters. However, even if the red and yellow clusters in Figure 12 were separated and clustered separately, and the similarity matrix was iterated and clustered again using DTW, DBSCAN still failed to distinguish these more similar trajectory clusters.

Next, the sample set reconstructed by OFAE was used to take the conventional Euclidean metric to measure the similarity of the trajectories, and the trajectories were clustered using DPCA. The truncation distance of DPCA was chosen as $d_c = 3.8$, while the choice of the center metric parameter is more flexible and was determined by the number of target cluster classes. Combined with the original trajectory pattern, the number of target cluster classes here should be 5 or 6. Figure 13 shows the clustering effect when the number of clusters was 5.

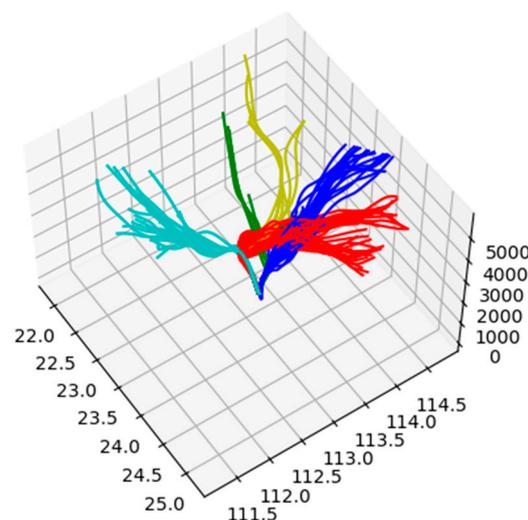


Figure 13. OFAE + DPCA clustering result.

The OFAE depolarization autoencoder mainly reconstructs the upper area of Figure 13 with a longitude greater than 114° and a latitude less than 22.5° . It can be seen that after the removal of abnormal tracks and the condensation of normal tracks, the traffic flow potential in this region was more obvious, and DPCA could well-extract the potential traffic flow in this region. However, similar to DBSCAN, the red clusters on the right cannot be further subdivided. Even if the number of target clusters was set to 6, the separated red clusters were not the red clusters in the current Figure 13. This is due to the rough distance measurement. In fact, after compression and reconstruction, the overall trajectory tends to be more cohesive. At the same time, the DPCA algorithm is more flexible than DBSCAN, which is essentially the identification of the center of clusters, and it is not necessary to use the same scanning radius for each cluster class. Even if there are relatively similar cluster classes, DPCA can effectively distinguish them as long as different centers can be identified. Next, we attempted to use the DTW iterative distance measurement matrix for the reconstructed sample set, and to cluster with DPCA again to further reduce the truncation distance, $d_c = 3.2$, to separate the red cluster. The number of cluster centers was 6, and the clustering results in Figure 14 were obtained.

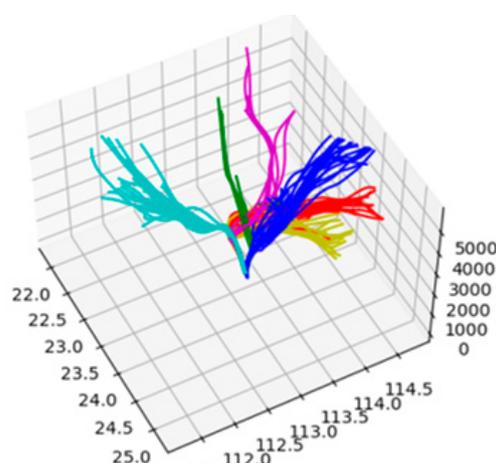


Figure 14. Final clustering result.

It can be seen that traffic flows with different densities were successfully distinguished and extracted after compression and reconstruction of AE, fine measurement of DTW, and the DPCA clustering algorithm.

As an example, we applied the standard instrument approach procedure and the area navigation approach procedure under the runway configuration of Baiyun Airport 01/02L/02R, as shown in Figure 15, and the light blue and yellow clusters and the aeronautical chart approach from the left and right sides. The number of flights arriving at the airport from the ATAGA and IGONO waypoints is large, the traffic flow is large, the track is more scattered, and the remaining clusters can accurately correspond to the approach procedures arriving from these two waypoints. It can be seen that the OFAE + DTW + DPCA algorithm proposed in this paper has strong robustness in the face of complex operating scenarios in the terminal area.

Combined with the contour coefficient, the performance of each algorithm is summarized in Table 3.

It can be seen that the combined algorithm of OFAE and DPCA took less time, had a better clustering effect than the traditional combined algorithm, and did not conflict with DTW. The reconstructed aggregation trajectory can also use fine distance measurement to achieve a better effect. Finally, the combined algorithm of OFAE + DTW + DPCA was successfully used to effectively extract each traffic flow.

