Aerospace Science and Technology 127 (2022) 107726

Contents lists available at ScienceDirect

Aerospace Science and Technology

www.elsevier.com/locate/aescte

An end-to-end framework for flight trajectory data analysis based on deep autoencoder network

Weining Zhang, Minghua Hu, Jinghan Du *

College of Civil Aviation, Nanjing University of Aeronautics and Astronautics, Nanjing, China

ARTICLE INFO

Article history:
Received 17 January 2022
Received in revised form 27 May 2022
Accepted 20 June 2022
Available online 27 June 2022
Communicated by Roberto Sabatini

Keywords:
Air traffic
Flight trajectory
Deep autoencoder network
Anomaly detection
Cluster analysis
Meteorological conditions

ABSTRACT

In order to jointly solve the tasks of abnormal trajectory detection and flow pattern recognition for flight trajectory data analysis, an end-to-end framework based on deep autoencoder network is proposed in this paper. Considering the coupling relationship between the two tasks, a structured sparsity-inducing norm is introduced into the reconstruction-based loss function to separate abnormal trajectories from the whole and extract low-dimensional and outliers-free representations from the remaining normal trajectories. On this basis, cluster assignment hardening is applied to further learn cluster-friendly representations as well as cluster assignment for each trajectory. The effectiveness and efficiency of the framework are validated on flight trajectories arriving at Hong Kong International Airport. Experimental results show that the proposed framework not only detects typical spatial anomalies, including holding patterns and rerouting patterns, but also identifies fine-grained cluster structures. Furthermore, it surpasses current state-of-the-art methods in terms of anomaly detection performance and cluster quality, with an improvement of 13.56% and 22.82%, respectively. With parallel computing, its time cost can be reduced to less than 1 second, which helps to perceive traffic situations and monitor abnormal behaviors in real time.

1. Introduction

The increasing demand for air traffic has promoted vigorous development of air transport industry, but it has also brought about the emergence of airspace congestion, flight delays, and environmental pollution. To ensure the safe and efficient operation of flights, it is necessary to have in-depth insights into the behavior of aircrafts and the flow structure they form. In recent years, data-driven trajectory analysis has gradually become the mainstream way to achieve this goal [1–3]. Especially since the popularity of open-source platforms deploying crowdsourced automatic dependent surveillance-broadcast (ADS-B) receivers, such as the FlightRadar24 and the OpenSky Network, the availability of flight trajectories has been greatly improved. By using artificial intelligence and data mining technology, valuable knowledge and information can be obtained from massive historical trajectories, which is conducive to enhancing the level of airspace management or traffic flow management from different aspects, such as situation monitoring [4], configuration optimization [5], and performance evaluation [6].

In general, data-driven trajectory analysis mainly involves two primary tasks, namely *flow pattern recognition* and *abnormal trajectory detection*. The former usually applies clustering methodology to identify typical operations represented by a small number of nominal trajectories and takes them as empirical insights for post-operational analysis or as input parameters for day-of-operations decision support. And the latter uses anomaly detection techniques to find trajectories that obviously deviate from the majority of trajectories, which provides an opportunity to discover potentially significant events by exploring their generation mechanism. Although flow pattern recognition needs to detect and remove abnormal trajectories in advance, they are usually regarded as two separate tasks. In reality, these two tasks are closely coupled and interdependent. Typical patterns identified by clustering are vulnerable to abnormal trajectory, while detecting abnormal trajectory requires accurate boundary information from typical patterns. However, few studies consider the two tasks in a unified framework. A density-based clustering algorithm, Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [7], is one of the few that identify flow patterns in the context of abnormal trajectories and has been widely used in flight trajectory data analysis [8–10]. Nevertheless, there are still some challenges due to the large-scale volume and high update frequency of trajectory data, especially the data from ADS-B receivers. First of all, the cluster-

* Corresponding author.
E-mail addresses: zwn7900@nuaa.edu.cn (W. Zhang), minghuahu@nuaa.edu.cn (M. Hu), dujinghan@nuaa.edu.cn (J. Du).

https://doi.org/10.1016/j.ast.2022.107726
1270-9638/

Nomenclature

X	initial trajectory
$\hat{X}$	reconstructed trajectory
N	number of trajectory
K	dimension of trajectory
I	interpretable matrix with normal trajectories
S	sparse matrix with abnormal trajectories
E	mapping from the input layer to the hidden layer
D	mapping from the hidden layer to the output layer
Z	latent space of trajectory
Y	cluster assignment
P	target distribution
Q	Student's t-distribution
L_R	reconstruction loss
L_O	outliers loss
L_C	clustering loss
z_i	the learnt representation of trajectory i
μ_j	centroid of cluster j
y_i	cluster assignment of trajectory i
λ, β	balance factors for pre-training and fine-tuning process
α, η	learning rates for pre-training and fine-tuning process
θ_e, θ_d	the parameters of encoder and decoder
q_{ij}, p_{ij}	calculated and target probabilities that trajectory i is assigned to cluster j
T_{al}, T_{gt}	abnormal trajectory sets identified by algorithms and ground truth

Abbreviations

UAV	unmanned aerial vehicle
GPU	Graphics Processing Unit
API	application programming interface
ATC	air traffic control
TMA	terminal maneuvering area
HKIA	Hong Kong International Airport
ICAO	International Civil Aviation Organization
ADS-B	automatic dependent surveillance-broadcast
PCA	principal components analysis
LSTM	long short-term memory
ADMM	alternating direction method of multipliers
DBSCAN	density-based spatial clustering of applications with noise
KL	Kullback–Leibler
JC	Jaccard coefficient
F_1	F1-score
SI	Silhouette Index
DBI	Davies-Bouldin Index
RMSE	root mean squared error
iForest	isolation forest
DAE	deep autoencoder
DEC	deep embedded clustering
AE-AD	autoencoder-based anomaly detection
DAE-CAD	deep autoencoder based clustering with anomaly detection

ing structure of DBSCAN largely depends on the distance measure function (such as Euclidean distance), and its discriminative ability is limited in high-dimensional space. In addition, the uneven distribution of the flight trajectory further magnifies this problem [11]. As the data size increases, the time consumption of DBSCAN algorithm is also huge, which affects its feasibility and scalability.

Such limitations can be solved with the help of deep learning. As an advanced representation learning method, it has rapidly become the state-of-the-art technology in various fields of aerospace, such as unmanned aerial vehicle (UAV) path planning [12,13], air traffic situational awareness [14,15], aircraft health management [16], etc. Among various network structures, deep autoencoder (DAE) network can mine inherent features from massive data in an unsupervised learning manner. Specifically, it uses a deep neural network to first encode the input data into a hidden representation and then decode it back to a reconstruction one. After learning the model parameters that reconstruct data well, its low-dimensional features can be extracted from the hidden layer, since the number of neurons in the hidden layer is generally lower than the dimension of the input data. More importantly, by introducing novel regularization terms into the reconstruction-based optimization objective, the practical application of DAE network is further extended, such as ℓ_1-norm regularization term for sparse models [17] and Kullback–Leibler (KL) divergence term for generative models [18].

Inspired by the powerful variations and extensions of DAE network in previous works, this paper aims to develop an end-to-end framework for flight trajectory data analysis to simultaneously realize flow pattern recognition and abnormal trajectory detection. Based on the fact that abnormal trajectories are difficult to reconstruct, we first apply a robust DAE technique to alleviate the negative impact of abnormal trajectories on flow pattern recognition, where outliers loss is used to identify abnormal trajectories from all recorded trajectories, and reconstruction loss is used to extract clean low-dimensional representations from the remaining part. By this means, the coupling relationship between the two is resolved to a certain extent. Then, a fine-tuning process is applied to the learned representations to make them cluster-friendly ones by jointly optimizing reconstruction loss and clustering loss. In addition, the centroids of all cluster structures and the cluster assignment of each trajectory can be directly obtained after the training process. The experimental evaluations on arrival flight trajectories at Hong Kong International Airport (HKIA) show that the proposed framework effectively identifies typical spatial anomalies (such as holding patterns and rerouting patterns) and captures fine-grained cluster structures. Compared with classical and state-of-the-art methods, our framework achieves better performance in anomaly detection and cluster quality. Furthermore, we also illustrate the high efficiency of the framework, which shows that it is suitable for real-time analysis of air traffic. All in all, the original contributions in this paper are summarized as follows:

a) To the best of our knowledge, this is the first attempt to propose a DAE network-based approach to jointly solve abnormal trajectory detection and flow pattern recognition in flight trajectory data analysis, which can be regarded as a powerful tool to enhance the situational awareness of air traffic flow.

b) Two specially designed objective functions are used to implement the end-to-end training process and generate low-dimensional representations that are free from abnormal trajectories and suitable for clustering.

c) A detailed literature review of mainstream methods for flow pattern recognition and abnormal trajectory detection is provided. And the pros and cons of each method and the difference from the proposed method are also illustrated.

d) Extensive experimental results show that the effectiveness and efficiency of our proposed framework significantly outperform existing methods in the tasks of abnormal trajectory detection and flow pattern recognition.

