

Arrival Pattern Recognition and Prediction Based on Machine Learning

GUI Xuhao¹, ZHANG Junfeng^{1*}, TANG Xinmin¹, KANG Bo²

1. College of Civil Aviation, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, P. R. China;

2. CAAC Central and Southern Regional Air Traffic Management Bureau, Guangzhou 510403, P. R. China

(Received 1 April 2021; revised 28 May 2021; accepted 17 December 2021)

Abstract: A data-driven method for arrival pattern recognition and prediction is proposed to provide air traffic controllers (ATCOs) with decision support. For arrival pattern recognition, a clustering-based method is proposed to cluster arrival patterns by control intentions. For arrival pattern prediction, two predictors are trained to estimate the most possible command issued by the ATCOs in a particular traffic situation. Training the arrival pattern predictor could be regarded as building an ATCOs simulator. The simulator can assign an appropriate arrival pattern for each arrival aircraft, just like real ATCOs do. Therefore, the simulator is considered to be able to provide effective advice for part of the work of ATCOs. Finally, a case study is carried out and demonstrates that the convolutional neural network (CNN)-based predictor performs better than the random forest (RF)-based one.

Key words: air traffic management; decision support; arrival scheduling; deep learning; convolutional neural networks

CLC number: V355

Document code: A

Article ID: 1005-1120(2021)06-0927-10

0 Introduction

Rapid growth of air traffic and limited space-time resources have led to traffic congestion in terminal control areas (TMAs), resulting in flight delays, energy waste, and air pollution. Improving runway utilization and scheduling arrival aircraft will mitigate these problems. Arrival management (AMAN)^[1] is one particular tool in this field and has been widely used around the world.

AMAN could allocate the runway for arrival aircraft^[2], as well as schedule the arrival aircraft according to the optimization objectives and operational constraints^[3-4]. Recent studies on AMAN focus on model construction and algorithm design. For optimization objectives, Rosenthal et al. aimed to minimize the total cost^[5], i.e., the sum of delay cost. Takeichi minimized the operational cost that was defined as the linear sum of the fuel cost and time cost^[6]. Ji et al. optimized the scheduled time of the

last aircraft^[7]. Mokhtarimousavi et al. evaluated ground staffs' workload^[8]. Zhao et al. chose the average schedule time, the maximum flow time, and the maximum delay time as multiple objectives^[9]. For solution algorithms, Vadlamani et al. applied CPLEX to solve small-scale AMAN problems^[10]; Lieder et al. implemented dynamic programming to solve large-scale AMAN problems^[11]; and Sölvelling et al. deployed branch and bound to solve large-scale AMAN problem^[12].

In previous studies, AMAN only outputs the optimal landing runway and the scheduled time of arrival (STA). These cannot provide air traffic controllers (ATCOs) with advices on how to arrange arrival aircraft to achieve such STAs. In other words, they failed to notice the feasibility of optimal arrival scheduling, and ATCOs had to spend extra time finding out potential tactical commands to maintain the landing sequence and guarantee safe separation for arrival aircraft. The point merge sys-

*Corresponding author, E-mail address: zhangjunfeng@nuaa.edu.cn.

How to cite this article: GUI Xuhao, ZHANG Junfeng, TANG Xinmin, et al. Arrival pattern recognition and prediction based on machine learning[J]. Transactions of Nanjing University of Aeronautics and Astronautics, 2021, 38(6):927-936.

<http://dx.doi.org/10.16356/j.1005-1120.2021.06.003>

tem (PMS), initialized by Eurocontrol, could help ATCOs to meet the requirements of AMAN outputs^[13-14]. However, implementing PMS needs a large area of airspace, which will further worsen the lack of airspace resources in China^[15].

To address such issues, this paper aims to develop a decision support tool for arrival pattern assignment. Inspired by multi-step prediction^[16-17], this paper provides a data-driven method for arrival pattern recognition and prediction. The arrival patterns are those commonly used by the arrival aircraft in horizontal trajectories, which could reflect the clearance delivered by the ATCOs. Recognizing the arrival pattern could be viewed as identifying the control instruction (clearance delivered by the ATCOs). To predict the arrival pattern could be viewed as an estimate of the control intention, which means the most possible clearance will be de-

livered by the ATCOs in a particular traffic situation. Therefore, exploring the relationship between the arrival pattern and air traffic situation is the main task of this study.

The rest of this paper is organized as follows. The whole scheme for arrival pattern recognition and prediction is described in section 1. Section 2 provides a clustering-based arrival pattern recognition method. Two classification-based arrival pattern prediction methods are proposed in section 3. The results and discussion are illustrated in section 4, followed by concluding remarks in section 5.

1 Overview

The process of building the decision support model for ATCOs in TMA is shown in Fig.1. This framework includes data preprocessing, arrival pattern recognition, and arrival pattern prediction.

