

# A Spatial-Temporal Neural Network Model for Estimated Time of Arrival Prediction of Flights in Terminal Manoeuvring Area

Yan Ma, Wenbo Du, *Member, IEEE*, Jun Chen, Yu Zhang

Yisheng Lv, *Senior Member, IEEE*, Xianbin Cao, *Senior Member, IEEE*

**Abstract**—Affected by the nondeterministic nature of flight trajectories, external environment, and airport operational conditions, the prediction of the Estimated Time of Arrival (ETA) is one of the most challenging tasks for air traffic control in terminal manoeuvring area (TMA). Previous studies lack adequate utilization of the spatial and temporal behaviors embedded in the continuous trajectories. We propose a novel spatial-temporal neural network model for estimating time of arrival (STNN-ETA), which consists of three components: (1) trajectory pattern recognition which classifies historical trajectories into several patterns/clusters; (2) trajectory prediction that predicts a target flight's subsequent positions based on trajectory pattern matching; and (3) the arrival time prediction in which nonlinear function and recurrent units are adopted to capture the spatial-temporal features for the prediction purpose. In the proposed model, we also utilize a spatial attention mechanism and a temporal attention mechanism to focus on important features from radar echo maps and trajectory series respectively, and suppress unnecessary ones for ETA prediction. To validate the effectiveness of the proposed method, we apply it to predict the ETA of flights within the Beijing TMA. Extensive experiments show that STNN-ETA outperforms the state-of-the-art models in terms of the mean absolute error (MAE).

**Index Terms**—Air Traffic Control, Terminal Manoeuvring Area, Estimated Time of Arrival, Deep Learning

## I. INTRODUCTION

The Estimated Time of Arrival (ETA) is the time when the flight is expected to touch down on the runway, and is one of the key performance indicators defined by the International

Civil Aviation Organization (ICAO). Accurate prediction of ETA is of great importance to mitigate flight delays, providing aviation authorities with an index for flow management to minimize the deviation from the scheduled time of arrival. The economic losses caused by flight delays [1] also can be effectively reduced with an accurate prediction of ETA [2], [3]. Moreover, the accurate prediction of arrival time at the waypoint has a major impact on trajectory sequencing improvement and alleviation of traffic congestion [4], [5]. Additionally, the operational efficiency of airports and airlines can benefit from a more accurate ETA prediction [6]. A terminal manoeuvring area (TMA), which is also called terminal control area in the U.S. and Canada, is a designated area of controlled airspace surrounding one or more airports with high traffic volume [4]. The route structure in the TMA is complex with different entry points and flight procedures, traffic flows from different entry points converge and interact. In addition, the weather conditions in the TMA are variable and fickle [7]. Under convective conditions, the capacity and operation efficiency of the TMA decrease significantly [8]-[11], which prevents flights from arriving on time. Therefore, the prediction of the ETA for flights in TMA is a challenging problem, attracting many scientists from both academia and industry.

In recent decades, researchers worldwide have paid extensive attention to ETA prediction. The existing research methods can be broadly classified into model-driven and data-driven methods. The model-driven methods generally calculate the ETA of flights based on aircraft performance models and parametric or physics-based trajectory models. Wang et al. combined a mathematical model for miles-in-trail metering with discrete event simulation to examine the anticipated miles-in-trail delays for all the flights scheduled to arrive at any crossing point [12]. Roy et al. derived discrete-time hybrid models and applied the Interacting Multiple Model for target

This paper is supported by National Key R&D Program of China (Grant No.2019YFF0301400) and National Natural Science Foundation of China (Grant Nos.61961146005 and 62088101). (*Corresponding author: Wenbo Du.*)

Y. Ma, W. Du and X. Cao are with the School of Electronic and Information Engineering, Beihang University, Beijing 100191, China, with the Key Laboratory of Advanced Technology of Near Space Information System (Beihang University). (e-mail: myanm@buaa.edu.cn; wenbodu@buaa.edu.cn; xbcao@buaa.edu.cn).