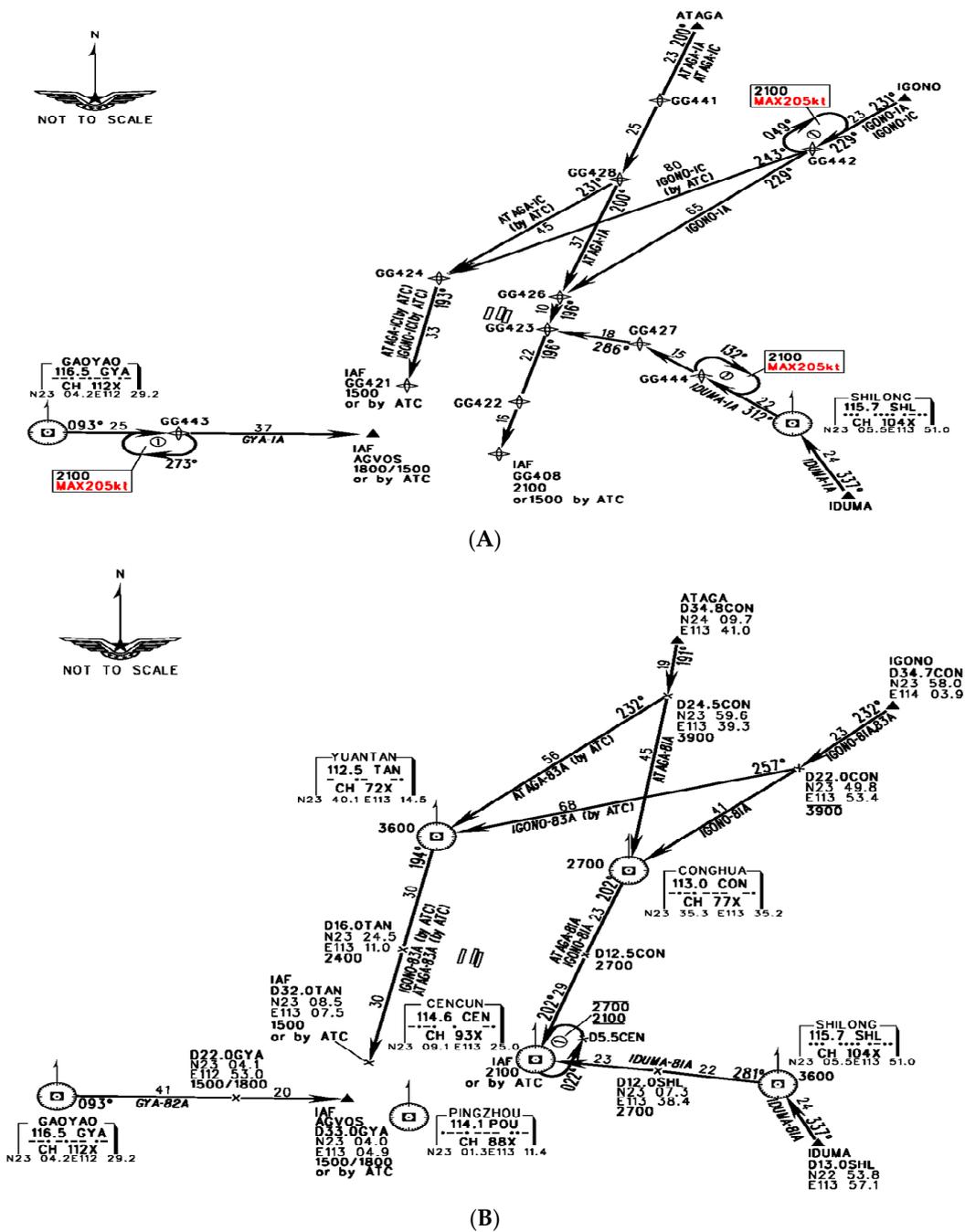


Figure 15. Baiyun Airport runway 01/02L/02R standard arrival flight procedures. (A) Standard instrument approach procedure. (B) Regional area navigation arrival procedure.

Table 3. Algorithm performance summary.

| Algorithm | Contour Factor | Algorithm Time Consumption | Effectiveness Evaluation |
|-------------------|----------------|----------------------------|--|
| DTW + DBSCAN | 0.46 | High | It is not possible to distinguish between similar cluster classes and extracted low-density cluster classes. |
| OFAE + DPCA | 0.65 | Low | Unable to distinguish between similar cluster classes. |
| OFAE + DTW + DPCA | 0.73 | High | Effectively extracted |

5. Conclusions

- (1) In the cluster analysis, a combined model of OFAE + DPCA was proposed to make up for the problem that DTW is time-consuming and DBSCAN finds it difficult to distinguish clusters with large density differences in the traditional model, DTW + DBSCAN. The track compressed by OFAE can directly use the Euclidean distance measurement. DPCA can extract the density peak and directly classify the adjacent tracks. The combination of the two greatly saves time.
- (2) In the process of solving the sparse autoencoder, the non-convex optimization objective was processed by convex relaxation, which was transformed into the optimization problem of L21-norm, and the ADMM algorithm was used for the step-by-step solution. Compared with the Lagrange solution with strong constraints, the ADMM solution allowed the model to deviate from the constraints to a certain extent while reducing the error function. When the deviation was limited, it was more conducive to the optimal solution.
- (3) In future research, we will consider high-quality datasets obtained by classifying and identifying typical aircraft operating modes and establish a track prediction model to make the prediction results more targeted. In the face of abnormal mode data with few similar samples at the same time, it can also provide a comprehensive prediction based on similar patterns and improve the accuracy of track prediction.