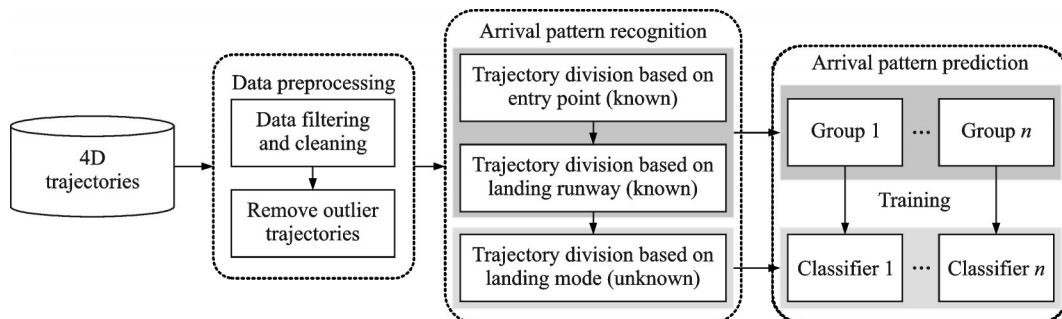


Fig.1 Illustration of building decision suggestion models

After the four-dimensional (4D) trajectories are decoded from the raw radar data, data preprocessing is applied to clean the data. Trajectories that neither pass the boundary nor land on runways are removed from the dataset since they are incomplete; and trajectory segments outside of the boundary are also filtered out.

After cleaning data and removing outliers from the raw data, arrival pattern recognition is applied to identify the arrival patterns step by step. Trajectories are divided into different groups according to entry fixes, landing runways, and landing modes. For the entry fix, once the aircraft enters the terminal area (TMA), we can capture its position, so the aircraft's entry fix is known. For the landing runway, since the runway selection is affected by its entry fix

and its stands assignment, and this is out of the scope of this paper, we assume that the landing runway is known. For the landing mode, aircraft fly differently in the TMA for different traffic situations. Therefore, the landing mode is unknown and needs to be recognized. In this work, we apply the trajectory clustering method to recognize the landing modes.

The task of arrival pattern prediction is to predict the landing mode that the aircraft is likely to adopt according to the traffic situation. Firstly, we divide trajectories into different groups according to their entry fixes and landing runways. Secondly, we build a classification model for each group to predict the aircraft's landing mode. When conducting the arrival pattern prediction for a new arrival aircraft, we choose the corresponding classification model ac-

cording to the entry fix and the landing runway.

2 Arrival Pattern Recognition

As an unsupervised method, trajectory clustering^[18] can group trajectories so that the trajectories in the same group share similar flight intentions. To cluster trajectories efficiently, a new clustering method is proposed that consists of a new form of trajectory representation and a step-by-step clustering framework.

For the new form of trajectory representation, a control-intention-oriented format is introduced to represent the trajectories. This format focuses on the heading of the aircraft rather than their geometric positions since ATCOs rely on radar vectoring for resolving conflicts and establishing the landing sequence.

The step-by-step clustering framework is introduced in this paper for not only mitigating information loss in the process of one-time similarity measurement but also reducing the calculation complexity of similarity measurement, thus the speed of clustering could be greatly improved.

2.1 Trajectory representation

Radar data used in this paper includes the recorded timestamp, and the aircraft position. Suppose there is a set of radar trajectories R , where $R = \{r_1, r_2, \dots, r_i, \dots, r_m\}$. Each trajectory r_i consists of a series of points: $r_i = (p_1^i, p_2^i, \dots, p_{L(i)}^i)$, where $L(i)$ represents the number of points of the trajectory r_i . Each record of trajectory r_i contains the time stamp t , heading ψ , latitude L_a , longitude L_o , and altitude A .

Heading and flight distance can represent control intention directly since ATCOs rely on radar vectoring for resolving conflicts and establishing the landing sequence. Therefore, this paper denotes trajectories by adjusted headings and distance-to-go (DTG) to capture more details about control intention^[19].

The way of representing trajectories by adjusted headings and DTG is described as follows.

Step 1 Find the difference in the heading between two consecutive points.

$$\Delta\psi_j^i = \psi_j^i - \psi_{j+1}^i \quad j=1, \dots, L(i)-1 \quad (1)$$

where j is the particular point of trajectory r_i , and $L(i)$ the number of points of the trajectory r_i .

Step 2 Modify $\Delta\psi$ according to the left- or right-turning

$$\Delta\psi_j^{i'} = \begin{cases} \Delta\psi_j^i - 360^\circ & \Delta\psi_j^i > 180^\circ \\ \Delta\psi_j^i + 360^\circ & \Delta\psi_j^i < -180^\circ \\ \Delta\psi_j^i & \text{Others} \end{cases} \quad (2)$$

where $\Delta\psi_j^{i'}$ is the modified value of $\Delta\psi_j^i$.

Step 3 Calculate the adjusted heading

$$\begin{cases} \theta_j^i = \psi_j^i & j = L(i) \\ \theta_j^i = \theta_{j+1}^i + \Delta\psi_j^{i'} & j = L(i) - 1, \dots, 2, 1 \end{cases} \quad (3)$$

Then, the trajectory is represented by a series of adjusted headings: $\theta_i = (\theta_1^i, \theta_2^i, \dots, \theta_{L(i)}^i)$, where θ_1^i is the initial adjusted heading of trajectory r_i and $\theta_{L(i)}^i$ the final adjusted heading (landing direction).