J. Chen is with the School of Engineering and Materials Science, Queen Mary University of London, Mile End Road London E1 4NS, U.K. (e-mail: jun.chen@qmul.ac.uk).

Y. Zhang is with the Department of Civil and Environmental Engineering at the University of South Florida 4202 E. Fowler Ave. ENB118 Tampa, FL 33620, USA. (e-mail: yuzhang@usf.edu)

Y. Lv is with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 10090, China (e-mail: yisheng.lv@ia.ac.cn)

tracking [13]. Moreover, they incorporated the Interacting Multiple Model algorithm with autonomous transitions in the discrete model estimation to predict the ETA of flights. Wei et al. abstracted the descent stage of flights into discrete modes, derived the nonlinear dynamics for each discrete mode and applied state-dependent-transition hybrid estimation method for the ETA prediction [14]. Bai et al. investigated two methods to predict ETA at a downstream point to better conduct interval management [15]. However, most of these models make ideal assumptions, which rarely consider the actual constraints. Moreover, these kinds of predictive models are ineffective when facing massive real-time data [16].

Recently, the data-driven methods have been widely used in the traffic field [17]-[22]. The trend of ETA prediction has gradually shifted to data-driven methods, which are capable of learning from historical data with weak or even without assumptions. In particular, machine learning algorithms have been commonly used for ETA prediction. Glina et al. applied Quantile Regression Forests to produce conditional probability distributions for the ETA of flights [23]. Takacs extracted 56 features from the raw traffic dataset and proposed a 6-stage composed of successive ridge regressions and gradient boosting machines to predict the runway and gate arrival times of U.S. domestic flights [24]. Kern et al. employed random forests and took into account general information about the flight as well as weather and air traffic to enhance the accuracy of ETA prediction provided by the Federal Aviation Administration's Enhanced Traffic Management System, which is based on model-driven methods [25]. De Leege et al. used Generalized Linear Models to predict the time over the significant points along the standard arrival routes (STARs). In order to determine which regressors to include in the Generalized Linear Models, a stepwise regression approach is employed [26]. Zhu et al. carried out feature importance ranking, correlation analysis and applied two boosting tree algorithms, eXtreme Gradient Boosting (XGBoost) and Light Gradient Boosting Machine for route flight time prediction [27]. Ayhan et al. collected a richer set of features by pertinent 3D grid points. They also employed various regression models for ETA prediction of Commercial Flights and compared the prediction performance of these models. The results indicated that the boosting algorithms (Adaptive Boosting and Gradient Boosting) have better performance than the bagging algorithm (Random Forest Regressor) on all routes [28]. Wang et al. introduced Deep Neural Networks (DNNs) model with nested cross validation to predict the ETA of flights in Beijing TMA [16]. Their research indicates that the DNNs have better prediction performance than shallow neural networks.

Clustering algorithms have been extensively used in many fields [29]-[32]. In the TMA, the trajectories can be clustered into diverse patterns because the aircraft are suggested to follow the published STARs and the standard instrument departure (SID) routes [33]. Trivedi et al. found that combining clustering with some machine learning models is of great significance to improving the prediction accuracy of ETA [34]. Based on these studies, models that combine clustering with machine learning predictors have been applied to ETA prediction in recent years to further improve accuracy. Hong and Lee used dynamic time warping (DTW) to identify major trajectory patterns and applied multiple linear regression (MLR) for each pattern to

predict the ETA of flights from a specified entry point [35]. Wang et al. clustered the trajectories into several patterns by the density-based spatial clustering of applications with noise (DBSCAN) and then trained an individual neural network (NN)-based model for each cluster [33].

Note that existing data-driven methods generally map the features of a single trajectory point of each flight to the ETA, without considering the spatial and temporal features embedded in the continuous trajectory. Spatial features are diverse, even complex. Continuous trajectory points describe the spatial location information and different flying situations, such as heading change, descending, speed reduction. Moreover, these spatial features are time-varying, in other words, temporal features are embedded in the continuous trajectory points. For example, in non-rush hours, flights are frequently instructed to the optimal path to shorten the distance and time. During rush hours, due to the traffic congestion in the TMA, flights may line up or detour, which leads to arrival time changes. However, it is time-consuming and even infeasible to extract these features explicitly, due to the non-deterministic of external operational circumstances and traffic. We need to consider them implicitly in the model.

