

# Aircraft Landing Time Prediction with Deep Learning on Trajectory Images

Liping Huang, Sheng Zhang, Yicheng Zhang, Yi Zhang, Yifang Yin  
Institute for Infocomm Research, Agency for Science, Technology and Research, Singapore

**Abstract**—Aircraft landing time (ALT) prediction is crucial for air traffic management, especially for arrival aircraft sequencing on the runway. In this study, a trajectory image-based deep learning method is proposed to predict ALTs for the aircraft entering the research airspace that covers the Terminal Maneuvering Area (TMA). Specifically, the trajectories of all airborne arrival aircraft within the temporal capture window are used to generate an image with the target aircraft trajectory labeled as red and all background aircraft trajectory labeled as blue. The trajectory images contain various information, including the aircraft position, speed, heading, relative distances, and arrival traffic flows. It enables us to use state-of-the-art deep convolution neural networks for ALT modeling. We also use real-time runway usage obtained from the trajectory data and the external information such as aircraft types and weather conditions as additional inputs. Moreover, a convolution neural network (CNN) based module is designed for automatic holding-related featurizing, which takes the trajectory images, the leading aircraft holding status, and their time and speed gap at the research airspace boundary as its inputs. Its output is further fed into the final end-to-end ALT prediction. The proposed ALT prediction approach is applied to Singapore Changi Airport (ICAO Code: WSSS) using one-month Automatic Dependent Surveillance-Broadcast (ADS-B) data from November 1 to November 30, 2022. Experimental results show that by integrating the holding featurization, we can reduce the mean absolute error (MAE) from 82.23 seconds to 43.96 seconds, and achieve an average accuracy of 96.1%, with 79.4% of the predictions errors being less than 60 seconds.

**Keywords**—Air Traffic Management, Aircraft Landing Time, Trajectory Image, Convolution Neural Networks.

## I. INTRODUCTION

The aircraft landing time (ALT) is the time when an aircraft touches down on the runway [1]. The ALT and its prediction are of great significance for many stakeholders and purposes. For example, airlines can inform passengers about the estimated arrival time. Airports and flight operators can use the ALT information to plan the support services for inbound flights, such as parking, fueling and loading [2]. Furthermore, accurate predictions of ALT are essential for optimizing arrival aircraft schedules [3]. The arrival manager (AMAN) systems depend on ALT estimation to sequence the runway slot efficiently [4], and to provide precise information (e.g., time to gain, time to lose) for air traffic controllers (ATCOs) to conduct speed control [5].

Given the importance of ALT and its prediction, various data-driven machine learning-based approaches have been developed for the ALT prediction. The most commonly used data is the trajectory data because aircraft trajectories of airborne

flights contain plenty of information for real-time air traffic. The aircraft trajectories are helpful to extract the aircraft location, speed, traffic context, relative spacing between aircraft, etc [6]. The study in [7] uses airborne trajectories in the extended terminal maneuver airspace of WSSS to obtain the latitude, longitude, altitude, speed, rate of climb, heading of the aircraft and the entry zone of an aircraft for ALT prediction. The study in [8] explores another set of features from the trajectories for modeling ALTs, including the calculated arrival pressure, and the calculated sequencing pressure. Meanwhile, researchers in [9] obtain the traffic densities at different granularities from the trajectory data. Though various information from aircraft trajectories is extracted for the ALT prediction, these studies do not involve aircraft holding stages, which makes the ALT prediction more challenging.

The most commonly applied machine learning methods for ALT prediction are the decision tree-based models [9], [10]. For instance, the study in [9] tested several decision tree-based machine learning models for the ALT predictions, including random forest and the gradient boosting machine(GBM). The performance of these kind of models depends on the features, which requires amounts of feature engineering work. Data missing may make the models lose efficacy since missing data is removed as in [11]. Besides, manual feature engineering-based methods [12] depend on the data quality and human knowledge of the research problem. Conversely, deep learning-based methods can help capture the unseen patterns of the data. The research in [13] proposes a deep learning-based method for ALT prediction, where the time series-based deep learning model, LSTM, is used on the non-linear (tanh) combination of the raw trajectory's latitude, longitude, altitude, and speed. Unlike CNNs, the time-series based model cannot directly capture the spatial context among aircraft in TMA, e.g., the distances between aircraft.

In this study, we propose using trajectory images instead of raw aircraft latitude, longitude, speed, heading information for ALT modeling. Specifically, we define a TMA research airspace circle (TRC) that is centered around the airport and covers the TMA for the problem formulation. For the case study of Singapore Changi Airport, the TRC has a radius of 50 nautical miles (NM), which is sufficient to cover the TMA of the airport since its TMA is located at a distance of 40NM from the airport. For each target aircraft that arrives at TRC, we generate a trajectory image with the target aircraft trajectory colored in red and the background aircraft trajectory