The rest of this paper is organized as follows. Section 2 gives a detailed literature review about flow pattern recognition and ab-

normal trajectory detection. In Section 3, the proposed framework for flight trajectory data analysis based on DAE network is presented. Section 4 shows the experimental results with real data from HKIA. Section 5 draws the conclusion and prospects.

2. Related works

In this section, we respectively introduce the related work of flow pattern recognition and abnormal trajectory detection in flight trajectory data analysis, and emphasize the differences and connections with the proposed framework.

2.1. Flow pattern recognition

The literature in flow pattern recognition has discussed various methods to characterize traffic structure, from data representation and similarity definition to clustering methods. Gariel et al. [8] used the DBSCAN clustering algorithm to get spatial patterns of flights with principal components analysis (PCA) based features as input. This knowledge was subsequently used to construct complexity metrics to monitor the health of airspace. To further capture the time-varying relationship of air traffic, Enriquez [19] leveraged spectral clustering to analyze the persistence and uncertainty of the flow, and successfully applied it to determine the minimum required number of performance-based navigation procedures. Murça et al. [9] presented a data-driven framework to characterize traffic flow in multi-airport systems where density-based clustering and ensemble-based classification methods were used together to obtain trajectory patterns, followed by the K-means algorithm to extract representation of daily traffic flow. This is the first time the expensive computational cost associated with large-scale trajectory clustering has been noticed. Moreover, since the definition of similarity between trajectories largely affects flow structure identified by the clustering algorithm, how to design a distance function that meets specific requirements has also been discussed [11,20]. In view of the fact that Euclidean distance potentially assumes the importance of each trajectory position is the same, Corrado et al. [11] developed a weighted Euclidean distance for trajectory clustering in the terminal airspace and proved that it can enhance the discrimination ability of Euclidean distance in high-dimensional space. Andrienko et al. [20] proposed a distance function that only considers the positions of interest. By assigning relevant flags to trajectory positions, any part of the trajectory can be selected and clustered to deal with different analysis tasks flexibly. Compared with these classical methods, the proposed method does not need to explicitly design input features and distance functions. And it is an end-to-end architecture based on DAE networks, which takes the raw trajectories as input and the Euclidean distance in latent space as the similarity function.

In addition to classical clustering technology, [21] tried for the first time to apply deep clustering algorithm based on autoencoder network to flow identification. And experimental results illustrated that the learned low-dimensional features can better separate clusters. However, since it does not take into account the adverse effects of abnormal trajectories on the model training process, trajectories belonging to different clusters still have an inevitable overlap in the low-dimensional space. In contrast, our framework maintains the advantages of deep learning in finding low-dimensional and non-linear representations. Meanwhile, it can also handle outliers like traditional clustering methods represented by DBSCAN, so as to realize clustering with abnormal trajectories removal.

2.2. Abnormal trajectory detection

Abnormal trajectory detection has always been one of the core tasks in trajectory data mining and has received extensive attention in the aviation field in recent years [22]. As mentioned in Section 1, DBSCAN algorithm is robust to outliers, where the outliers are regarded as additional outputs. Therefore, many studies have tried to use it to find abnormal flight trajectories. Gariel et al. [8] applied DBSCAN algorithm to the resampled trajectories to identify spatial anomalies, and respectively explored their frequency as a function of aircraft type or time distribution. Olive and Morio [10] utilized DBSCAN algorithm to recursively cluster significant points of flight trajectory around airports, and built a dependency tree to label whether a trajectory is abnormal. The possibility of using these abnormal trajectories in risk assessment analysis was subsequently discussed. Jarry et al. [23] proposed an energy atypical behavior detection method for flight trajectories based on functional PCA (FPCA) and DBSCAN algorithm, where FPCA is used to extract features from trajectories with time-series characteristics, and DBSCAN identifies anomalies in trajectory segments obtained by sliding windows. Despite being relatively time-consuming, it can accurately locate different types of abnormal energy situations. In addition, Corrado et al. [24] used DBSCAN with its hierarchical algorithm to capture spatial anomalies and energy anomalies in ADS-B trajectory data, and analyzed the interdependent relationship between the two. Although the DBSCAN algorithm is popular in the above literature, its practicality is limited to a certain extent because it can only find abnormal trajectories without quantifying the degree of abnormality.

Another effort uses neural network as the basic structure to develop abnormal trajectory identification procedures for its advantages in processing large-scale high-dimensional trajectory data. For instance, Ji et al. [25] proposed to find abnormal trajectories by comparing the similarity between the standard trajectory model obtained by long short-term memory (LSTM) and the trajectory to be detected. Although the superior detection accuracy was verified by public flight trajectory from Flightradar24, their method requires a set of normal trajectories to be given in advance, which is often difficult to obtain in real-world problems. Besides, autoencoder-based anomaly detection technology has also been tried to detect and identify significant events in various operational contexts, from terminal maneuvering area (TMA) [26] to en-route airspace [27]. Specifically, based on the assumption that autoencoder can effectively reconstruct most normal trajectories, they regarded the reconstruction error as an anomaly score to quantify the severity of the trajectory deviation. Quantitative experimental results showed that trajectory with the highest anomaly score is often caused by bad weather conditions, and the second highest anomaly score is related to actions of air traffic controllers. Despite the valuable and interesting findings, the calculated reconstruction error of normal trajectory may be biased. This is because the autoencoder was trained on all trajectories, not on the set of normal trajectories. More recently, Corrado et al. [28] proposed to build an anomaly detection model based on deep autoencoder, in which aircraft trajectories, weather, and traffic information are integrated to identify anomalies in a specific context. Compared with the autoencoder-based methods, the proposed framework introduces advanced regularization terms to directly isolate those trajectories that are difficult to reconstruct (that is, abnormal trajectories), and utilizes the remaining part to obtain more accurate reconstruction by training the deep autoencoder. Here, accurate reconstruction means more faithful to the low-dimensional representation of the normal trajectory, which is helpful for flow pattern recognition task.

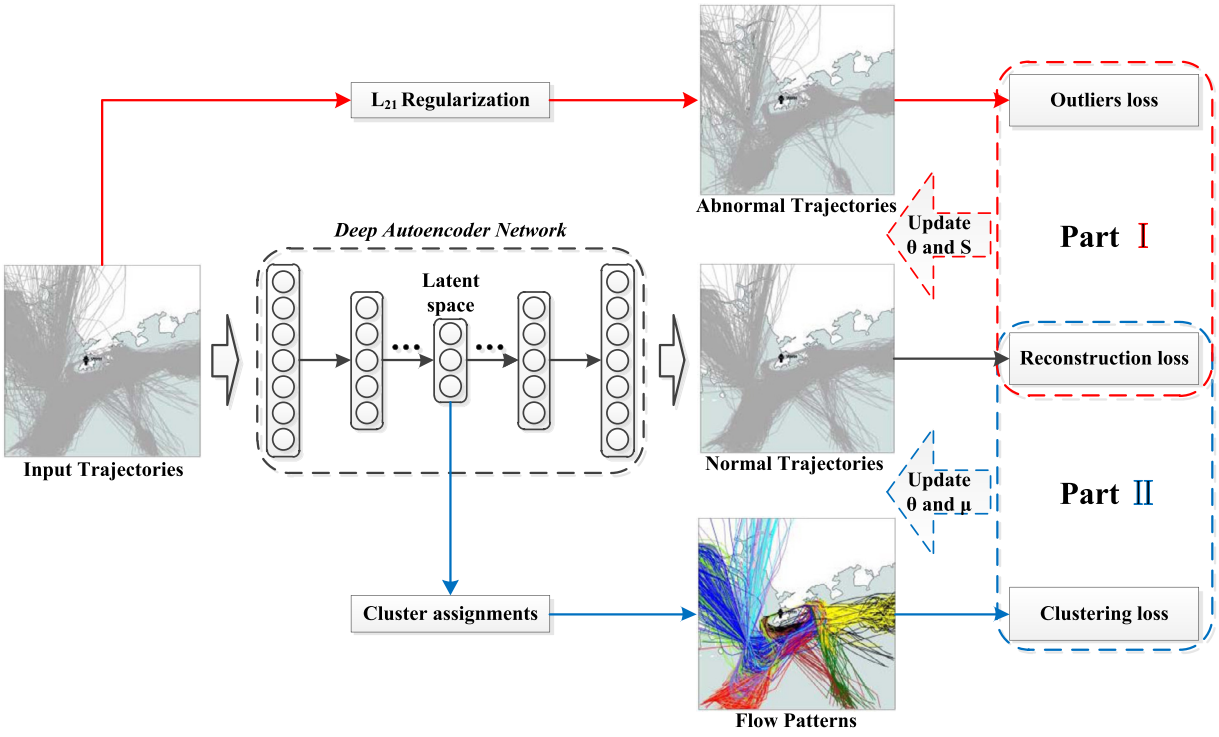


Fig. 1. The proposed framework based on deep autoencoder network. (Hereby, θ is the network parameters, S is the matrix containing abnormal trajectories, and μ is the cluster centroids.)