To apply the spatial and temporal features to ETA prediction, we need to predict the future trajectory of the target flight from the current position to the touch down point on the runway. Although there are STARs in the TMA, deviations exist between the actual trajectories and STARs due to the influence of convective weather, the high density of aircrafts, the constrained airspace structure, etc. Therefore, the future trajectory of the target flight cannot be predicted by STARs directly. In light of the above, we propose a novel model, namely STNN-ETA, for ETA prediction in the TMA which consists of three main components: trajectory pattern recognition, trajectory prediction and arrival time prediction.

The main contributions of this paper are summarized as follows:

- 1) We propose a novel model for ETA prediction, in which trajectories are regarded as time series. Comparing previous models, we first design a layer including Bi-LSTM to effectively extract spatial and temporal features from trajectories.
- 2) We develop a multi-layer feature fusion component in which a spatial attention mechanism is designed to focus on important convective weather features and an additional factor layer is introduced to incorporate other features. The temporal attention mechanism is presented to integrate features.
- 3) We evaluate our model based on Beijing TMA. Extensive experiments show our models outperformers all baselines in terms of the mean absolute error (MAE).

The rest of this paper is as follows: Section II is the research problem and motivation, Section III introduces the model we propose, Section IV is the experimental description, and we conclude this paper in Section V.

## II. RESEARCH PROBLEM AND MOTIVATION

### A. Problem Statement

In the TMA, a flight moves from an entry point to the runway. The ETA prediction problem is defined as follows: for the target flight entering the TMA, we aim to design a frame to predict the estimated travel time between the current position to the touchdown point on the runway based on history trajectory series  $\mathcal{H}$ , environmental information  $\mathcal{E}$  and airport operational conditions  $\mathcal{C}$ . As the flight moves in the TMA, the predicted ETA should be updated,

$$\hat{y} = f(\mathcal{H}, \mathcal{E}, \mathcal{C}), \quad (1)$$

where  $f$  is the prediction function, which can be a deep neural network model.  $\hat{y}$  is the predicted value for ETA.

### B. Research Motivation

To address the shortcoming of the above approach, we propose a spatial-temporal neural network model based on the continuous trajectory series for ETA prediction. Given the target flight in the TMA, the future trajectory of the target flight from the current position to the touch down point on the runway needs to be predicted. Subject to the traffic situation in the TMA, flights often deviate from STARS. Therefore, it is not good to predict the future trajectory of the target flight based on STARS. In this work, we apply a clustering algorithm to extract trajectory patterns from the historical trajectory set. Then, based on the patterns, the trajectory of the target flight from the current position to the touchdown point on the runway can be predicted. At last, the arrival time prediction component makes full use of trajectory series, environmental information, and airport operational conditions to predict the ETA of flights.

We take the Beijing TMA as an example. Fig. 1 shows the range of the Beijing TMA according to the China Electronic Aeronautical Information Publication [36]. Beijing TMA contains six entry points, namely KM, JB, BOBAK, VYK, DOGAR, and GITUM [33]. The eight STARS and lateral trajectories of arrival between the entry points and 3 parallel runways (36L, 36R, 01) are shown in Fig. 1, which provide a good illustration of the air traffic complexity in the TMA.

It is straightforward to do the ETA prediction by following the STARS and taking into account the aircraft performance models. However, such kind of idea is ineffective as we can clearly see from Fig. 1 that most flights do not follow the STARS. Although the trajectories shown in Fig. 1 look haphazard, they are resulted from a couple of factors which are often non-trivial to be characterized by mathematical models. As compared to model-driven methods for ETA prediction, a machine learning model can learn the underlying patterns from the massive historical data to do the ETA prediction in a purely data-driven manner. While some machine learning methods for ETA prediction are available in the literature, they failed to make full use of massive air traffic data. Motivated by all these reasons, in this study we propose a new method that not only makes full use of historical flight trajectory information but also considers the weather information as well as the TMA congestion degree to improve the prediction accuracy of ETA.