TABLE I. ACRONYM (VARIABLE) AND CORRESPONDING DEFINITION

Acronym (Variable)	Definition	Description
TRC	TMA Research Airspace Circle	50 NM to Changi Airport
TBX	TMA Extended Boundary	60 NM to Changi Airport
TBE	between TRC and TBX	flying distance is around 10 NM
$\mathcal{T}_{\text{TRC}}^j$	timestamp when an aircraft $j$ arrives at TRC	calculated from ADS-B data
$\mathcal{T}_{\text{THR}}^j$	timestamp when an aircraft $j$ arrives at the runway threshold	calculated from ADS-B data
$\mathcal{T}_{\text{TRC}}^{\mathcal{L}(j)}$	timestamp of leading aircraft of $j$ arrives at TRC	$\mathcal{L}(j)$ is the leading one of $j$ when $j$ arrive at TRC
$t_j$	landing time for aircraft $j$ as $\mathcal{T}_{\text{THR}}^j - \mathcal{T}_{\text{TRC}}^j$	target prediction variable
$v_E^j$	mean speed of aircraft $j$ within TBE	in knots, calculated from ADS-B
$v_E^{\mathcal{L}(j)}$	speed of the leading aircraft of $j$ within TBE	in knots, calculated from ADS-B
$\bar{v}_E$	the average $v_E^j$ values for all aircraft within TBE	in knots, calculated from ADS-B
$\tau$	trajectory image capture window with size $\tau$ seconds	time window as $[\mathcal{T}_{\text{TRC}}^j - \tau, \mathcal{T}_{\text{TRC}}^j]$
$\delta$	tabular feature capture window with size $\delta$ seconds	time window as $[\mathcal{T}_{\text{TRC}}^j - \delta, \mathcal{T}_{\text{TRC}}^j]$
$X_j^{\tau_j}$	generated trajectory image for aircraft $j$ in capture window $\tau$	used to represent TMA context
$X_j^{\delta_j}$	tabular features for aircraft $j$ captured in window $\delta$	runway usage, weather, aircraft type, etc.

colored in blue. The trajectory images inherently contain the aircraft's location, speed, heading, relative distance, entry zone, and arrival traffic flow in the TRC airspace. Based on the generated trajectory images, we design an end-to-end deep learning model that leverages state-of-the-art convolution neural networks for predicting aircraft landing times. Contributions of this study are summarized as follows:

- Trajectory images are proposed to be generated and utilized in the ALT modeling. These images can capture the aircraft's position, speed, heading, entry zone, arrival traffic flow, and relative distances among airborne flights. The trajectory image generation can reduce our major efforts in the complicated feature engineering process and enable us to apply state-of-the-art convolution neural networks for the ALT modeling problem.
- A CNN-based module is designed for the automatic holding-related featurization. Except for trajectory images, this module also takes the holding status of the leading aircraft, the time/speed gap at TRC with the leading aircraft, and the speed variation at TRC as its inputs. Its output is further fed into the final ALT prediction to make it capable of capturing the holding-related features, and hence an end-to-end ALT prediction model is proposed.
- The proposed ALT prediction approach has been testified with the case study of Singapore Changi Airport, where one-month ADS-B data from November 1 to November 30, 2022 has been used. Experimental results show that incorporating the holding featurization module reduced the mean absolute error (MAE) from 82.23 seconds to 43.96 seconds. The bad prediction ratio dropped by 1.31 percent. Furthermore, 79.4% of the prediction errors are less than 60 seconds.

The rest of this paper is organized as follows. Section II presents the methodology. Section III presents the experimental evaluations and corresponding analysis of the prediction accuracy. Conclusions and discussion for future work are finally drawn in Section IV.

## II. METHODOLOGY

### A. Problem Statement

For clear and concise expression, acronyms, variables, and corresponding definition used in this study are summarized in Table I. In this study, we focus on predicting the landing time of each inbound aircraft when it arrives at TRC of Singapore Changi Airport. The landing time  $t_j$  for an aircraft  $j$  is the time calculated as  $t_j = \mathcal{T}_{\text{THR}}^j - \mathcal{T}_{\text{TRC}}^j$ , where  $\mathcal{T}_{\text{THR}}^j$  and  $\mathcal{T}_{\text{TRC}}^j$  are respectively the timestamps when the aircraft  $j$  arrives at the runway threshold and the TRC (50NM to the airport). Both timestamps are extracted from aircraft trajectory data. For predicting  $t_j$ , we intend to construct a mapping function  $t_j = \mathcal{F}(X_j^{\tau, \delta})$ . In this study, we utilize deep neural networks to construct the mapping function, and the input feature  $X_j^{\tau, \delta}$  is a combination of the generated trajectory image  $X_j^{\tau_j}$  within the capture window  $[\mathcal{T}_{\text{TRC}}^j - \tau, \mathcal{T}_{\text{TRC}}^j]$ , and the tabular features  $X_j^{\delta_j}$  captured in the time window  $[\mathcal{T}_{\text{TRC}}^j - \delta, \mathcal{T}_{\text{TRC}}^j]$ . Hence, the objective is to construct the mapping function

$$t_j = \mathcal{F}(X_j^{\tau_j}, X_j^{\delta_j}), j \in \mathcal{J} \quad (1)$$

where  $X_j^{\tau_j}, X_j^{\delta_j}$  are respectively the trajectory image and tabular features for aircraft  $j$ . In the following subsections, we elaborate on the methodology overview, dataset description, data analysis, feature engineering (including the trajectory image generation and the related tabular feature extraction), holding featurization, and deep convolution neural networks for the proposed approach.

### B. Methodology Overview

The framework for the proposed aircraft landing time prediction method is described in Fig. 1. The ALT prediction takes the generated trajectory image, the aircraft information, runway threshold usage, weather condition, and the output from the Holding Featurization as its inputs. Historical trajectories of aircraft are used to generate the trajectory images and extract the runway arrivals, which are the inputs for both the Holding Featurization and the ALT Prediction. Additionally,