2.2 Step-by-step clustering framework

In this subsection, details of how to recognize arrival patterns are illustrated step by step.

Step 1 Eliminate trajectories with holding patterns. As shown in Fig.2, AC#4 has a holding pattern, and its adjusted heading has a continuous

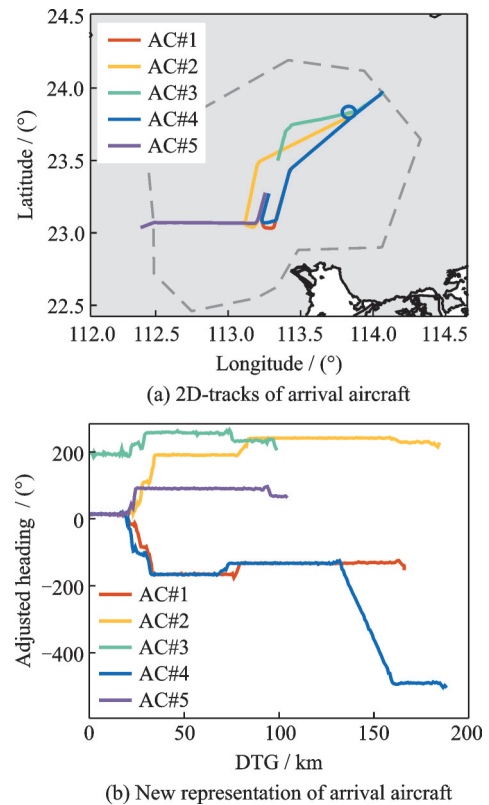


Fig.2 Trajectory clustering diagram

change when its DTG between 130 and 160 km. Therefore, AC#4 is separated from other aircraft.

Step 2 Divide trajectories into groups according to their entry fixes. As shown in Fig.2(a), according to the aircraft's entry fix, AC#5 can be separated from other aircraft.

Step 3 Divide trajectories into groups according to their landing runways.

After the above steps, only AC#1 and AC#2 are in the same group since both of them have the same entry fix and landing runway. Then, the initial adjusted heading (θ_i) of AC#1 and AC#2 can be used to distinguish AC#1 and AC#2 conveniently since they are different.

Further clustering can be conducted if necessary. For trajectories with the same entry fix, landing runway, and landing mode, the K-means algorithm can be used to cluster those trajectories directly. K-means assigns trajectories to exactly one of k clusters defined by centroids, where k is a pre-defined parameter. Further, squared Euclidean distance is adopted to measure the similarity between trajectories in this paper. However, such a measure requires that the number of trajectory points should be equal. Thereupon, it is necessary to resample the points of each trajectory by interpolation.

3 Arrival Pattern Prediction

Two different classification methods are adopted to predict the most likely command issued by the ATCOs in a particular traffic situation. The first one is based on random forest (RF)^[20], which requires efficient feature engineering to extract features from data. The second one is based on convolutional neural network (CNN)^[21], which could learn features from data automatically.

3.1 RF-based arrival pattern classification

Ensemble learning methods, such as RF, could achieve a better performance than other methods, both in Kaggle competitions and academic research. RF builds multiple decision trees and merges them to get a more accurate and stable prediction. Table 1 shows the feature set and the corresponding description for RF. The features, used in

previous studies^[22-23], are also included in this feature set.

The sine value and cosine value of the heading (#Fea1 and #Fea2) are used to replace the aircraft's heading. #Fea3 and #Fea4 are used to reflect the control intentions of ATCOs. Since the change of altitude and speed are the most common tactical commands. #Fea5 and #Fea6 are used to represent the position information of the aircraft. #Fea7 and #Fea8 are used to capture periodic information of TMA. The values of #Fea7 and #Fea8 range from 0 to 1. For example, while the arrival time of the aircraft is 8:30, #Fea7 = 0.354 ($8.5 \times 3600 / 86400 \approx 0.354$).

Table 1 Features for arrival pattern classification

Symbol	Description
#Fea1	Sine value of heading
#Fea2	Cosine value of heading
#Fea3	Altitude
#Fea4	Ground speed
#Fea5	Longitude of the aircraft
#Fea6	Latitude of the aircraft
#Fea7	The moment of entering
#Fea8	Weekly periodicity

3.2 CNN-based arrival pattern classification

3.2.1 Air traffic situation image construction

One of the contributions of this study is to extract features about the air traffic situation from images to predict arrival patterns. To this end, we construct the images to reflect the air traffic situations. The traffic situation image is composed of three channels, each channel is an image of 28×28 pixels, as shown in Fig.3.