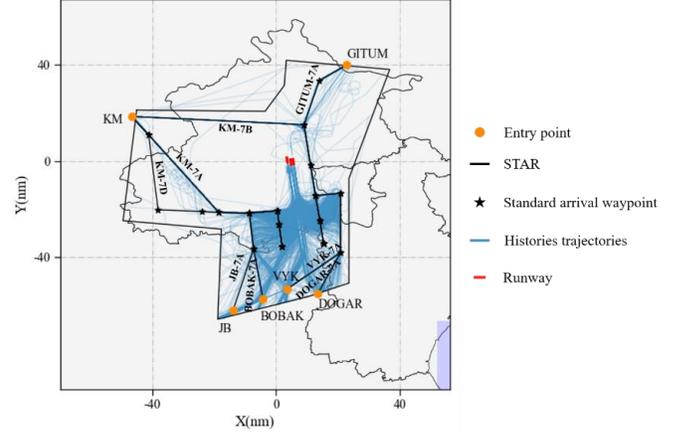


Fig. 1. Structure of Beijing TMA. nm represents nautical mile.

## III. METHODOLOGY

### A. Model Architecture

As shown in Fig. 2, the proposed spatial-temporal neural network model for estimating the time of arrival (STNN-ETA) is comprised of three components. The trajectory pattern recognition component efficiently categorizes the historical trajectories into different clusters and noise using DBSCAN. The trajectories of each cluster share a similar pattern. The abnormal trajectories in special cases are regarded as noise. The trajectory prediction component is designed to predict the trajectory from the current position to the touchdown point on the runway of the targeted flight based on the patterns recognized and the trajectory already accomplished in the TMA. The arrival time prediction component aims to predict the estimated time of arrival of the target flight, which contains four layers: weather layer (WL), additional factor layer (AFL), spatial-temporal layer (STL), and prediction layer (PL). The weather layer is designed to capture convective weather features based on the spatial attention mechanism. The additional factor layer is used to integrate the additional factors to enhance the performance. The spatial-temporal layer aims to learn the spatial and temporal features from trajectories based on Bi-LSTM. The prediction layer applies the temporal attention mechanism and full connection method to predict the ETA of the flight.

In order to illustrate STNN-ETA, we further define some notations used in this paper.

**Definition 1 (Trajectory Set)** Trajectories of landing aircraft, only portions between the TMA entry points and runways are considered. We resample each trajectory with equal time intervals, as the flight speed is relatively slow in the TMA, it is sufficient to set 30s as the time intervals. After being resampled, each trajectory consists of a sequence of points. The  $l^{th}$  trajectory is expressed as  $Tr_l = \{P_1, P_2, \dots, P_{n_l}\}$ . Each point  $P_i$  contains the longitude, latitude, altitude, horizontal ground velocity and timestamp; therefore, the point can be formulated as  $P_i = \{lon_i, lat_i, alt_i, v_i, t_i\}$ .

**Definition 2 (Trajectory of the target flight)** We resample

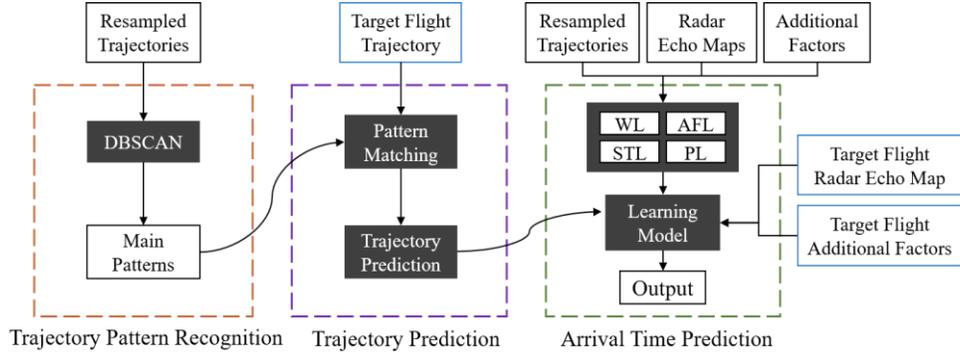


Fig. 2. The framework of the proposed STNN-ETA

the already accomplished trajectory of the target flight in the TMA with equal time intervals of 30s and obtain  $Tr_a = \{P_1, P_2, \dots, P_{n_a}\}$ .