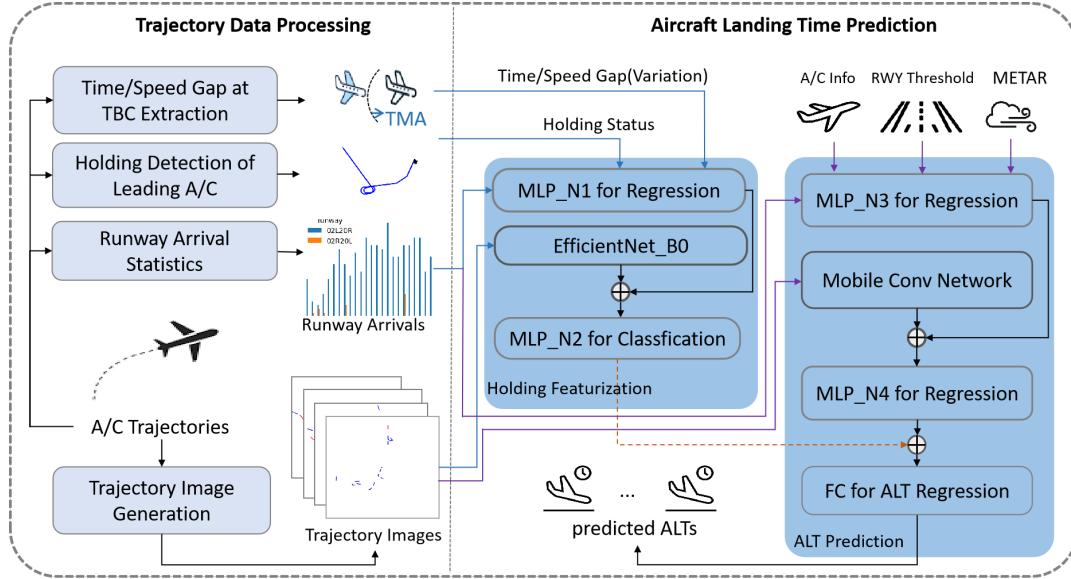


Figure 1. Overview of the Proposed Method.

the leading aircraft holding status, the time gap and speed gap with the leading aircraft at TRC, and the speed variation at TRC are used as the inputs of the Holding Featurization.

### C. Dataset

1) *Trajectory Data*: The trajectory data source of this study is Automatic Dependent Surveillance-Broadcast (ADS-B) flight data. One-month ADS-B data from November 1 to November 30, 2022, have been used in this paper. Even though the ADS-B dataset is of high quality, where the data sampling granularity is typically one positioning point per second, some of the data segments have slight missing points, hence a quadratic interpolation is used to impute the missing points for both latitude and longitude. By matching the imputed ADS-B positioning points to Changi airport runways, we extract the arrival aircraft data and get the corresponding landing runway for each aircraft. After removing the outliers (outside 3 times standard deviation to the mean value), we have 8396 aircraft for this study. Since the TRC is 50NM from the airport, the longitude range is (103.0, 105.0) and the latitude range is (0.5, 2.25). Fig. 2 shows the one day's trajectories of arrival flights with the distance circles.

2) *Meteorological Data*: the meteorological data (METAR) for Singapore Changi airport station are obtained from Iowa Environmental Mesonet from Iowa State University [14], where the wind direction, wind speed in knots, visibility in miles, wind gust in knots, sky coverage and the altitude in feet are provided.

3) *Flight Plan and Online Aircraft Performance Database*: Flight Plan (FPL) and online aircraft performance database (APD) [15] are used to obtain the aircraft RECAT-EU. Each record in the FPL contains the aircraft model type (e.g., A320), and the online APD provides the mapping from model type to the RECAT-EU. The aircraft are classified into six

categories of RECAT-EU, namely, Light, Lower Medium, Upper Medium, Lower Heavy, Upper Heavy, and Supper Heavy. Our dataset contains five of them, as "light" aircraft did not land at Singapore Changi Airport in November 2022.

### D. Data Analysis

1) *Landing Time Analysis*: As described in the problem statement section, the landing time  $t_j$  for aircraft  $j$  is calculated as the time gap between the timestamp for landing at the runway threshold  $T_{THR}^j$ , and the timestamp  $T_{TRC}^j$  when the aircraft arrives at TRC, where both timestamps are extracted from the ADS-B trajectory data. The landing time distribution

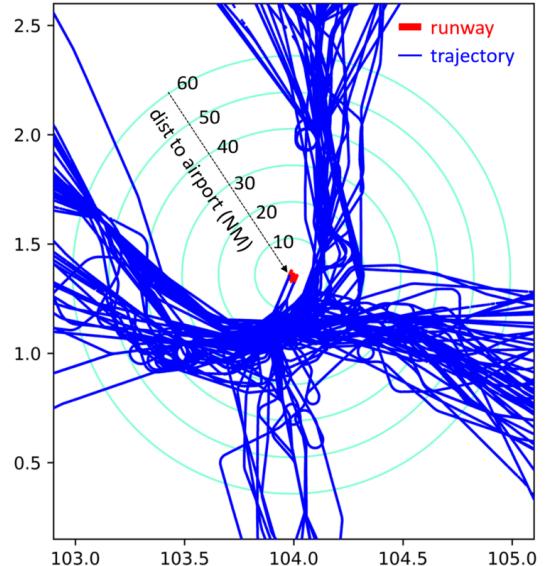


Figure 2. Arrival Aircraft Trajectories in One Day for Changi Airport.

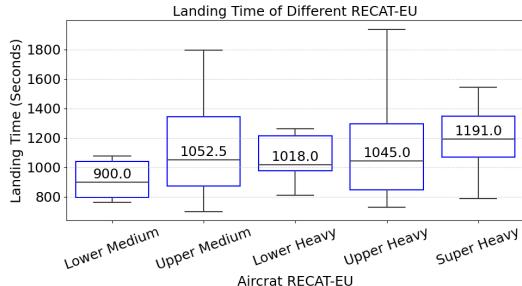


Figure 3. Aircraft Landing Time vs RECAT-EU.