Fig.3(a) presents the historical information layer, which records trajectories of all the arrival aircraft that land on the runway in the past 30 min. From the historical information layer, we can find the ATCOs' working habits and the current landing direction. The process is as follows: First, the TMA is converted into an image of 28×28 pixels; second, for each pixel in this image, if there is no landing aircraft that has passed this pixel in the past 30 min, the value of this pixel is set to 0, otherwise, set to 1.

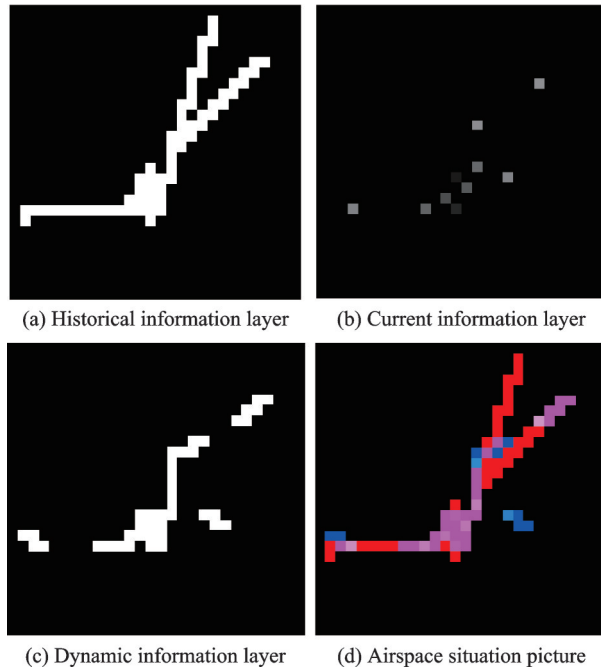


Fig.3 Traffic situation image

Fig.3(b) presents the current information layer with records of current positions of all the arrival aircraft within the TMA. From the current information layer, we can find the relative position of each aircraft. The process is as follows: First, the TMA is converted into an image of 28×28 pixels; second, for each pixel in this image, if there is no arrival aircraft, the value of this pixel is set to 0, otherwise, set to a number greater than 0 (when the aircraft

reaches higher altitude, the value is closer to 1). Since each pixel is a square with 7.14 km of each side. Therefore, two aircraft are unlikely to appear in the same pixel.

Fig.3(c) presents the dynamic information layer, which records the movement information of all the arrival aircraft within the TMA. From the dynamic information layer, we can find aircraft's movements in the past period. The specific process is as follows: Firstly, the TMA is converted into an image of 28×28 pixels; secondly, for each pixel in this image, if there is no arrival aircraft that has passed this pixel in the past 5 min, the value of this pixel is set to 0, otherwise, set to 1.

So far, we have defined three layers with different information, then we combine them into a color image, as shown in Fig.3(d). The red, the green, and the blue layers denote the historical information layer, the current information layer, and the dynamic information layer, respectively.

3.2.2 CNN configuration

CNN is used to extract features from a traffic situation image.

The CNN configuration proposed in this paper is simple, in order to improve the speed of training and prediction and reduce the risk of overfitting. Table 2 lists the CNN configuration used in this paper.

Table 2 Network configuration of CNN

Name	Input	Operation	Kernel	Output
Imageinput	—	Image Input	—	$28 \times 28 \times 3$
Conv	$28 \times 28 \times 3$	Convolution	Size : $3 \times 3 \times 3$ Number : 32 e : [1 1] Padding : [0 0 0 0]	$26 \times 26 \times 32$
Maxpool	$26 \times 26 \times 32$	Max pooling	Size : 2×2 Number : 1 e : [2 2] Padding : [0 0 0 0]	$13 \times 13 \times 32$
Fc	$13 \times 13 \times 32$	Fully connected	—	$1 \times 1 \times \text{LabelNum}$
Softmax	$1 \times 1 \times 2$	Softmax	—	$1 \times 1 \times \text{LabelNum}$
Classoutput	$1 \times 1 \times 2$	Classification output	—	$1 \times 1 \times 1$

3.3 Performance indicators

If there are only two arrival patterns in a group of trajectories, the problem of arrival pattern predic-

tion could be regarded as a binary classification problem. Precision, recall, and accuracy are used to measure the performance of the binary classifier.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (4)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (5)$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \quad (6)$$

where true-positive (TP) is the number of positive arrival patterns correctly predicted; false-positive (FP) the number of positive arrival patterns incorrectly predicted; true-negative (TN) the number of negative arrival patterns correctly predicted, and false-negative (FN) the number of negative arrival patterns incorrectly predicted.

If there are more than two arrival patterns in a group of trajectories, the problem of arrival pattern prediction could be viewed as a multi-classification problem. Purity and rand index (RI) are used to measure the performance of the multi-classifier.

$$\text{Purity}(\Omega, C) = \frac{1}{m} \sum_i |\omega_i \cap c_j| \quad (7)$$

$$\text{RI} = \frac{N_{\text{SS}} + N_{\text{DD}}}{N_{\text{SS}} + N_{\text{SD}} + N_{\text{DS}} + N_{\text{DD}}} \quad (8)$$

where Ω is the classification results $\Omega = \{\omega_i | i = 1, 2, \dots, K\}$; ω_i the i th class in the classification results; C the actual class $C = \{c_j | j = 1, 2, \dots, L\}$; c_j the j th label in the actual class; m the number of data; N_{SS} the number of pairs of arrival patterns in the same class in C and the same cluster in Ω ; N_{SD} the number of pairs of arrival patterns in the same class in C but not in the same cluster in Ω ; N_{DS} the number of pairs of arrival patterns in the same cluster in Ω but not in the same class in C ; and N_{DD} the number of pairs of arrival patterns in the different class in C and the different cluster in Ω .