3. Methodology

3.1. Proposed framework

Before presenting the details of the proposed framework, the overview is shown in Fig. 1. Obviously, the whole framework is based on DAE network to successively realize the tasks of abnormal trajectory detection (Part I) and flow pattern recognition (Part II) in trajectory data analysis, which can be regarded as an end-to-end architecture.

For Part I, a DAE with $\ell_{2,1}$-norm regularization is used to find abnormal trajectories. By jointly optimizing reconstruction loss and outliers loss, the entire trajectory set is divided into two groups, namely the abnormal trajectories that are difficult to reconstruct and the normal trajectories that are well reconstructed, which relieves the coupling relationship between the two to a certain extent. Meanwhile, the low-dimensional representation that exists in the latent space is extracted from the normal trajectory. As for Part II, cluster assignment hardening is further applied to the low-dimensional representation to make it cluster-friendly and generate discriminative clusters. By jointly optimizing reconstruction loss and clustering loss, the cluster assignment of each trajectory can be directly obtained, together with the corresponding cluster centroids. It should be noted that the reconstruction loss here is to avoid the collapse of latent space learned in Part I.

In the following subsections, Section 3.2 first gives the basic principles of DAE network, then Section 3.3 and 3.4 describe the implementation details of Part I and Part II, respectively. Finally, the learning algorithm of the proposed framework is summarized in Section 3.5.

3.2. Representation learning based on deep autoencoder with reconstruction loss

In general, a vanilla DAE [29] is a multi-layer neural network composed of a symmetrical structure of encoder and decoder. It first encodes the input to a latent space and then decodes it back to a reconstruction one, which is formulated as:

$$\hat{X} = D_{\theta_d}(E_{\theta_e}(X)) \tag{1}$$

where $X, \hat{X} \in \mathbb{R}^{N\times K}$ are the description matrix of input trajectories and its reconstructed ones. N is the number of trajectories, and K is the dimension of each trajectory. θ_d and θ_e denote the parameters of encoder $E(\cdot)$ and decoder $D(\cdot)$, respectively. To make the reconstructed trajectories as similar as possible to the input trajectories, the objective function is set to minimize the reconstruction loss (L_R) as follows:

$$L_R(X;\theta) = \|X - \hat{X}\|_2 = \|X - D_{\theta_d}(E_{\theta_e}(X))\|_2 \tag{2}$$

where $\theta = \{\theta_e, \theta_d\}$. Note that $||\cdot||_2$ is the entry-wise ℓ_2-norm of matrix. After learning the model parameters using the method of back-propagation [30], low-dimensional representation can be obtained from the output of the encoder. Since DAE has a complicated nonlinear structure and its learning does not require any additional label information, it is widely used in the related fields of unsupervised representation learning [31]. In our framework, it is regarded as the most basic module.

3.3. Abnormal trajectory detection based on deep autoencoder with $\ell_{2,1}$-norm regularization

To capture abnormal trajectories and extract high-quality low-dimensional representations of normal trajectories, a deep autoencoder with $\ell_{2,1}$-norm regularization is used, which has been proven to be able to effectively identify structured anomalies in images [17]. Unlike vanilla deep autoencoder that reconstructs the entire input matrix X, it tries to split X into two parts, the interpretable part I with small reconstruction error and the sparse part S which is difficult to reconstruct.

Fig. 2 gives an intuitive illustration, where I represents normal trajectories that forms main traffic flows, and S contains abnormal

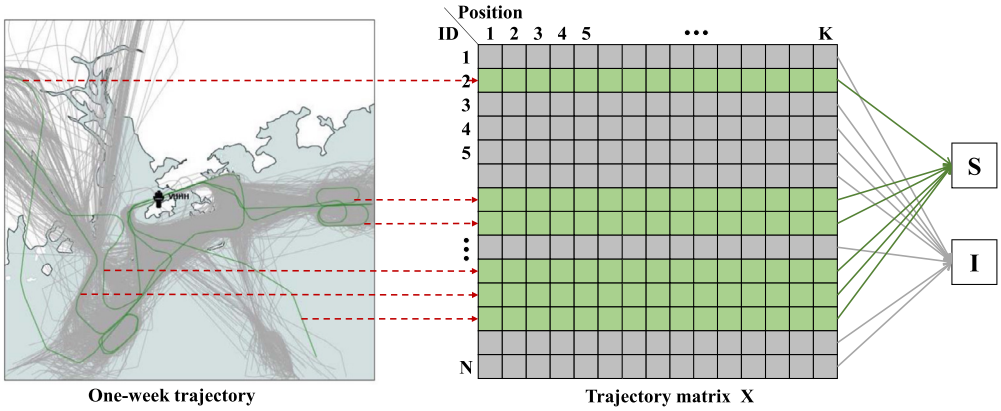


Fig. 2. Toy example of trajectory matrix X composed of I and S. Left is the visualization of one-week trajectories where abnormal trajectories are in green, and the remaining ones are in gray. Right is the corresponding matrix representation where each row is a trajectory and each column is a position. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

trajectories, those behaviors that deviate from most trajectories due to certain potential factors, such as traffic rerouting caused by severe weather and holding pattern caused by air traffic control (ATC) activities. To penalize these potentially abnormal trajectories in S, an outliers loss (L_O) based on $\ell_{2,1}$-norm regularization is introduced:

$$L_O(S) = \|S\|_{2,1} = \sum_{i=1}^{N} \|S(i,:)\|_2 \tag{3}$$

where $S(i,:)$ refers to the i-th row vector of matrix S. It can be viewed as summing the ℓ_2-norm of each row vector in matrix S. And minimizing L_O means reducing some rows to zero, also known as row sparsity. As a structured sparsity regularization, it is widely used in discriminative feature selection and abnormal event detection [32,33].

With L_R and L_O in mind, the objective function used to separate the two parts (i.e., I and S) is defined as follows:

$$\begin{aligned} &\min_{\theta,S} L_R(I;\theta) + \lambda L_O(S) = \|I - D_{\theta_d}(E_{\theta_e}(I))\|_2 + \lambda\|S\|_{2,1} \\ &\text{s.t. } X - I - S = 0 \end{aligned} \tag{4}$$

where λ is the balance factor between reconstruction ability and outlier robustness. A smaller λ means that more trajectories are isolated into S as outliers, thereby reducing the reconstruction error of the remaining part I. After alternating optimization (Details are shown in Section 3.5), the following indicator function is used to determine whether trajectory i is abnormal.

$$Indicator(i) = \begin{cases} 1, & ||S(i,:)||_2 \neq 0 \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

In other words, when the optimization is done, all non-zero row vectors in sparse matrix S correspond to abnormal trajectories, while those zero vectors correspond to normal trajectories.

3.4. Flow pattern recognition based on deep autoencoder with cluster assignment hardening

So far, not only the set of abnormal trajectories has been found, but also a low-dimensional, outliers-free latent space Z has been obtained from the well-learnt encoder $E_{\theta_e}(\cdot)$. Although the representation of the latent space can be used to identify flow patterns by combining traditional clustering algorithms, such as K-means or DBSCAN, cluster assignment hardening is further introduced to fine-tune the well-learned latent space to make it more suitable for clustering and directly generate cluster assignments for each trajectory.

Specifically, the Student's t-distribution [34] is used to calculate the probability q_{ij} that trajectory i is assigned to cluster j (also known as soft assignment), and its distribution Q is denoted as:

$$q_{ij} = \frac{\left(1 + \|z_i - \mu_j\|^2\right)^{-1}}{\sum_{j'}\left(1 + \|z_i - \mu_{j'}\|^2\right)^{-1}} \tag{6}$$

where $z_i = E_{\theta_e}(x_i) \in Z$ is the learnt representation of trajectory i, and μ_j is the centroid of cluster j. And the initial cluster centroids $\mu = \{\mu_j\}_{j=1}^{k}$ are estimated on the latent space Z of all trajectories using K-means algorithm. To further improve the purity of clusters by learning from high confidence assignments, the distribution P proposed by [35] is selected as the target distribution:

$$p_{ij} = \frac{q_{ij}^2 / \sum_i q_{ij}}{\sum_{j'}(q_{ij'}^2 / \sum_i q_{ij'})} \tag{7}$$

where p_{ij} is the target probability that trajectory i is assigned to cluster j. From its squared and normalized form, it can be seen that the target distribution P has a stricter probability of polarization (closer to 0 and 1) than the Q distribution. To measure the distance between these two probability distributions, a clustering loss (L_C) based on KL divergence is defined:

$$L_C(X;\theta,\mu) = \mathrm{KL}(P\|Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}} \tag{8}$$

Minimizing L_C means faithful to its own high confidence prediction, which can be considered as a self-training strategy [35].