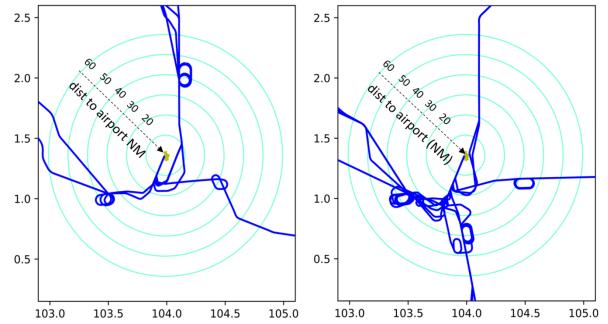


Figure 5. Trajectories with Holdings in Two Days.

for all aircraft categories during a day is shown in Fig. 3. It indicates that "Super Heavy" aircraft tends to have longer landing times, and "Lower Medium" aircraft tends to have the shorter landing times. Hence, it's essential to involve the RECAT-EU of an aircraft in the ALT prediction modeling.

2) *Runway Threshold Usage for Aircraft Landing:* Specific to the runway 02L20R, the runway threshold used during a day can be changed due to the runway operation (e.g., due to wind direction change, etc.). The runway thresholds for the physical runway 02L20R are 02L and 20R. The actual runway threshold usage for aircraft landing in two days are shown in Fig. 4. The figure shows that the runway threshold 02L is used before 12:00 on Day 1 and changed to 20R at 12:00. On Day 2, the runway is closed during the early morning and changed from 02L to 20R between 12:00-13:00. The runway threshold changes for landing can cause extra waiting time for aircraft in the TMA. Accordingly, we use a runway change label (0/1) in the ALT modeling.

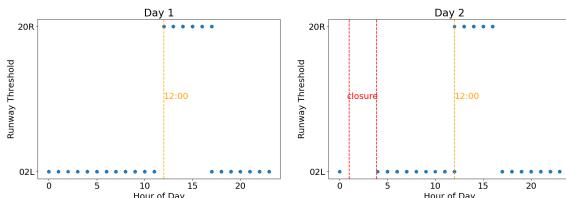


Figure 4. Runway Threshold Usage in Two Days.

3) *Holding Detection and Impact on ALT:* An aircraft may have to enter a holding stage due to the air traffic control, which makes the landing time prediction more difficult. In this study, we propose to integrate the holding potentials of each aircraft for the aircraft landing time prediction. In this section, we will analyze the aircraft holdings and their effect on the ALTs. The holding detection is based on the ADS-B data [16]. The trajectories with holdings in two days are shown in Fig. 5. We can note that arrival flights can enter the TMA of Changi airport from four directions, and in each direction, flights may enter a holding stack to follow the air traffic control. The holdings mostly happen between 30NM and 50NM to the airport, and we set the TRC as 50NM to cover these holdings.

To illustrate the effect of holdings on the ALTs, we compare

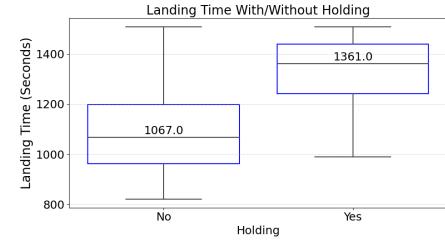


Figure 6. Aircraft Landing Times vs Holdings.

the ALTs for aircraft with and without holdings in Fig. 6. It shows that flights with holdings have higher ALTs than the flights without holdings. For this reason, we propose to utilize a module to capture the holding probability-related features for the ALT prediction. The following subsection presents the input features for the proposed approach.

#### E. Feature Engineering

1) *Factors and Representation:* As we analyzed in the previous subsection, various factors can affect  $t_j$ , such as the TMA traffic context, the weather conditions, the runway settings and operations, as well as the control intent for safety guarantees. In this study, we explore how to integrate these factors for predicting the aircraft landing time  $t_j$ . We summarize the potential infecting factors in Table II. Three categories of features are used: the TMA context, the runway operation, and the external information. The TMA context is represented as the generated trajectory image, which is generated from the ADS-B data to capture the aircraft position, speed, heading, entry zone, and the TMA arrival traffic. We will describe each type of features in the follow subsections.

2) *Trajectory Image Generation:* In this study, we generate the trajectory image for representing the TMA context. Given the capture window size  $\tau$ , we generate the trajectory image for an aircraft  $j$  when it arrives at TRC. The track position points for all aircraft inside TRC and within the time window  $[\mathcal{T}_{\text{TRC}}^j - \tau, \mathcal{T}_{\text{TRC}}^j]$  are collected to plot the image. Specifically, the trajectory of the target aircraft  $j$  are marked in red, and all other planes' trajectories are marked in blue.

Fig. 7 shows examples of the generated images for aircraft landing from four different directions to Singapore Changi Airport. The image generation is based on the ADS-B data,

TABLE II. FEATURES FOR ALTS AND THEIR REPRESENTATIONS.

Indicator	Description	Representation	Feature Category	Data Source
aircraft position	pixel positions in the image	trajectory image	TMA Context	ADS-B
aircraft speed	trajectory length in the image			
aircraft heading	tail to head direction			
entry zone	pixel positions of the target aircraft			
arrival traffic	number of trajectories in the image			
dynamic runway setting	runway threshold change label	0/1		
runway usage	arrivals on each runway	real number	Runway Operation	METAR
Weather	drct, sknt, gust, vsby, sky1l, skyc1	real number (in knots), 0/1		
seasonality	is_peakhour, is_weekday	0/1		
A/C info	Aircraft RECAT-EU	0,1,2,3,4,5	External Info	Flight Plan/APD

Figure 7. Examples of Generated Trajectory Images for Aircraft Landing in Four Different Directions,  $\tau = 60$ ,  $\delta = 10$ .

which provides one point per second for each aircraft (missing data points are imputed with interpolation). For a given  $\tau$  value, all aircraft in one image have the same flying time, so the relative length of the trajectory indicates the aircraft speed. That is, longer trajectory in one image means a higher speed of the aircraft. Moreover, the aircraft heading is determined by the direction from the last point to the first point of the trajectory, where the first point is usually closer to the runway. By generating such a trajectory image, we can capture several kinds of TMA context simultaneously in the image as described in Table. II: the aircraft position, speed, heading, the entry zone, relative distances. Additionally, the number of trajectories in the image directly represents the arrival traffic flow in the capture window.