**Definition 3 (Radar echo map)** A radar echo map is generated based on the reflectivity of the target radar wave. Radar echo maps describe the whole environmental area with a grid map, and at spatial scale the area is divided into different levels, which are represented by different tones. Different tones of the radar echo map indicate different radar echo intensities. In addition, the radar echo map is updated at a certain frequency in time. The controller can modify the flight path or adjust the interval between aircrafts according to the tones of the radar echo map. The original horizontal resolution of the radar echo map is 4000 by 7000. The longitude ranges from 70E to 140E, and the latitude ranges from 15N to 55N. We crop the original images and only keep space in the studied region.

### B. Trajectory Pattern Recognition

The clustering algorithm is classified as unsupervised learning, which is designed to identify groups of similar observations in a dataset without prior knowledge [37]. The trajectory pattern recognition problem aims to find clusters of similar trajectories in the spatial dimension [37]. The trajectories in each cluster have a similar pattern.

Density-based spatial clustering of applications with noise (DBSCAN), a commonly used density-based clustering algorithm [38], has some advantages that make it suitable for trajectory pattern recognition. First, due to the complexity of air traffic, patterns of trajectories are diverse and have various shapes in spatial scale, DBSCAN can find arbitrarily shaped patterns. Second, abnormal trajectories can occur in some special cases and can be considered noise [33]. DBSCAN is robust to the quality of datasets and can divide historical trajectories into several main patterns and noise.

DBSCAN clusters the trajectories based on two input parameters: the distance threshold  $\epsilon$  and the minimum number of points MinPts [37]. The core concept of DBSCAN is that the number of points in the  $\epsilon$ -neighborhood is not less than MinPts. DBSCAN does not set the number of clusters in advance but starts with a randomly selected core point of the dataset. It finds all the density-reachable points in the  $\epsilon$ -neighborhood; these density-reachable points and the core point are defined as a

cluster. After that, the algorithm selects another core point and applies the same procedure until all the core points are processed.

Since the primary purpose of trajectory clustering is to recognize the spatial pattern, the spatial shape of the trajectory is very important. Euclidean distance is inappropriate because a small misalignment in time would result in a large distance between the trajectories. To address this problem, DTW is applied to calculate the distance used in DBSCAN. The DTW algorithm finds the optimal alignment of two trajectories which minimizes the effects of shifting and distortion in time [39][40]. Then the distance can be calculated based on the alignment. The formulation is,

$$DTW(i, j) = dist(p_i, q_j) + \min \begin{cases} DTW(i, j-1) \\ DTW(i-1, j) \\ DTW(i-1, j-1) \end{cases} \quad (2),$$

where  $p_i$  and  $q_j$  are the  $i^{th}$  and  $j^{th}$  points of trajectory  $Tr_p$  and  $Tr_q$  and  $dist(p_i, q_j)$  represents the Euclidean distance.

After clustering, we calculate the average distance of each trajectory pair in the same pattern and select the one with the smallest average distance as the representative trajectory.

### C. Trajectory Prediction

In practice, flights enter the TMA individually through different entry points and fly toward the runway by following the STAR with a possible deviation resulted from controllers' recommendations. Given the trajectory that the target flight has passed through the TMA, we need to predict the trajectory from the current position to the runway. To achieve this goal, trajectory pattern matching and trajectory prediction are utilized.