With L_R and L_C in mind, the objective function for flow pattern recognition is defined as follows:

$$\min_{\theta,\mu} L_R(I;\theta) + \beta L_C(I;\theta,\mu) = \|I - D_{\theta_d}(E_{\theta_e}(I))\|_2 + \beta \mathrm{KL}(P\|Q) \tag{9}$$

where β is a balance factor between reconstruction ability and clustering loss. L_R is retained in this part to ensure that the latent space is not destroyed. It should also be noted that clustering is performed on the normal trajectories in I, not all trajectories in X. And after optimization (Details are shown in Section 3.5), the cluster assignment y_i for a given trajectory i can be directly obtained, as follows:

$$y_i = \mathrm{argmax}_{j=1,2,3,\ldots,k}\, q_{ij}. \tag{10}$$

3.5. Algorithm implementation of the framework

Essentially, the learning of the proposed framework can be seen as two procedures. The first is a pre-training process (i.e., Part I), which is used to obtain low-dimensional and outliers-free representation. The second is a fine-tuning process (i.e., Part II) to further make the representation suitable for clustering.

1) Pre-training process: Since Eq. (4) is a non-convex objective function, based on the idea of Alternating Direction Method of Multipliers (ADMM) [36], two independent parts, i.e., $L_R(I;\theta)$ and $L_O(S)$, are alternately optimized to achieve the overall minimization. Specifically, firstly, I is updated by $X-S$ (at the very beginning, S is initialized to a zero matrix) and with S fixed, $L_R(I;\theta)$ is optimized by back-propagation [30] with the following update rule:

$$\theta \leftarrow \theta - \alpha \frac{\partial L_R(I;\theta)}{\partial \theta} \tag{11}$$

where α is the learning rate. And then, based on the current $\theta = \{\theta_e, \theta_d\}$, I is updated by its reconstruction $D_{\theta_d}(E_{\theta_e}(I))$ and S is updated by $X-I$ to measure the reconstruction error of each trajectory. After that, with I fixed, $L_O(S)$ is optimized by using a proximal operator $prox_{\lambda,l_{2,1}}(S)$ [17], as follows:

$$prox_{\lambda,l_{2,1}}(S) = \begin{cases} S(i,j) \leftarrow S(i,j) - \lambda \frac{S(i,j)}{||S(i,:)||_2}, & ||S(i,:)||_2 > \lambda \\ S(i,j) \leftarrow 0, & ||S(i,:)||_2 \leq \lambda \end{cases} \tag{12}$$

where $S(i,j)$ represents the element corresponding to the *i-th* row and the *j-th* column of the matrix S, and the λ here is the same as in Eq. (4). It can be regarded as a shrinkage operator to realize $\ell_{2,1}$-norm optimization problems. For each iteration, the constraint is guaranteed by forcing $S = X - I$ or $I = X - S$. By alternatively reconstructing I and shrinking S, the high error trajectories in S are obtained and claimed as abnormal trajectories. And empirical evidence in [17] has shown the effectiveness of this kind of alternating optimization.

2) Fine-tuning process: After obtaining the preliminary estimate of network parameters θ and cluster centroids μ, we iterate between calculating the target distribution P and minimizing the total loss $L_R + L_C$ to learn their optimal ones during the fine-tuning process. Using gradient descent via back-propagation [30], the total loss is differentiable with respect to network parameters θ and cluster centroids μ, which are calculated as:

$$\theta \leftarrow \theta - \eta \frac{\partial L_R(I;\theta) + \beta L_C(I;\theta,\mu)}{\partial \theta} \tag{13}$$

$$\mu \leftarrow \mu - \eta \frac{\beta L_C(I;\theta,\mu)}{\partial \mu} \tag{14}$$

where η is the learning rate. Since the ultimate goal of the fine-tuning process is to identify common traffic flow patterns, the iterations continue until the cluster assignment for each trajectory no longer changes.

All in all, Algorithm 1 summarizes the optimization procedure of the proposed framework.

Algorithm 1 Training procedure of the proposed framework for flight trajectory data analysis.

Input: Original trajectory matrix $X \in \mathbb{R}^{N\times K}$, number of clusters k, balance factors λ and β, learning rates α and η.
Output: Abnormal trajectory matrix S, normal trajectory matrix I with corresponding cluster assignment Y, and optimal network parameters θ.

Initialize S with a zero matrix, and network parameters $\theta = \{\theta_e, \theta_d\}$ by truncated normal distribution with a mean of 0 and a standard deviation of 0.01.
repeat *– – Pre-training process*
  Set $I = X - S$.
  Update θ by Eq. (11).
  Set $I = D_{\theta_d}(E_{\theta_e}(I))$.
  Set $S = X - I$.
  Update S by Eq. (12).
until the total loss $L_R + \lambda L_O$ converges.
Initialize cluster assignment Y and corresponding centroids $\mu = \{\mu_j\}_{j=1}^{k}$ using K-means on latent space $Z = E_{\theta_e}(I)$.
repeat *– – Fine-tuning process*
  Compute distribution Q and P by Eq. (6) and Eq. (7), respectively.
  Update θ by Eq. (13).
  Update μ by Eq. (14).
  Compute cluster assignment Y by Eq. (10).
until the cluster assignment Y remains unchanged.
return S, I, Y, and θ.

4. A case study on Hong Kong International Airport

4.1. Data preparation

In order to effectively evaluate the proposed framework, flight trajectory data arriving at HKIA was collected from the OpenSky Network [37], a network that deploys crowdsourced ADS-B receivers. Since it can easily obtain real-time aircraft information, including timestamp (updating interval is about 1 second), digital identifier (unique 24-b International Civil Aviation Organization (ICAO) address), location (longitude, latitude, and altitude), ground speed, etc., OpenSky data is very popular in the academic community.

Fig. 3. Visualization of arrival flight trajectory.

Specifically, due to the rich application programming interface (API) and high scalability, the *traffic* toolbox [38] implemented in python was used to extract and process two-week (from June 15th to June 30th, 2019) ADS-B trajectory data from OpenSky. First, we intercepted all the trajectory data within the bounding box of the latitude of [21.3, 23.3] and longitude of [113, 115.2], and kept trajectories that landed at HKIA with a flight time greater than 10 minutes. In this paper, we consider the 2-dimensional spatial position (i.e., pairs of latitude and longitude) of the trajectory as the original input feature. To meet the input requirements of autoencoder for equal length dimension, these trajectories were then resampled to an update frequency of 1 second, and on this basis, a second resampling was performed to obtain 200 temporally evenly distributed positions as the trajectory representation. Finally, min-max normalization was used to rescale the range of each dimension to [0,1], and a dataset with 3649 trajectories was constructed for the following experiments. A snapshot of the trajectory is shown in Fig. 3. Preliminary observations reveal that the

flow structure formed by most of the trajectories is accompanied by a certain amount of messy and irregular trajectories, especially flights from the north and southeast. These outliers will inevitably affect the identification of typical operations, which is one of the problems to be solved by our framework.

4.2. Experiment settings and evaluation protocols

We named the proposed framework as DAE-CAD, which means deep autoencoder based-clustering with anomaly detection, and compared it with classic and state-of-the-art methods to verify the performance on the tasks of abnormal trajectory detection and flow pattern recognition. Specifically, Isolation forest (iForest) [39], DBSCAN [7], autoencoder-based anomaly detection (AE-AD) [26], and deep extension of AE-AD (DAE-AD) [28] are considered for abnormal trajectory detection. Among them, DBSCAN directly treats trajectories that cannot form clusters as abnormal trajectories, while the other methods filter them by calculating anomaly scores. As for flow pattern recognition, we choose two representative methods for comparisons, which are DBSCAN [7] and deep embedded clustering (DEC) [35], respectively. According to different dimensionality reduction techniques used for trajectory preprocessing, DBSCAN is further refined into PCA-based DBSCAN (PCA-DBSCAN), FPCA-based DBSCAN (FPCA-DBSCAN), autoencoder-based DBSCAN (AE-DBSCAN), and deep extension of AE-DBSCAN (DAE-DBSCAN). DEC is one of the competitive deep clustering methods, which learns representations for clustering by directly optimizing a KL divergence objective in the latent space learnt by the encoder.