4 Case Study

In this section, we first introduce the dataset and the procedure of data preprocessing. Then, we offer two arrival pattern recognition cases. The number of arrival patterns of those two cases are two (ATAGA-RWY01) and three (GYA-RWY19). Finally, we present the results of arrival pattern prediction. The two cases, ATAGA-RWY01 and GYA-RWY19, are used to verify the proposed classification method in both binary and multivariate

classification problems.

4.1 Data preparation

The radar data used in this paper is from Guangzhou Baiyun International Airport (ZGGG), where there are three parallel runways 01/19, 02L/20R, 02R/20L, and six entry fixes ATAGA, IGOONO, P270, IDUMA, GYA, P71. Fig.4 presents the heatmap of radar tracks with a two-month operation (November and December in 2019). In Fig.4, the research scope is defined as a green circle with the airport reference point as the center and the radius of 100 km. Most of the terminal areas are surrounded by the scope defined in this paper.

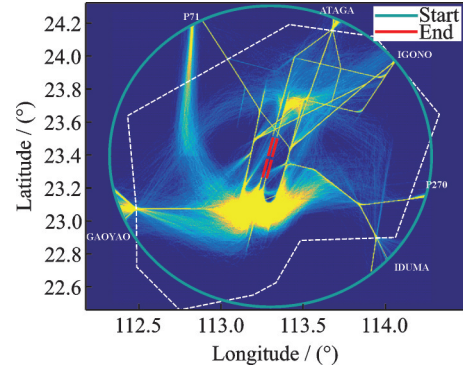


Fig.4 Heatmap image of radar tracks and entry fixes

The pseudo-code of preprocessing is described as follows.

```
[ValidTra] = Preprocessing(RadarData)
```

```
ValidTra = []
```

```
for  $i_{\text{flight}}$  in RadarData.ArrFlight:
```

```
     $i_{\text{flight}}$ .zip() #Eliminate invalidate or redundancy points
```

```
     $p_{\text{dist}} = \text{greatcircledistance}(i_{\text{flight}}, \text{AirportRefPoint})$ 
```

```
     $l_{1st100} = (p_{\text{dist}} > 100 \text{ km}).\text{index.head}(1)$ 
```

```
     $i_{\text{flight}}$ .remove( $l_{1st100}:\text{end}$ ) # Confirm start point
```

```
     $\text{infnl} = \text{isincuboid}(i_{\text{flight}}.\text{index.tail}(1))$ 
```

```
     $i_{\text{flight}}$ .remove( $1:\text{infnl}$ ) #Confirm end point
```

```
    if  $\sim(\text{empty}(l_{1st100})|\text{empty}(\text{infnl}))$ :
```

```
        ValidTra.append( $i_{\text{flight}}$ )
```

For each flight, we eliminated the invalidate or redundancy points. Then, we found the last point whose distance from airport reference point (ARP) was larger than 100 km and removed all the points

before that point. The above step ensured that all the following points were within the research scope. Next, we found the first point passed through the predefined cuboids and removed all the points after that point. The cuboids were predefined to determine whether the aircraft had joined the final. Finally, if a trajectory had both the start and the end-point, it was a valid trajectory.

After data preprocessing, there are 38 771 arrival flights from the original raw data.

4.2 Results of arrival pattern recognition

A step-by-step recognition method was adopted to recognize arrival patterns from the cleaned data.

Firstly, we divided trajectories into different groups according to their entry fixes. For example, Fig.5 presents the trajectories' entry from ATAGA, where the northbound trajectories are denoted by red while the southbound by blue.

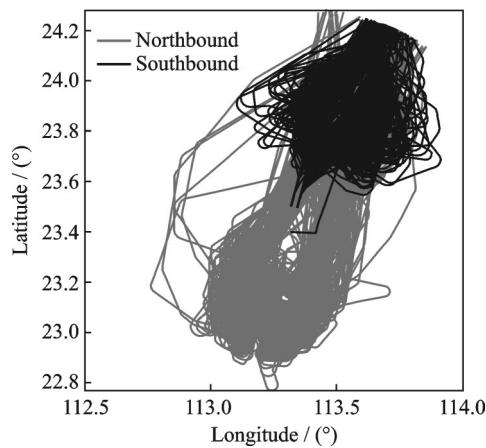


Fig.5 Trajectories entry from ATAGA

Secondly, for each group of trajectories with the same entry fix, we divided them into different groups based on the landing runways. Fig.6 displays the trajectories entered from ATAGA and landed on RWY01, where the right-turning trajectories are denoted by red while the left-turning by blue. However, these two kinds of trajectories, which reflect different control intentions, are difficult to separate, since a large part of them overlap. That is why the following step is needed.