3) *Runway Operation Feature*: We use the ADS-B data to determine the landing runway and runway threshold for each aircraft. We also count the number of arrival aircraft for each runway within the tabular feature capture window  $[\mathcal{T}_{\text{TRC}}^j - \delta, \mathcal{T}_{\text{TRC}}^j]$ . Furthermore, we set the runway threshold change label to 1 if the runway threshold is changed during the capture window  $[\mathcal{T}_{\text{TRC}}^j - \delta, \mathcal{T}_{\text{TRC}}^j]$ , and to 0 otherwise.

4) *External Info*: We obtain the aircraft model type, e.g., A320, from the flight plan dataset and use it to get the corresponding model RECAT-EU type, e.g., Upper Medium, from the online APD. Then we encode the RECATEU type

as a number from 0 to 5 (0 for light, 5 for super heavy), and use it as one dimension of the input feature for  $X_j^{\delta_j}$ . For the weather-related features, we utilize the wind direction (drct), wind speed (sknt), visibility (vsby) in miles, wind gust in knots, sky level 1 altitude in feet, and the sky level 1 coverage, which are all from the METAR dataset. We extract all these features, as well as the seasonality features, within the capture window  $[\mathcal{T}_{\text{TRC}}^j - \delta, \mathcal{T}_{\text{TRC}}^j]$ , and each is set as one dimension of  $X_j^{\delta_j}$  for the ALT modeling.

#### F. Holding Featurization

As shown in Fig. 1, we design a module to estimate the holding potential of an aircraft for the ALT prediction. The Holding Featurization module takes trajectory images and other tabular features as its input. In this section, we describe the tabular features for the Holding Featurization module. These include the runway arrivals that are also the input for the ALT Prediction module. Besides, the time gap and speed of an aircraft with its leading aircraft at TRC, and the variation of the speed compared to the average speed within TBE are used. As Fig. 7c illustrates, the target aircraft's leading aircraft enters the holding status, which have a high probability of making the target aircraft also enter the holding stack. We explain these features in detail below.

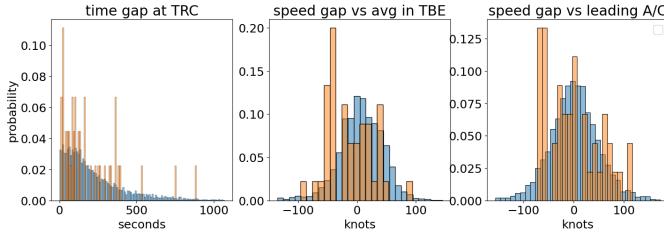


Figure 8. Statistical probability of time gap at TRC, speed gap vs average (avg) speed in TBE, and the speed gap vs its leading aircraft's speed in TBE. Blue: without holding, Orange: with holding .

- the arrivals on runways: the number of aircraft entering holding stage increases when the the runways are busy. During this period, parallel landing can be conducted.
- The time gap  $\Delta\mathcal{T}^{j,\mathcal{L}(j)} = \mathcal{T}_{\text{TRC}}^j - \mathcal{T}_{\text{TRC}}^{\mathcal{L}(j)}$ , denoting the the time gap between the target and its leading aircraft arriving at the TRC, here  $\mathcal{L}(j)$  is the leading aircraft of aircraft  $j$  when aircraft  $j$  arrives at TRC;
- the speed variation  $\Delta v_{\text{E}}^{j,\text{avg}} = v_{\text{E}}^j - \bar{v}_{\text{E}}$ , which is the gap between the speed of aircraft  $j$  with the average speed of all aircraft in TBE, where  $\bar{v}_{\text{E}}$  is the average speed of all aircraft in TBE.
- The speed gap  $\Delta v_{\text{E}}^{j,\mathcal{L}(j)} = v_{\text{E}}^j - v_{\text{E}}^{\mathcal{L}(j)}$ , which representing the speed difference between aircraft  $j$  and its leading aircraft  $\mathcal{L}(j)$  within the 10 NM in TBE.
- the holding status of the leading aircraft  $\mathcal{L}(j)$ , which is a 0/1 label.

The statistical probability of  $\Delta\mathcal{T}^{j,\mathcal{L}(j)}$ ,  $\Delta v_{\text{E}}^{j,\text{avg}}$ , and  $\Delta v_{\text{E}}^{j,\mathcal{L}(j)}$  are shown in Fig. 8, which is subject to the with holding and without holding aircraft. It indicates that holding aircraft tend to have smaller values in all three features. Hence, these features are extracted and set as the input for the holding featurization module. Besides, the holding status of a target aircraft's leading aircraft (0/1) is set as an additional input.