Correspondingly, different evaluation metrics are used to analyze the performance of models on different tasks. To evaluate anomaly detection performance, we first generate label information by considering two types of spatial anomalies, including holding pattern and traffic rerouting. The holding pattern is a structured anomaly, manifested in the shape of a racetrack, usually used to delay an aircraft in flight [2]. Hereby, such anomalies are automatically labelled by judging whether there are one or more self-intersecting segments in a trajectory. For traffic rerouting, since there is no additional contextual information, experts represented by air traffic controllers manually label those trajectories with large deviations due to detours as this type. Then, based on the above ground truth, Jaccard Coefficient (JC) and F1-score (F_1) are used to compare the detection results of different methods. And they are defined as follows:

$$JC(T_{al}, T_{gt}) = \frac{\left|T_{al} \cap T_{gt}\right|}{\left|T_{al} \cup T_{gt}\right|}, \tag{15}$$

$$F_1 = \frac{2 * TP}{2 * TP + FP + FN}, \tag{16}$$

where T_{al} and T_{gt} are the abnormal trajectory sets identified by algorithms and ground truth, and TP, FP, and FN are the true positive, false positive, and false negative, respectively. In addition, for the evaluation of clustering performance, two validity indexes are introduced, namely Silhouette Index (SI) and Davies-Bouldin Index (DBI). As internal indexes, they quantitatively measure clustering results by considering both the compactness within the clusters and the separation between clusters, which are computed as follows:

$$SI = \frac{1}{N} \sum_{i=1}^{N} \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}. \tag{17}$$

$$DBI = \frac{1}{K} \sum_{i=1}^{K} \max_{j \neq i} \frac{c(i) + c(j)}{||\mu_i - \mu_j||_2}, \tag{18}$$

In SI, N is the number of trajectories, $b(i)$ is the minimum mean distance between trajectory i and other clusters, and $a(i)$ is the mean distance between trajectory i and all other trajectories in the same cluster. In DBI, K is the number of clusters, $c(i)$ is the mean distance between trajectories in cluster i, and μ_i is the corresponding centroid. It should be noted that the larger the JC, F_1, and SI, the better the performance, while DBI is the opposite.

4.3. Implementation details

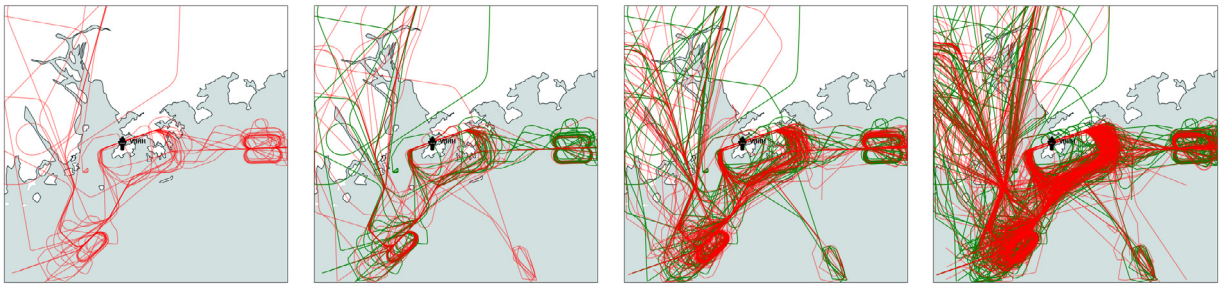
All experiments in this paper are carried out on a Dell G15 laptop with an Intel Core i7-11800H@2.30 GHz and a 16 GB DDR3 RAM, and the related algorithms are implemented based on Python (3.7.10) and TensorFlow (2.0.0). For the sake of fairness, the network dimensions of autoencoder-based methods (i.e., AE-AD and AE-DBSCAN) are set to D-100-D, where D is the dimension of input trajectory. As for deep autoencoder-based methods (i.e., DAE-AD, DAE-DBSCAN, DEC, and DAE-CAD), D-200-100-50-100-200-D are set as their network dimensions. In this paper, D is set to 400 since each trajectory consists of 200 location points determined by longitude and latitude. Besides, all layers are fully connected with the sigmoid activation function (i.e., $f(x) = (1 + e^{-x})^{-1}$). In order to obtain reliable reconstruction results, based on the Adam optimizer with a learning rate of 0.01 and a batch size of 512, autoencoder in AE-AD and AE-DBSCAN is trained for 200 epochs, and deep autoencoder in DAE-AD and DAE-DBSCAN is trained for 500 epochs.

As for DEC and DAE-CAD, in the pre-training process, the epoch for DEC is set to 500 while the epoch for DAE-CAD is set to 3000 (300 internal iterations for I times 10 external iterations for S). With the well-learned deep autoencoder in mind, we repeatedly perform the K-means algorithm 10 times in the latent space of all trajectories to obtain the best estimate of the initial cluster centroids. After that, in the fine-tuning process, the epochs for DEC and DAE-CAD are both set to 50. And the learning rates α and η are set to 0.01 and 0.001, respectively.

In addition, for the PCA-based method (i.e., PCA-DBSCAN and FPCA-DBSCAN), the first 100 principal components are kept for comparison with AE-AD and AE-DBSCAN. For iForest, the number of trees is set to 150, and the sub-sampling size is set to 256. It should be noted that the settings of all relevant hyper-parameters are determined by the grid search method, and the final settings correspond to the optimal performance of the specified task.

4.4. Performance analysis of abnormal trajectory detection

In order to analyze the utility of the proposed DAE-CAD in abnormal trajectory detection, we set a range of λ from 0.6 to 0.3 with the step-size of -0.1 and visualize the respective detected abnormal trajectories in Fig. 4. In each case, we divide these trajectories into two categories, where the green is the same trajectories detected by the previous λ, and the red is the new trajectories detected by the current λ. On the whole, as the λ decreases, the detected abnormal trajectories gradually increase. This is because a smaller λ will allow more trajectories into S as abnormal ones. Through preliminary statistical analysis, we found that the small λ completely covers the set of abnormal trajectories detected by the large λ, and the newly added ones have a smaller reconstruction error. This phenomenon indicates that the trajectories detected by a larger λ have a higher degree of abnormality, and as the λ decreases, the degree of abnormality of the newly added trajectories will also decrease. Specifically, Fig. 4(a) mainly captures abnormal trajectories coming from the north and east sides of the map, most of which have two abnormal types (i.e., holding pattern and traffic rerouting) at the same time. In contrast, Fig. 4(b) further regards trajectories with holding patterns from the southeast as abnormal



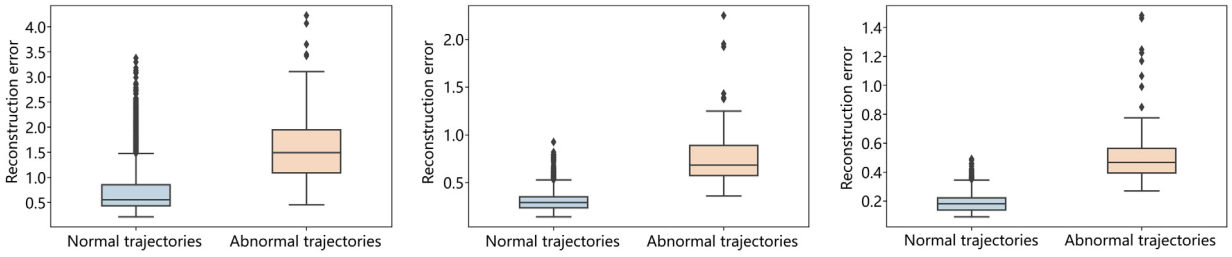
(a) $\lambda = 0.6$ (b) $\lambda = 0.5$ (c) $\lambda = 0.4$ (d) $\lambda = 0.3$

Fig. 4. The detected abnormal trajectories vary with the value of λ. For each sub-figure, trajectories in green are the same abnormal trajectories as its left figure, and trajectories in red are the new abnormal trajectories compared with its left figure.

Table 1
Performance comparison of different models in abnormal trajectory detection (%).

Models	JC					F_1				
	5%	10%	15%	20%	r_{opt}	5%	10%	15%	20%	r_{opt}
iForest	16.06	31.85	42.03	43.68	44.94	27.67	48.31	59.19	60.80	62.41
DBSCAN	20.82	40.58	53.10	54.08	57.78	34.47	57.73	69.37	70.19	73.26
PCA-DBSCAN	21.34	42.20	55.49	57.25	59.12	35.62	59.44	71.32	72.93	74.80
FPCA-DBSCAN	21.65	42.31	55.73	57.66	59.51	35.84	59.93	71.84	73.41	75.37
AE-AD	19.37	37.61	49.74	47.80	50.92	32.45	54.66	66.44	64.68	67.48
DAE-AD	25.39	48.03	64.16	66.29	68.10	40.50	64.89	78.17	79.73	81.77
DAE-CAD	**29.27**	**53.13**	**73.24**	**78.93**	**80.14**	**45.28**	**69.40**	**84.56**	**88.23**	**89.49**

The best results for the evaluation metrics are shown in bold.