Author Contributions: Conceptualization, X.T., Y.L. and K.C.; methodology, X.T.; software, K.C. and Y.L.; validation, X.T., Y.L. and K.C.; formal analysis, X.T.; investigation, X.T.; resources, Y.L.; data curation, X.T. and Y.L.; writing—original draft preparation, X.T. and K.C.; writing—review and editing, Y.L. and K.C.; visualization, X.T.; supervision, K.C.; project administration, X.T.; funding acquisition, X.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (61773202, 52072174), the Foundation for the National Key Laboratory of Science and Technology on Avionics System Integration (6142505180407), and the Civil Aviation Management Institute of China Key Foundation of General Aviation (CAMICKFJJ-2019-04).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Wang, Q.; Liu, S.Z. Improvement of local outlier mining based on density. *Comput. Appl. Res. J. Aerosp. Inf. Syst.* **2014**, *31*, 1693–1696.
2. Hong, S.; Lee, K. Trajectory prediction for vectored area navigation arrivals. *J. Aerosp. Inf. Syst.* **2015**, *12*, 490–502. [[CrossRef](#)]
3. Xu, T.; Li, Y.X.; Lyu, Z.P. Research on track clustering based on normal distance of track point. *J. Syst. Eng. Electron.* **2015**, *37*, 2198–2204.
4. Wang, L.L.; Peng, B. Track clustering based on lofc time window segmentation algorithm. *J. Nanjing Univ. Aeronaut. Astronaut.* **2018**, *50*, 661–665.
5. Rehm, F. *Clustering of Flight Tracks*; AIAA: Reston, VA, USA, 2010.
6. Gariel, M.; Srivastava, A.N.; Feron, E. Trajectory clustering and an application to airspace monitoring. *IEEE Trans. Intell. Transp. Syst.* **2011**, *12*, 1511–1524. [[CrossRef](#)]
7. Wang, C.; Huang, B.J.; Wang, F. Applicability analysis of terminal area approach program control based on track clustering. *J. Nanjing Univ. Aeronaut. Astronaut.* **2013**, *45*, 130–133.
8. Wang, T.B.; Peng, B. Fuzzy clustering analysis of track based on 4D Track. *J. Transp. Inf. Secur.* **2013**, *31*, 38–42.
9. Zeng, W.; Xu, Z.; Cai, Z.; Chu, X.; Lu, X. Aircraft Trajectory Clustering in Terminal Airspace Based on Deep Autoencoder and Gaussian Mixture Model. *Aerospace* **2021**, *8*, 266. [[CrossRef](#)]
10. Bolić, T.; Castelli, L.; De Lorenzo, A.; Vascotto, F. Trajectory Clustering for Air Traffic Categorisation. *Aerospace* **2022**, *9*, 227. [[CrossRef](#)]
11. Corrado, S.J. *A Data-Driven Methodology to Analyze Air Traffic Management System Operations within the Terminal Airspace*; Georgia Institute of Technology: Atlanta, GA, USA, 2021.

12. Marzuoli, A.; Gariel, M.; Vela, A. Data-based modeling and optimization of en-route traffic. *J. Guid. Control. Dyn.* **2014**, *37*, 1930–1945. [[CrossRef](#)]
13. Bombelli, A.; Soler, L.; Trumbauer, E. Strategic air traffic planning with Fréchet distance aggregation and rerouting. *J. Guid. Control. Dyn.* **2017**, *40*, 1117–1129. [[CrossRef](#)]
14. Fu, J.P.; Chen, X.H.; Ge, X.Q. Face recognition by generalized kernel discriminant analysis via L_{2,1}-norm regularization. *J. Front. Comput. Sci. Technol.* **2017**, *11*, 124–133.
15. Liu, Z.; Liu, G.; Pu, J. Orthogonal sparse linear discriminant analysis. *Int. J. Syst. Sci.* **2018**, *49*, 848–858. [[CrossRef](#)]
16. Wen, J.; Fang, X.; Cui, J. Robust sparse linear discriminant analysis. *IEEE Trans.* **2019**, *29*, 390–403. [[CrossRef](#)]
17. Olive, X.; Grignard, J.; Dubot, T. Detecting controllers' actions in past mode S data by autoencoder-based anomaly detection. In Proceedings of the 8th SESAR Innovation Days, Salzburg, Austria, 3–7 December 2018; pp. 1–9.
18. Gallego, C.E.; Gomez, V.F.; Saez, F.J. *Discussion on Density-Based Clustering Methods Applied for Automated Identification of Airspace Flows*; Institute of Electrical and Electronics Engineers: Piscataway Township, NJ, USA, 2018; pp. 1–10.
19. Hinton, G.E.; Salakhutdinov, R.R. Reducing the dimensionality of data with neural networks. *Science* **2006**, *313*, 504–507. [[CrossRef](#)] [[PubMed](#)]
20. Candes, E.J.; Li, X.; Ma, Y. Robust Principal Component Analysis Preprint. *arXiv* **2009**, arXiv:0912.3599.