#### G. Deep Convolution Networks

1) *Mobile Convolution Network for ALT Prediction:* The MobileNetV2 has better efficiency since it utilizes depthwise separable convolutions, which are the key building blocks for many efficient neural network architectures [17]. Assume that the dimension of the input tensor  $L_i$  in layer  $i$  is  $h_i \times w_i \times d_i$ , the dimension for the output tensor  $L_j$  is  $h_j \times w_j \times d_j$ , and then

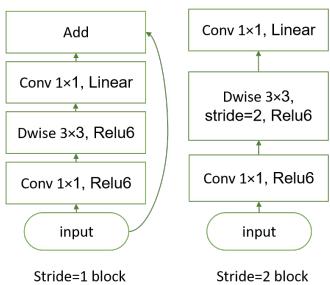


Figure 9. Two kinds of blocks in MobileNetV2 [17].

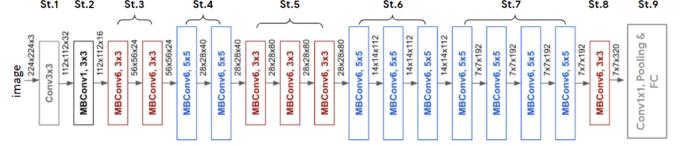


Figure 10. Architecture of EfficientNet-B0 [18].

the standard convolution's kernel size is  $K \in \mathbb{R}^{k \times k \times d_i \times d_j}$ , which require the computation cost of  $h_i \cdot w_i \cdot d_i \cdot d_j \cdot k \cdot k$ . Whereas, for depthwise separable convolutions, the computation cost is only  $h_i \cdot w_i \cdot d_i (k^2 + d_j)$ , which reduces computation compared to standard convolution by almost a factor of  $k^2$ . Generally, the kernel size is 3, then it means for the computation cost of depthwise separable convolution is 8 to 9 times smaller than that of standard convolutions [17].

The main innovation of MobileNetV2 is that it is an inversed residual network [17], which means the residual connection is used only when the stride is 1 and the number of channels remains the same. Two kinds of blocks in MobineNetV2 are shown in Fig. 9. Depth-wise convolution uses the kernel  $3 \times 3$ , and point-wise convolution uses 1 convolution. Between the residual connections, it contains the decompression phase, where the output channels are expanded among connected layers. In this study, we use the default settings for the expanding ratio, i.e., 6, as in the original code [17].

2) *Efficientnet for Holding Probability:* The holding probability is as part of inputs for the landing time prediction, which is captured by the Efficientnet [18] with the trajectory image as its input. The Efficientnet proposes a new compound scaling method, which uses a compound coefficient  $\phi$  to uniformly scales convolution network's width, depth, and resolution in the following principled way [18]:

$$\begin{aligned} \text{depth : } d &= \alpha^\phi \\ \text{width : } w &= \beta^\phi \\ \text{resolution : } r &= \gamma^\phi \\ \text{s.t., } \alpha \cdot \beta^2 \cdot \gamma^2 &\approx 2 \\ \alpha \geq 1, \beta \geq 1, \gamma \geq 1 & \end{aligned} \quad (2)$$

where  $\alpha, \beta, \gamma$  are constants that can be determined by a small grid search.  $\alpha \cdot \beta^2 \cdot \gamma^2 \approx 2$  is to make the FLOPS (floating point operations per second) approximately increase by  $2^\phi$ . In this study, we utilize the EfficientNet-B0, where the parameters are  $\phi = 1, \alpha = 1.2, \beta = 1.1, \gamma = 1.15$ , and the main building block is the mobile inverted bottleneck MBCConv. Its architecture is shown in Fig. 10.

### III. EXPERIMENTAL RESULTS AND ANALYSIS

#### A. Experimental Settings

We randomly split the total 8396 aircraft in the dataset  $\mathcal{D}$  into 70%, 15%, 15% for model training, validation, and testing, respectively, where features are normalized first. We set the loss function as the L1 loss, i.e., the Mean Average

TABLE III. PREDICTION ACCURACY EVALUATION OF EACH MODEL FOR ALL SCENARIOS.

Metrics	$\delta$ (in Mins)	MAPE		MAE		RMSE		BadRatio ( $\gamma = 0.3$ )	
		$\tau = 60$	$\tau = 90$	$\tau = 60$	$\tau = 90$	$\tau = 60$	$\tau = 90$	$\tau = 60$	$\tau = 90$
Baseline (No holding Module)	10	<b>0.0738</b>	0.0899	<b>82.23</b>	102.04	<b>115.91</b>	140.44	<b>0.0156</b>	0.0175
	15	0.0789	0.0829	86.10	91.38	116.17	124.81	0.0166	0.0191
	20	0.0808	0.0799	88.11	89.28	120.87	123.76	0.0216	0.0175
	25	0.0767	0.0786	86.81	87.05	123.28	119.23	0.0183	0.0158
	30	0.0788	0.0974	87.22	101.23	120.65	130.20	0.0166	0.0308
Proposed Method	10	<b>0.0390</b>	0.0759	<b>43.96</b>	86.76	<b>71.91</b>	123.95	<b>0.0025</b>	0.0208
	15	0.0762	0.0766	84.86	85.58	116.12	116.89	0.0066	0.0150
	20	0.0859	0.0740	94.36	84.89	126.61	121.13	0.0250	0.0125
	25	0.0763	0.0727	86.23	81.95	119.45	115.40	0.0108	0.0141
	30	0.0772	0.0786	86.05	88.64	118.97	123.98	0.0141	0.0175

Error (MAE), the optimizer as Adam, the batch size as 64, the epoch as 1000, and the learning rate as 0.001. The network has 6.4M parameters in total.