(a) AE-AD (b) DAE-AD (c) DAE-CAD

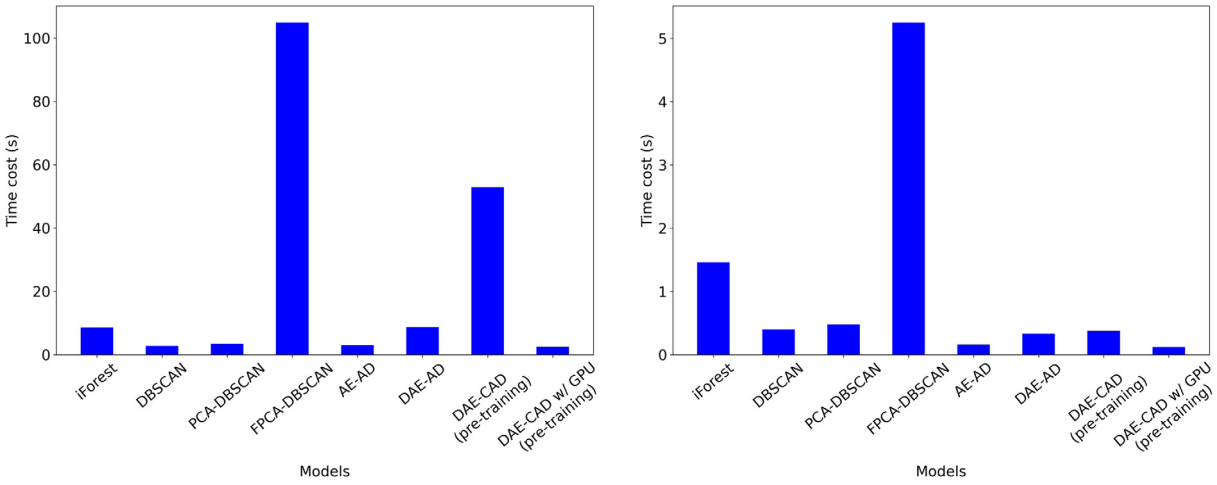
Fig. 5. Reconstruction error distribution of normal trajectories and abnormal trajectories.

ones. For Fig. 4(c) and Fig. 4(d), more trajectories with a single abnormal type and some incomplete trajectories are found, which have a lower degree of abnormality than the former.

Furthermore, two evaluation metrics are calculated to quantitatively compare the proposed DAE-CAD with other anomaly detection models. Table 1 shows the performance under the detection ratio (percentage of abnormal trajectories detected in all trajectories) of {5%,10%,15%,20%,r_{opt}}, where r_{opt} is the detection ratio corresponding to the respective optimal performance. The change in detection ratio is achieved by adjusting hyper-parameters of models, such as λ in DAE-CAD, and $MinPts$ and ϵ in DBSCAN. Four obvious observations were found. (1) The high-dimensional nature of the trajectory poses a challenge to abnormal trajectory detection problem, which is verified by the poor performance of iForest and DBSCAN. (2) Trajectory features extracted by PCA or FPCA can significantly improve the performance of DBSCAN, but the difference between the two is not obvious. (3) The better performance of DAE-AD than AE-AD proves that the latent space learned by deep autoencoder is more robust to abnormal trajectories than that learned by autoencoder. However, there is still a gap compared with DAE-CAD. Since all trajectories are used for model training, reconstruction of those abnormal trajectories will affect the quality of latent space. (4) The proposed DAE-CAD is superior to all other methods under any detection ratios and evaluation metrics. This apparent improvement illustrates the importance of a low-dimensional, outliers-free latent space.

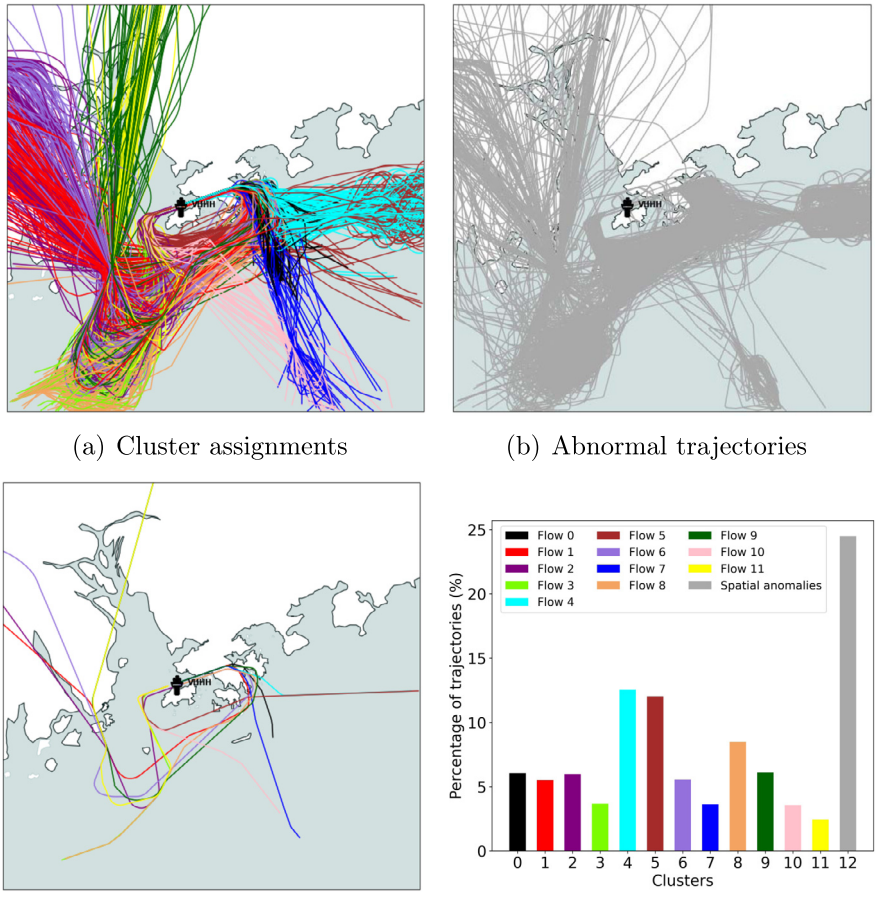
We also analyze the robustness of different models to abnormal trajectories by comparing the distribution of reconstruction error between the abnormal trajectory set labelled by ground truth and the remaining normal trajectory set. Fig. 5 gives the boxplot of reconstruction error of AE-AD, DAE-AD, and DAE-CAD from a statistical perspective. It can be seen from Fig. 5(a) that although there is a large overlap between the distribution of normal trajectories and abnormal trajectories, the reconstruction error of most normal trajectories is significantly smaller than that of abnormal trajectories, which reflects autoencoder has effectively learned the main behavior of flights. However, due to the existence of abnormal trajectories, AE-AD inevitably caters to them during the learning process, resulting in large reconstruction errors on the overall level. And this problem is alleviated when deep autoencoder is used to learn low-dimensional representations, just like the DAE-AD model shown in Fig. 5(b). In Fig. 5(c), the discrimination of reconstruction error is further enhanced, where the proposed DAE-CAD learns a more reliable latent space from a relatively clean set by isolating the identified abnormal trajectory.

To evaluate different models in terms of efficiency, Fig. 6 compares the time cost of model training and model testing (that



(a) Model training (b) Model testing

Fig. 6. Time cost for abnormal trajectory detection.

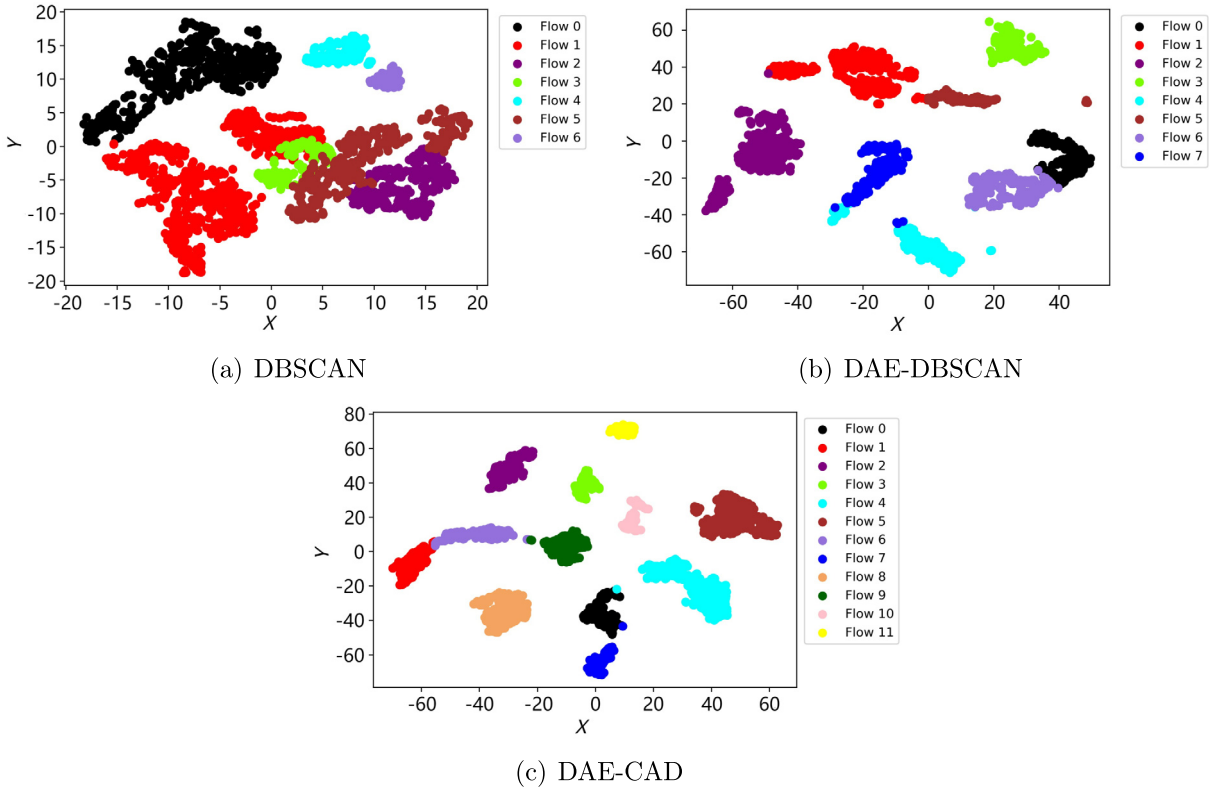


(a) Cluster assignments (b) Abnormal trajectories

(c) Cluster centroids (d) Cluster distribution

Fig. 7. Results of flow pattern recognition.