Trajectory pattern matching aims to match the trajectory of the target flight with the identified trajectory pattern based on the representative trajectories of each pattern. We calculate the average distance between the given trajectory  $Tr_a$  and the representative trajectory of each pattern.  $Tr_a$  and the representative trajectory with the smallest average distance from  $Tr_a$  belong to the same pattern.

The purpose of trajectory prediction is to predict the future trajectory of the targeted flight from the current position to landing. We assess the similarity between  $Tr_a$  and the historical trajectories that are categorized as the same pattern, the

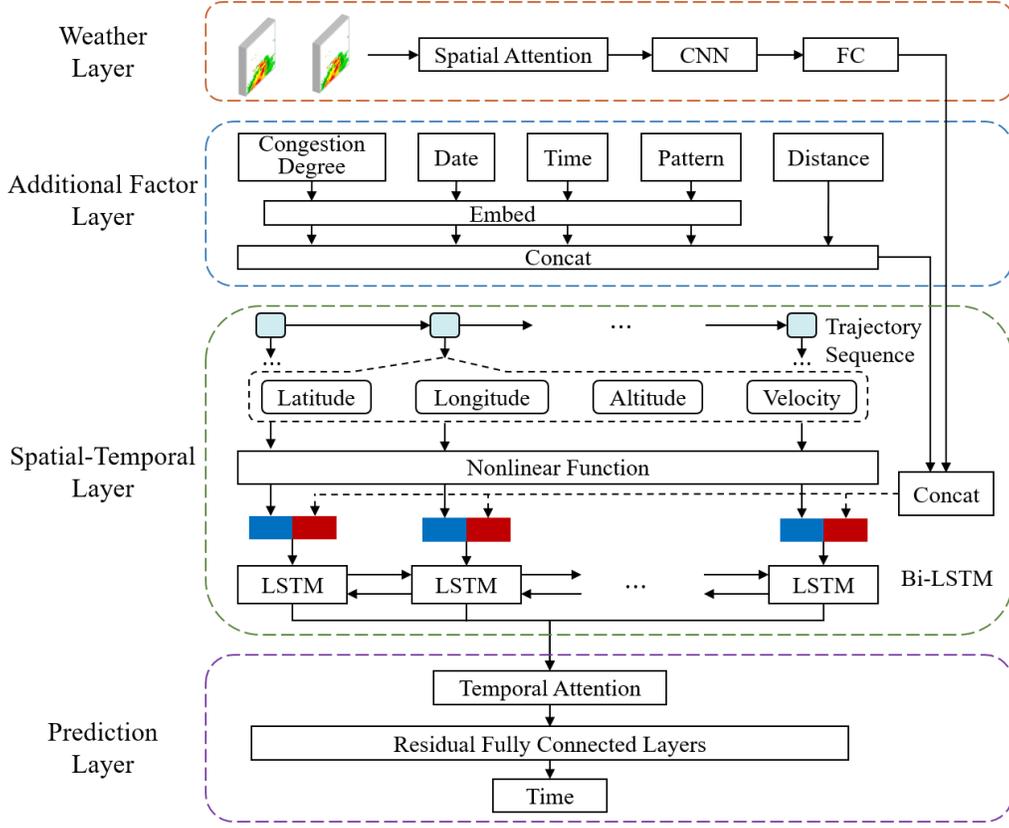


Fig. 3. The architecture of arrival time prediction component

formulation is,

$$sim = \sum_{i=1}^{n_a} ((lon_i^a - lon_i^m)^2 + (lat_i^a - lat_i^m)^2 + (alt_i^a - alt_i^m)^2 + (v_i^a - v_i^m)^2)^{1/2} \quad (3),$$

where  $m$  is the  $m^{th}$  historical trajectory belonging to the same pattern as  $Tr_a$ .  $lon$ ,  $lat$ ,  $alt$  and  $v$  have been normalized. And the detailed process of normalization is,

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}}, \quad (4),$$

where longitude, latitude, altitude and velocity are substituted into  $x$ , and  $x_{max}$ ,  $x_{min}$  represent the maximum and minimum values of the variable  $x$ , respectively.