Thirdly, for each group of trajectories with the same entry fix and the same landing runway, we di-

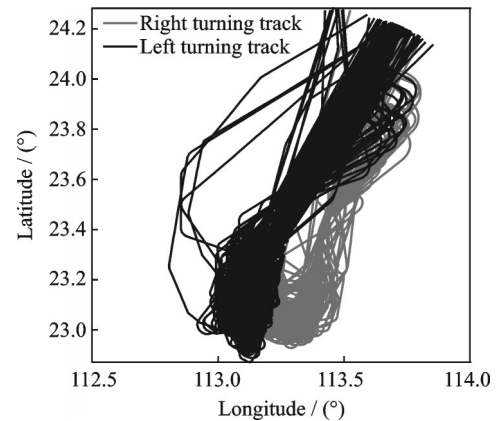


Fig.6 Trajectories via ATAGA to RWY01 (2D-tracks)

vided them into different groups according to their landing modes. To this end, we denoted trajectories by their DTG and adjusted heading. Then, the trajectories with different turning directions can be divided easily. As shown in Fig.7, trajectories with different landing modes are significantly different according to the new representation.

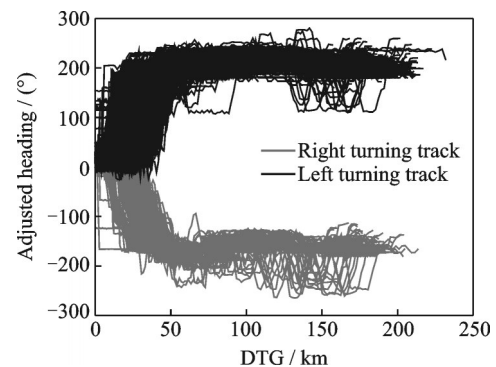


Fig.7 Trajectories via ATAGA to RWY01 (new representation)

Further clustering can be done if necessary. As shown in Fig. 8, although we got trajectories with

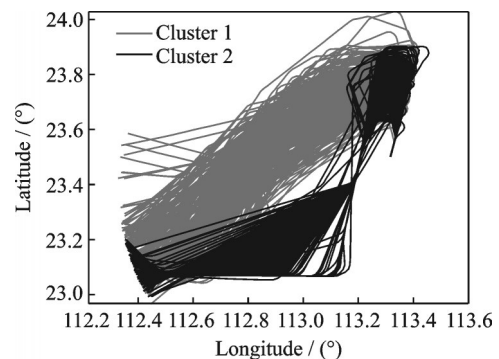


Fig.8 Clustering results of arrival trajectories via GYA to RWY19 (Right turning)

the same entry fix, the same landing runway, and the same landing mode, there were two different control intentions. To divide these trajectories into two groups, Euclidean distance was adopted to measure the similarity between trajectories, and K-means was used to assign these trajectories to exactly one of two clusters defined by centroids.

Fig.9 presents the final clustering results of the trajectories entered from GYA and landed on RWY19. Compared with trajectories in Fig.6 that entered from ATAGA and landed on RWY01, there are three different landing modes in the trajectories in Fig.8. This means the clearance delivered by the ATCOs for these trajectories can be more diverse than that for trajectories in Fig.6.

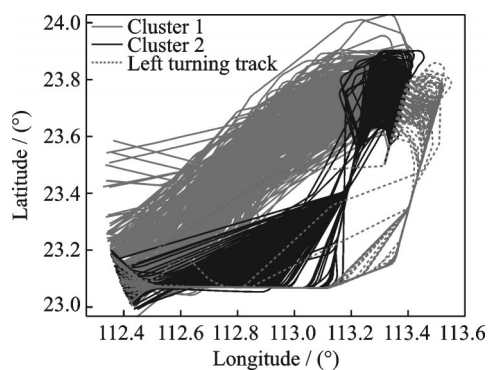


Fig.9 Clustering results of arrival trajectories via GYA to RWY19

4.3 Results of arrival pattern prediction

As mentioned above, for an aircraft when entering the TMA, its entry fix and landing runway are known, and its landing mode is unknown. Therefore, the task of arrival pattern prediction is to predict the landing mode that the aircraft is likely

to adopt. This paper uses data set ATAGA-RWY01 to refer to the trajectories in Fig.6 that entered from ATAGA and landed on RWY01, and data set GYA-RWY19 to refer to the trajectories in Fig.9 that entered from GYA and landed on RWY19.

Arrival pattern prediction for trajectories in ATAGA-RWY01 could be viewed as a binary classification problem since there are only two arrival patterns. Precision, recall, and accuracy were adopted to measure the performance of the predictor. Arrival pattern prediction for trajectories in GYA-RWY19 could be viewed as a multi-classification problem since there are three arrival patterns. Purity and RI were adopted to measure the performance of the predictor. The results of arrival pattern prediction are shown in Table 3.