is, practical application) in abnormal trajectory detection task, respectively. Generally speaking, in the model training process, the efficiency of DBSCAN-based models largely depends on trajectory preprocessing techniques, which is confirmed by the obvious gap between PCA-DBSCAN and FPCA-DBSCAN. It is worth noting that the time cost of the proposed model (i.e., the pre-training of DAE-CAD) is higher than that of AE-AD and DAE-AD due to the increase of training epochs. One possible solution is to accelerate training with parallel processing architectures such as Graphics Processing Unit (GPU). And the experimental results also fully confirm this point. As for the model testing process, the time cost of most models, including the proposed one, is less than one second, which can be generalized to real-time analysis of air traffic.



(a) DBSCAN (b) DAE-DBSCAN

(c) DAE-CAD

Fig. 8. t-SNE visualizations for clustering.

4.5. Performance analysis of flow pattern recognition

After detecting abnormal trajectories through the pre-training process of DAE-CAD, we use the remaining trajectories in I to perform its fine-tuning process to obtain a more cluster-friendly latent space and the cluster assignment of each trajectory. Fig. 7 presents the results of trajectory clustering, together with the abnormal trajectories corresponding to the optimal detection performance. It can be found that among all arrival flows in the Hong Kong terminal area, the two most dominant flow patterns come from the east, each exceeding 10% of the observations. As all observations were selected in June, one of the most active periods for convective meteorological conditions (e.g., thunderstorms) in Hong Kong, the flight routes are usually subject to great uncertainty. In this context, the proposed method treats nearly 25% of the observations as spatial anomalies, where a large number of holding patterns and rerouting patterns can be visualized intuitively from Fig. 7(b). In addition, Fig. 7(c) illustrates that flow patterns in the east, northeast, southwest, and southeast are simple, and each consists of two patterns according to different runway configurations. By contrast, flow patterns in the northwest are more complex and changeable. Note that a large number of incomplete trajectories from the southeast also form a flow pattern (i.e., Flow 0 in black). Although these trajectories are also anomalous in a sense, they are not isolated into S by the proposed algorithm due to relatively small reconstruction errors.

To further evaluate the clustering performance, we compare the proposed DAE-CAD with other clustering methods. Table 2 shows the optimal performance of each model, as well as the corresponding number of clusters and the proportion of outliers. It can be inferred that the DBSCAN-based methods have poor clustering quality and fewer clusters, which reflect the coarse-grained clustering results of such methods due to the uneven distribution of trajectory density. Preprocessing trajectories beforehand can bring benefits. In contrast, DEC and DAE-CAD obtain more fine-grained clustering results, both of which achieve their respective optimal performance when the number of clusters is 12. Although DEC does not assume the existence of abnormal trajectories, it still gets the second best performance. On this basis, DAE-CAD achieves a significant improvement, which once again confirms the effectiveness of removing abnormal trajectories before clustering.

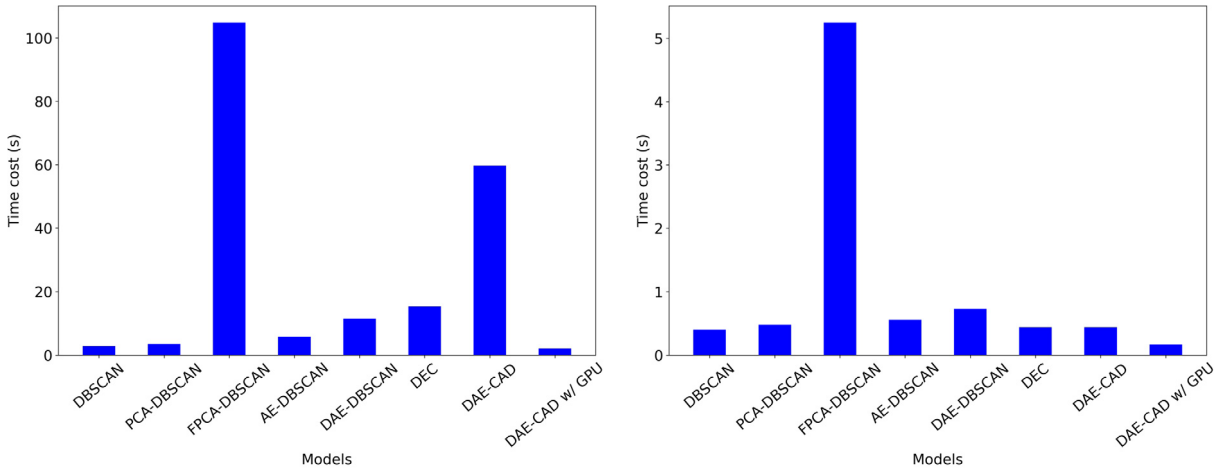
Table 2
Performance comparison of different models in flow pattern recognition.

Models	SI (%)	DBI (%)	Number of clusters	Percentage of outliers (%)
DBSCAN	58.33	63.93	7	31.47
PCA-DBSCAN	60.74	60.53	7	27.38
FPCA-DBSCAN	61.85	59.62	7	26.58
AE-DBSCAN	70.83	53.04	7	25.09
DAE-DBSCAN	73.20	44.37	8	22.25
DEC	76.63	38.06	12	0
DAE-CAD	**86.14**	**25.41**	12	24.48

The best results for the evaluation metrics are shown in bold.

Fig. 8 visualizes the latent space of three representative methods by t-SNE [34], one of the commonly used high-dimensional data visualization techniques. Among them, the latent space of DBSCAN is 400 dimensions (that is, the dimension of the original trajectory), and the latent space of DAE-DBSCAN and DAE-CAD is 50 dimensions. And different clusters are marked with different colors. From the perspective of separation and compactness, it can be clearly seen that the proposed DAE-CAD has a more friendly clustering space. Not only the low-dimensional latent space learned from DAE can improve the original space, but the clustering loss further moves towards better clustering performance.

Fig. 9 presents the time cost of model training and model testing in flow pattern recognition task, respectively. And the conclusion is similar to that of the abnormal trajectory detection task. Admittedly, the proposed DAE-CAD needs to detect abnormal trajectories beforehand, which requires additional execution time



(a) Model training (b) Model testing

Fig. 9. Time cost for flow pattern recognition.

than the DEC model. The whole framework can also be greatly accelerated by parallel computing. With the help of GPU, it outperforms all other models, both in model training and model testing.

5. Conclusions

In this paper, we considered the joint flow pattern recognition and abnormal trajectory detection problem and proposed an intelligent and end-to-end framework. In particular, it introduced two advanced regularization terms in the reconstruction-based deep autoencoder network, where $\ell_{2,1}$-norm regularization term was applied to generate low-dimensional outliers-free trajectory representation, and KL divergence term further transformed it into a cluster-friendly one. The effectiveness and efficiency of the framework were demonstrated for Hong Kong International Airport. At an acceptable time cost, it outperforms other mainstream methods in terms of anomaly detection performance and trajectory clustering quality. When using the GPU for parallel computing, it can be significantly accelerated to meet real-time requirements.

In fact, the proposed framework has a wide range of practical applications in air transportation. Identifying typical patterns and anomalous behaviors from massive historical trajectory data can serve as the core module of an automated surveillance system to monitor airspace health. Moreover, this knowledge can be further used to measure the complexity of air traffic to assess controller workload and inform decision-making for dynamic airspace configuration. Since air traffic is a safety-critical area, detecting and quantifying abnormal trajectories also provides opportunities for air traffic control experts to analyze potential risks and improve safety models from a smaller subset. In addition to its success with flight trajectory data, the framework can be flexibly extended to more general scenarios such as fault detection and health management related applications based on large-scale high-dimensional time-series data [40,41].