For the parameters of EfficientNet\_B0, all parameters are set as in [18]. For MobileNetV2, a pretrained mode is used, and all parameters are as in [17]. Since MobileNetV2 [17] is designed for classification problems, however, our problem is a regression problem, hence, we use these networks for feature engineering and the final classification layer is replaced by following sequential layers: Linear(1280, 64), BatchNorm1d(64) LeakyReLU(Linear(128)), Dropout(0.1). The MLPs in the proposed method as shown in Fig. 1 are structured as following sequential layers. (1) MLP\_N1: Linear(5, 16), LeakyReLU. (2) MLP\_N2: Linear(48, 32), BatchNorm1d(32), LeakyReLU, Linear(32, 8), Dropout(0.1), Sigmoid. (3) MLP\_N3: Linear(12, 16), BatchNorm1d(16), LeakyReLU. (4) MLP\_N4: Linear(80, 64), BatchNorm1d(64), LeakyReLU.

### B. Evaluation Metrics

To measure the accuracy of the ALT predictions, we adopted the commonly used metrics RMSE (Root Mean Square Error) (3), MAE (Mean Absolute Error) (4), MAPE (Mean Absolute Percentage Error) (5), as well as the defined  $BadRatio_\gamma$  (6), which are described below.

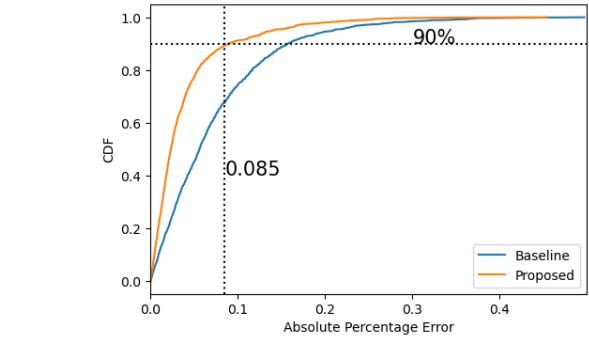
$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|^2} \quad (3)$$

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (4)$$

$$MAPE = \frac{1}{n} \sum_{i=1}^n \frac{|y_i - \hat{y}_i|}{y_i} \quad (5)$$

$$BadRatio_\gamma = \frac{\sum_{i=1}^n \sigma\left(\frac{|y_i - \hat{y}_i|}{y_i} > \gamma\right)}{n} \quad (6)$$

Here,  $n$  is the number of test samples,  $y_i$  is the ground truth label for the test sample  $i$ , and  $\hat{y}_i$  is the corresponding estimation. For all four metrics, lower values denote better estimations. RMSE is sensitive to the outliers and MAE is actually the L1 loss that directly represents the average estimation errors. MAPE (5) helps generally evaluate the estimations in terms of large ranges of ground truth values.  $BadRatio_\gamma$  is

Figure 11. CDF of Absolute Percentage Error.  $\tau = 60$ ,  $\delta = 10$ .

to evaluate the robustness of the model, where  $\sigma(x)$  is a sign indicator function, i.e.,  $\sigma(x) = 1$  if the expression  $x$  is true, otherwise 0.  $\gamma$  is a manually set threshold value, e.g.,  $\gamma = 0.3$  represents that the absolute percentage errors  $\frac{|y_i - \hat{y}_i|}{y_i}$  are higher than 30%, i.e., the relative accuracy would be lower than 70%. Hereafter, the metric  $BadRatio_\gamma$  denotes the ratio of bad predictions over the total number of test samples. Hence, a lower  $BadRatio_\gamma$  presents a better robustness of the model.

### C. Result Analysis

In the experiment study, we remove the "Holding Featurization" as a baseline to testify its effect on the ALT prediction. Additionally, the predictions are evaluated on different settings of the window size parameters  $\tau = 60, 90$  (in seconds) and  $\delta = 10, 15, 20, 25, 30$  (in minutes). Since the results of  $\tau = 30$  are quite worse, the corresponding results are not shown. Measured by the metrics of RMSE, MAE, MAPE and  $BadRatio_\gamma$ , the experimental results are shown in III, where  $\gamma = 0.3$ . The predictions measured by all metrics in III show that both the baseline and the proposed method performs better with  $\tau = 60$ . Meanwhile, the proposed method performs better than the baseline on all metrics ( $\tau = 60, \delta = 10$ ).

To further compare the performance of the baseline and the proposed method, we calculate the Absolute Percentage Error (APE) as in Eq. 7.

$$APE = \frac{|y_i - \hat{y}_i|}{y_i} \quad (7)$$

The cumulative distribution function (CDF) of the APE is shown in Fig. 11. We can find that the proposed ALT

TABLE IV. PERFORMANCE IMPROVEMENT

$\tau = 60$	ratio of $ err  < 60$ seconds	bad ratio $\gamma = 0.3$
$\delta = 10$		
Baseline	51.07%	1.56%
Proposed	79.40%	0.25%
Improvement	28.37%	1.31%

prediction method has quite lower APE values on all cumulative percentages, and it converges to 1 much faster than the baseline. For the proposed ALT prediction method, 90% predictions are of lower than 85% prediction errors.

In addition, as shown in Table. IV, the *BadRatio* of the proposed method is improved by 1.31% compared to the baseline with  $\gamma = 0.3$ . For the proposed method, 79.4% prediction errors are lower than 60 seconds, which has an 28.37% improvement compared to the baseline, where the prediction error is  $|err| = |y_i - \hat{y}_i|$ .