For ATAGA-RWY01, the CNN-based arrival pattern predictor performed better than that of the RF-based arrival pattern predictor. Specifically, the CNN-based prediction precision for right-turning trajectories (93.8%) was higher than the RF-based one (72.1%). The CNN-based prediction recall for right-turning trajectories (91.0%) was higher than the RF-based one (75.6%). Besides, the CNN-based prediction accuracy (92.2%) was higher than the RF-based one (74.4%).

For GYA-RWY19, the CNN-based arrival pattern predictor performed better than that of the RF-based arrival pattern predictor. Specifically, the CNN-based prediction purity (89.1%) was higher than the RF-based one (60.9%). Besides, the RI value for the CNN-based predictor was 0.853, which was 0.288 larger than the RF-based predictor.

Table 3 RF-based arrival pattern prediction results

Dataset	Method	Prediction performance						
		Precision (right-turn)	Recall (right-turn)	Precision (left-turn)	Recall (left-turn)	Accuracy	Purity	RI
ATAGA-RWY01	CNN	0.938	0.910	0.907	0.936	0.922	—	—
	RF	0.721	0.756	0.767	0.733	0.744	—	—
GYA-RWY19	CNN	—	—	—	—	—	0.891	0.853
	RF	—	—	—	—	—	0.609	0.565

The accuracy in binary classification and the purity in multi-classification can be calculated in the

same way, so the two were comparable. By comparing the two data sets, we can see that as the number

of landing modes increased, the prediction performance decreased.

5 Conclusions

This paper aims to provide decision support for ATCOs when providing service for the arrival aircraft. One contribution of this work is to recognize control intentions from historical data, and the other contribution is to build a classification model to predict ATCOs' intentions.

First, this paper proposes an arrival pattern recognition method to divide arrival patterns hierarchically by control intentions. Compared with traditional trajectory clustering methods, dividing trajectories step-by-step can diminish the loss of trajectory information.

Second, this paper proposes two arrival pattern prediction algorithms to estimate the most likely commands issued by the ATCOs in a particular traffic situation. The case study of ZGGG proves that the CNN-based arrival pattern predictor performs better than that of the RF-based arrival pattern predictor (The prediction purity could achieve 89.1%). The reason is probably that the CNN-based predictor extracts more information about the traffic situation in the TMA. That means the traffic situation has a non-negligible effect on trajectory prediction in most cases. Therefore, this paper may be a reference for future trajectory prediction.

In this paper, we have to reduce the number of arrival patterns to ensure that the predictor could achieve satisfactory performance. However, the limited number of flight patterns means that it is impossible to provide detailed decision support for ATCOs. Therefore, increasing the number of arrival patterns while maintaining the prediction performance is our future work.

References

- [1] ITOH E, BROWN M, SENOGUCHI A, et al. Future arrival management collaborating with trajectory-based operations[C]//Proceedings of Electronic Navigation Research Institute Air Traffic Management and Systems II. Tokyo, Japan: Springer, 2017.
- [2] BRISKORN D, STOLLETZ R. Aircraft landing problems with aircraft classes[J]. *Journal of Scheduling*, 2014, 17(1): 31-45.
- [3] ZHANG J, ZHAO P, YANG C, et al. A new meta-heuristic approach for aircraft landing problem[J]. *Transactions of Nanjing University of Aeronautics and Astronautics*, 2020, 37(2): 197-208.
- [4] ZHANG J, ZHAO P, ZHANG Y, et al. Criteria selection and multi-objective optimization of aircraft landing problem[J]. *Journal of Air Transport Management*, 2020, 82: 101734.
- [5] ROSENTHAL E C, EISENSTEIN E C. A rescheduling and cost allocation mechanism for delayed arrivals[J]. *Computers & Operations Research*, 2016, 66: 20-28.
- [6] TAKEICHI N. Nominal flight time optimization for arrival time scheduling through estimation/resolution of delay accumulation[J]. *Transportation Research Part C*, 2017, 77: 433-443.
- [7] JI X, CAO X, DU W, et al. An evolutionary approach for dynamic single-runway arrival sequencing and scheduling problem[J]. *Soft Computing*, 2017, 21(2): 7021-7037.
- [8] MOKHTARIMOUSAVI S, RAHAMI H, KAVEH A. Multi-objective mathematical modeling of aircraft landing problem on a runway in static mode, scheduling and sequence determination using NSGA- II [J]. *International Journal of Optimization in Civil Engineering*, 2015, 5(1): 21-36.
- [9] ZHAO P, ZHANG J, YOU L. A composite dispatching rule-based method for multi-objective aircraft landing problem[C]//Proceedings of the 19th COTA International Conference of Transportation Professionals. Nanjing, China: COTA, 2019.
- [10] VADLAMANI S, HOSSEINI S, ZHANG Y, et al. A novel heuristic approach for solving aircraft landing problem with single runway[J]. *Journal of Air Transport Management*, 2014, 40: 144-148.
- [11] LIEDER A, BRISKORN D, STOLLETZ R. A dynamic programming approach for the aircraft landing problem with aircraft classes[J]. *European Journal of Operational Research*, 2015, 243(1): 61-69.
- [12] SÖLVELING G, CLARKE J. Scheduling of airport runway operations using stochastic branch and bound methods[J]. *Transportation Research Part C: Emerging Technologies*, 2014, 45(8): 119-137.
- [13] HONG Y, CHOI B, KIM Y, et al. Two-stage stochastic programming based on particle swarm optimization for aircraft sequencing and scheduling[J]. *IEEE Transactions on Intelligent Transportation Systems*, 2018, 20(4): 1-13.
- [14] LIANG M, DELAHAYE D, MARECHAL P. Con-