Future work will develop automated tools to monitor airspace in real time. Based on the identified abnormal trajectories and flow patterns, it will be a worthy topic to analyze the correlation between their evolution and specific influencing factors (such as weather conditions and controllers' actions [26]). It will also be interesting to explore more detailed underlying generation mechanisms (for example, asymmetric flight [42,43]) for abnormal trajectories.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work is partially supported by the National Natural Science Foundation of China (71731001), Postgraduate Research & Practice Innovation Program of Jiangsu Province (KYCX22_0373, KYCX22_0377), and China Scholarship Council (202106830077, 202106830100).

References

[1] M.C.R. Murça, Data-driven modeling of air traffic flows for advanced Air Traffic Management, Ph.D. thesis, Massachusetts Institute of Technology, 2018.
[2] X. Olive, J. Sun, A. Lafage, L. Basora, Detecting events in aircraft trajectories: rule-based and data-driven approaches, MDPI Proc. 59 (2020).
[3] G.N. Lui, T. Klein, R.P. Liem, Data-driven approach for aircraft arrival flow investigation at terminal maneuvering area, in: AIAA AVIATION 2020 FORUM, 2020, p. 2869.
[4] H. Wang, Z. Song, R. Wen, Y. Zhao, Study on evolution characteristics of air traffic situation complexity based on complex network theory, Aerosp. Sci. Technol. 58 (2016) 518–528.
[5] I. Gerdes, A. Temme, M. Schultz, Dynamic airspace sectorisation for flight-centric operations, Transp. Res., Part C, Emerg. Technol. 95 (2018) 460–480.
[6] M.C.R. Murça, M.X. Guterres, M. de Oliveira, J.B.T. Szenczuk, W.S.S. Souza, Characterizing the Brazilian airspace structure and air traffic performance via trajectory data analytics, J. Air Transp. Manag. 85 (2020) 101798.
[7] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al., A density-based algorithm for discovering clusters in large spatial databases with noise, in: kdd, vol. 96, 1996, pp. 226–231.
[8] M. Gariel, A.N. Srivastava, E. Feron, Trajectory clustering and an application to airspace monitoring, IEEE Trans. Intell. Transp. Syst. 12 (4) (2011) 1511–1524.
[9] M.C.R. Murça, R. DeLaura, R.J. Hansman, R. Jordan, T. Reynolds, H. Balakrishnan, Trajectory clustering and classification for characterization of air traffic flows, in: 16th AIAA Aviation Technology, Integration, and Operations Conference, 2016, p. 3760.
[10] X. Olive, J. Morio, Trajectory clustering of air traffic flows around airports, Aerosp. Sci. Technol. 84 (2019) 776–781.
[11] S.J. Corrado, T.G. Puranik, O.J. Pinon, D.N. Mavris, Trajectory Clustering Within the Terminal Airspace Utilizing a Weighted Distance Function, Multidisciplinary Digital Publishing Institute Proceedings, vol. 59, 2020, p. 7.
[12] Y. Liu, H. Wang, J. Fan, J. Wu, T. Wu, Control-oriented UAV highly feasible trajectory planning: a deep learning method, Aerosp. Sci. Technol. 110 (2021) 106435.
[13] L. He, N. Aouf, B. Song, Explainable deep reinforcement learning for UAV autonomous path planning, Aerosp. Sci. Technol. 118 (2021) 107052.

[14] Y. Lin, J.-w. Zhang, H. Liu, Deep learning based short-term air traffic flow prediction considering temporal–spatial correlation, Aerosp. Sci. Technol. 93 (2019) 105113.
[15] J. Du, M. Hu, W. Zhang, J. Yin, Finding similar historical scenarios for better understanding aircraft taxi time: a deep metric learning approach, IEEE Intell. Transp. Syst. Mag. (2022) 2–17, https://doi.org/10.1109/MITS.2021.3136329.
[16] C. Che, H. Wang, Q. Fu, X. Ni, Combining multiple deep learning algorithms for prognostic and health management of aircraft, Aerosp. Sci. Technol. 94 (2019) 105423.
[17] C. Zhou, R.C. Paffenroth, Anomaly detection with robust deep autoencoders, in: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2017, pp. 665–674.
[18] D.P. Kingma, M. Welling, Auto-encoding variational Bayes, arXiv preprint, arXiv: 1312.6114.
[19] M. Enriquez, Identifying temporally persistent flows in the terminal airspace via spectral clustering, in: Tenth USA/Europe Air Traffic Management Research and Development Seminar (ATM2013)/Federal Aviation Administration (FAA) and EUROCONTROL, Chicago, IL, USA, 2013, pp. 10–13.
[20] G. Andrienko, N. Andrienko, G. Fuchs, J.M.C. Garcia, Clustering trajectories by relevant parts for air traffic analysis, IEEE Trans. Vis. Comput. Graph. 24 (1) (2017) 34–44.
[21] X. Olive, L. Basora, B. Viry, R. Alligier, Deep trajectory clustering with autoencoders, in: Proceedings of the International Conference on Research in Air Transportation, 2020.
[22] L. Basora, X. Olive, T. Dubot, Recent advances in anomaly detection methods applied to aviation, Aerospace 6 (11) (2019) 117.
[23] G. Jarry, D. Delahaye, F. Nicol, E. Feron, Aircraft atypical approach detection using functional principal component analysis, J. Air Transp. Manag. 84 (2020) 101787.
[24] S.J. Corrado, T.G. Puranik, O.P. Fischer, D.N. Mavris, A clustering-based quantitative analysis of the interdependent relationship between spatial and energy anomalies in ADS-B trajectory data, Transp. Res., Part C, Emerg. Technol. 131 (2021) 103331.
[25] Y. Ji, L. Wang, W. Wu, H. Shao, Y. Feng, A method for LSTM-based trajectory modeling and abnormal trajectory detection, IEEE Access 8 (2020) 104063–104073.
[26] X. Olive, J. Grignard, T. Dubot, J. Saint-Lot, Detecting controllers' actions in past mode s data by autoencoder-based anomaly detection, in: SID 2018, 8th SESAR Innovation Days, 2018.
[27] X. Olive, L. Basora, Identifying anomalies in past en-route trajectories with clustering and anomaly detection methods, in: ATM Seminar 2019, 2019.
[28] S.J. Corrado, T.G. Puranik, O.J. Pinon-Fischer, D. Mavris, R. Rose, J. Williams, R. Heidary, Deep autoencoder for anomaly detection in terminal airspace operations, in: AIAA AVIATION 2021 FORUM, 2021, p. 2405.
[29] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol, L. Bottou, Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion, J. Mach. Learn. Res. 11 (2010).
[30] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016.
[31] P. Baldi, Autoencoders, unsupervised learning, and deep architectures, in: Proceedings of ICML Workshop on Unsupervised and Transfer Learning, JMLR Workshop and Conference Proceedings, 2012, pp. 37–49.
[32] F. Nie, H. Huang, X. Cai, C. Ding, Efficient and robust feature selection via joint L2, 1-norms minimization, Adv. Neural Inf. Process. Syst. 23 (2010).
[33] Y. Cong, J. Yuan, J. Liu, Sparse reconstruction cost for abnormal event detection, in: CVPR 2011, IEEE, 2011, pp. 3449–3456.
[34] L. Van der Maaten, G. Hinton, Visualizing data using t-SNE, J. Mach. Learn. Res. 9 (2008).
[35] J. Xie, R. Girshick, A. Farhadi, Unsupervised deep embedding for clustering analysis, in: International Conference on Machine Learning, PMLR, 2016, pp. 478–487.
[36] S. Boyd, S.P. Boyd, L. Vandenberghe, Convex Optimization, Cambridge University Press, 2004.
[37] M. Schäfer, M. Strohmeier, V. Lenders, I. Martinovic, M. Wilhelm, 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.
[38] X. Olive, Traffic, a toolbox for processing and analysing air traffic data, J. Open Sour. Softw. 4 (39) (2019) 1518-1.
[39] F.T. Liu, K.M. Ting, Z.-H. Zhou, Isolation-based anomaly detection, ACM Trans. Knowl. Discov. Data 6 (1) (2012) 1–39.
[40] M. Kordestani, M. Saif, M.E. Orchard, R. Razavi-Far, K. Khorasani, Failure prognosis and applications—a survey of recent literature, IEEE Trans. Reliab. 70 (2) (2019) 728–748.
[41] M. Kordestani, A.A. Safavi, M. Saif, Recent survey of large-scale systems: architectures, controller strategies, and industrial applications, IEEE Syst. J. 15 (4) (2021) 5440–5453.
[42] P. Stojakovic, B. Rasuo, Minimal safe speed of the asymmetrically loaded combat airplane, Aircr. Eng. Aerosp. Technol. (2016).
[43] P. Stojakovic, B. Rasuo, Single propeller airplane minimal flight speed based upon the lateral maneuver condition, Aerosp. Sci. Technol. 49 (2016) 239–249.