#### IV. CONCLUSIONS

In this paper, we propose a novel approach for predicting aircraft landing time (ALT) based on deep convolution neural networks applied to the generated trajectory images. The trajectory images capture various information about the aircraft, such as its spatial position, speed, heading, entry zone, relative distances to other aircraft, and real-time arrival traffic flows in the research airspace. These images can replace most of the complex feature engineering, and allow us to use state-of-the-art CNNs for the ALT prediction problem. Moreover, since the aircraft holding stage can significantly affect the ALT, our method use a module for the automatic holding-related feature engineering, which takes the generated images, as well as the leading aircraft holding status, the time and speed gap with the leading aircraft at TRC, and the speed variation at TRC as its input. Its output is further fed into the ALT prediction, and finally an end-to-end ALT prediction model is proposed. We conduct our study on the Changi Airport and use the related ADS-B dataset, as well as the METAR, flight plan, and online APD data in our experiment. Results show that our method can achieve an average accuracy of 96.1%, with 79.4% of the prediction errors being lower than 60 seconds. Experimental findings also verified that by integrating the holding featurization module with holding-related features, we can significantly improve the final ALT predictions. Compared to the baseline without the holding featurization, the proposed method reduces the mean absolute error (MAE) from 82.23 seconds to 43.96 seconds, and reduces the bad ratio of the predictions from 1.56% to 0.25%.

Predicting ALTs for inbound aircraft is crucial for sequencing them efficiently. In this paper, we only predict the ALT for each aircraft at the research airspace boundary (TRC), which is 50NM to Changi airport in our case study. However, predicting the ALT with longer distances (e.g., 100NM), choice of the Standard Arrival Route (STAR), and actions of the air traffic controllers would be further beneficial for managing arrival aircraft. Therefore, we plan to improve our proposed ALT prediction method by considering these factors in the future.

#### ACKNOWLEDGMENT

This work is supported by the National Research Foundation, Singapore, and the Civil Aviation Authority of Singapore (CAAS), under the Aviation Transformation Programme. Any opinions, findings and conclusions, or recommendations expressed in this material are those of the authors and do not reflect the views of the National Research Foundation, Singapore, and the Civil Aviation Authority of Singapore. The authors would like to thank all colleagues from CAAS for providing valuable comments and suggestions on this work.

#### REFERENCES

- [1] Y. Ma, W. Du, J. Chen, et al. A spatiotemporal neural network model for estimated-time-of-arrival prediction of flights in a terminal maneuvering area, IEEE Intelligent Transportation Systems Magazine, 2022, 15(1): 285-299.
- [2] D. Wesely, A. Churchill, J. Slough, et al. A Machine Learning Approach to Predict Aircraft Landing Times using Mediated Predictions from Existing Systems, AIAA AVIATION 2021 FORUM. Aug 2021, VIRTUAL EVENT.
- [3] H. Hardell, T. Polishchuk, L. Smetanová. Fine-grained evaluation of arrival operations, 10th SESAR Innovation Days (SIDs), Virtual, 2020.
- [4] K. Dönmez. Aircraft sequencing under the uncertainty of the runway occupancy times of arrivals during the backtrack procedure, The Aeronautical Journal, vol. 127, pp. 562-580, 2023.
- [5] O. Dhieb, Z. Lim, Sk. Goh, D. Pham, S. Alam, M. Schultz. Speed control strategies for e-aman using holding detection-delay prediction model, Proc. 10th Eurocontrol SESAR Innovation Days, pp. 1-10, 2020.
- [6] C. Deng, H. C. Choi, H. Park, et al. Trajectory pattern identification and classification for real-time air traffic applications in Area Navigation terminal airspace, Transportation Research Part C: Emerging Technologies, vol. 142, pp. 103765, 2022.
- [7] I. Dhieb, Z. Y. Wang, M. Liang, S. Alam, Michael Schultz, et al. Predicting Aircraft Landing Time in Extended-TMA Using Machine Learning Methods, ICRAT 2020, 9th International Conference for Research in Air Transportation, Sep 2020, Tampa, United States
- [8] J. Zhang, Z. Peng, C. Yang, et al. Data-driven flight time prediction for arrival aircraft within the terminal area, IET Intelligent Transport Systems, vol. 16, pp. 2663-275, 2022.
- [9] G. Chen, J. Rosenow, M. Schultz, et al. Using Open Source Data for Landing Time Prediction with Machine Learning Methods, Proceedings. MDPL, vol. 59, pp. 5, 2020.
- [10] A. Abecassis, D. Delahaye, M. Idan. A Machine Learning Framework to Predict General Aviation Traffic Counts A Case Study for Nice Cote D'Azur Terminal Control Center, SESAR Innovation Days. 2022.
- [11] O. Basturk, C. Cetek. Prediction of aircraft estimated time of arrival using machine learning methods, The Aeronautical Journal, vol. 125, pp. 1245-1259, 2021.
- [12] Z. Wang, M. Liang, D. Delahaye. Automated data-driven prediction on aircraft Estimated Time of Arrival, Journal of Air Transport Management, vol. 88, pp. 101840, 2020.
- [13] Y. Ma, W. Du, J. Chen, et al. A spatiotemporal neural network model for estimated-time-of-arrival prediction of flights in a terminal maneuvering area, IEEE Intelligent Transportation Systems Magazine, 2022, vol. 15, No. 1, pp. 285-299.
- [14] Iowa Environmental Mesonet, "Asos-awos-metar data download."
- [15] Aircraft Performance Database. <https://contentzone.eurocontrol.int/aircraftperformance/default.aspx?>
- [16] S. Zhang, Y. Zhang, T. Tay, et al. Learning-based Aircraft Trajectory Analysis Tool for Holding and Vectoring Identification with ADS-B Data, IEEE 25th International Conference on Intelligent Transportation Systems (ITSC), IEEE, pp. 1100-1105, 2022.
- [17] M. Sandler, A. Howard, M. Zhu, et al. Mobilenetv2: Inverted residuals and linear bottlenecks, CVPR 2018, Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4510-4520, 2018.
- [18] M. Tan, Q. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. International conference on machine learning. pp. 6105-6114, 2019.