- flict-free arrival and departure trajectory planning for parallel runway with advanced point-merge system [J]. *Transportation Research*, 2018, 95: 207-227.
- [15] ZHANG J, YOU L, YANG C, et al. Arrival sequencing and scheduling based on multi-objective imperialist competitive algorithm[J]. *Acta Aeronautica et Astronautica Sinica*, 2021, 42(2): 324439.
- [16] MING W, BAO Y, HU Z, et al. Multistep-ahead air passengers traffic prediction with hybrid ARIMA-SVMs models[J]. *The Scientific World Journal*, 2014 (2014): 567246.
- [17] MIAO X, WU B, ZHOU Y, et al. Examining the impact of different periodic functions on short-term free-way travel time prediction approaches[J]. *Journal of Advanced Transportation*, 2020, 32(2): 1-15.
- [18] OLIVE X, MORIO J. Trajectory clustering of air traffic flows around airports[J]. *Aerospace Science and Technology*, 2019, 84: 776-781.
- [19] GUI X, ZHANG J, PENG Z. Trajectory clustering for arrival aircraft via new trajectory representation[J]. *Journal of Systems Engineering and Electronics*, 2021, 32(2): 473-486.
- [20] BREIMAN L. Random forest[J]. *Machine Learning*, 2001, 45: 5-32.
- [21] KRIZHEVSKY A, SUTSKEVER I, HINTON G. Imagenet classification with deep convolutional neural networks[C]//*Proceedings of Advances in Neural Information Processing Systems 25*. [S.l.]: Curran Associates Inc, 2012.
- [22] WANG Z, LIANG M, DELAHAYE D. A hybrid machine learning model for short-term estimated time of arrival prediction in terminal manoeuvring area[J]. *Transportation Research Part C: Emerging Technologies*, 2018, 95(7): 280-294.
- [23] ZENG W, QUAN Z, ZHAO Z, et al. A deep learning approach for aircraft trajectory prediction in terminal airspace[J]. *IEEE Access*, 2020(8): 151250-151266.

Acknowledgements This work was supported by the National Natural Science Foundation of China (Nos. U1933117, 61773202, 52072174).

Authors Mr. GUI Xuhao received the B. S. degree at transportation (air traffic management) from Civil Aviation University of China in 2018. He is currently a postgraduate in Nanjing University of Aeronautics and Astronautics. His research interests are air traffic, management and machine learning.

Dr. ZHANG Junfeng received his B.S. degree at automated industry from Nantong University in 2002 and Ph.D. degree in control theory and control engineering from Nanjing University of Aeronautics and Astronautics in 2008. He joined Nanjing University of Aeronautics and Astronautics in July 2008, where he is currently an associate professor in the College of Civil Aviation. His research interests include (1) 4D trajectory generation, prediction and optimization; (2) decision support tool (DST) for the air traffic control, like AMAN and DMAN.

Author contributions Mr. GUI Xuhao designed the study, compiled the models, conducted the analysis, interpreted the results and wrote the manuscript. Dr. ZHANG Junfeng contributed to the data and model components for flight pattern classification models. Prof. TANG Xinmin interpreted the results. Mr. KANG Bo provided the discussions and background of the research. All authors commented on the manuscript draft and approved the submission.

Competing interests The authors declare no competing interests.

(Production Editor: ZHANG Bei)

基于机器学习的进场模式识别与预测

归旭豪¹, 张军峰¹, 汤新民¹, 康博²

(1. 南京航空航天大学民航学院, 南京 211106, 中国; 2. 民用航空中南地区空中交通管理局, 广州 510403, 中国)

摘要:为了给空中交通管制员提供更好的决策支持,提出了一种数据驱动的进场飞行模式识别与预测方法。对于进场飞行模式识别,提出了一种基于聚类的方法用于发现不同的管制意图。对于飞行模式预测,分别训练了两种预测器用于估计在特定的交通态势下,管制员将会发布的指令。其中对于进场模式的预测器可以当作是空中交通管制员模拟器,其可以像管制员一样为进场的航空器指定合适的进场模式。因此,该模拟器被认为可以为管制员提供有效的建议。最后,通过案例研究证明了基于卷积神经网络的预测器性能优于基于随机森林的预测器。

关键词:空中交通管理;决策支持;进场调度;深度学习;卷积神经网络