Resampling is then performed to transform each historical trajectory sequence after point  $P_{n_a}$  into a sequence with length  $L$ . Then, we calculate the mean of the attributes of the five trajectories at each time point, including the longitude  $lon_i$ , latitude  $lat_i$ , altitude  $alt_i$ , and horizontal ground velocity  $v_i$ . Finally, we obtain the predicted trajectory of the targeted flight from the current position to landing,  $Tr_a^p = \{P_1, P_2, \dots, P_L\}$ .

#### D. Arrival time prediction

The arrival time prediction component contains four layers: the spatial-temporal layer, weather layer, additional factor layer, and prediction layer. The structure is shown in Fig. 3. In the training phase, the arrival time prediction component learns from historical data. In the test phase, the predicted trajectory of the target flight, radar echo map, TMA congestion degree, time of day, day of the month, and the trajectory pattern are input to the trained model to generate the predicted ETA.

##### 1) Weather Layer

The weather layer is designed to extract convective weather features from radar echo maps, which is shown in Fig. 4.

Convolutional neural network (CNN) is a neural network that is used to extract image features in many fields. As the convective weather in different regions has different effects on the flight [41], we introduce a spatial attention mechanism to focus on important features and suppress unnecessary ones.

The spatial attention mechanism mainly consists of channel attention and spatial attention modules. The essence of the channel attention module is to assign a weight to each channel and the weight represents the relevance of the features of each channel to the arrival time of flights. Given a radar echo map as input  $E$ , the formulation of the channel attention module is,

$$M_c(E) = \sigma \left( MLP(avgpool(E)) + MLP(maxpool(E)) \right) \quad (5),$$

$$E_c = M_c(E) \otimes E \quad (6),$$

where  $\sigma$  is the sigmoid function. For generating the channel attention of the radar echo map  $E$ , the spatial dimension needs to be squeezed.  $avgpool$  and  $maxpool$  are applied to aggregate spatial information by selecting the average and the maximum value of the feature map of each channel as the representative of the channel respectively and generate two different feature vectors. Then, two feature vectors are forwarded to multi-layer perceptron (MLP) with a hidden layer to produce the channel map  $M_c$ .  $\otimes$  is the Hadamard product, used for broadcasting channel attention along the spatial dimension.

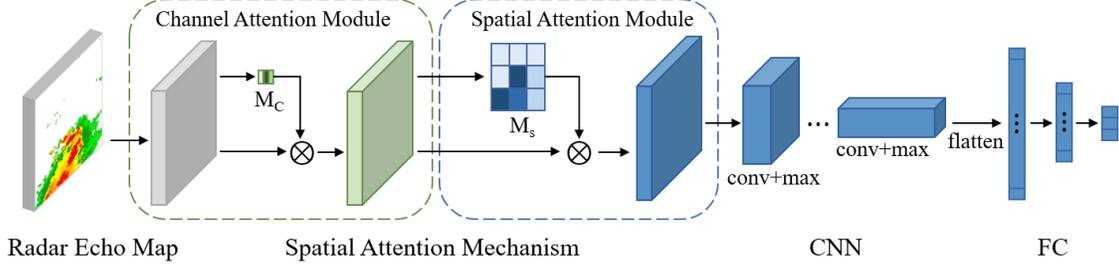


Fig. 4. The architecture of the weather layer

Spatial attention module emphasizes spatial features by assigning weights to different positions of each channel. The output of the channel spatial attention module  $E_c$  is taken as the input of spatial attention module and the formulation is,

$$M_s(E_c) = \sigma([\text{avgpool}(E_c); \text{maxpool}(E_c)] * W^M) \quad (7),$$

$$E_s = M_s(E_c) \otimes E_c \quad (8),$$

where  $[\cdot; \cdot]$  denotes the concentration operation that combines different features,  $*$  represents the convolutional operation,  $\sigma$  is the sigmoid function.  $\text{avgpool}$  and  $\text{maxpool}$  are used to aggregate channel features by selecting the average and the maximum values along the channel axis, and obtain two different feature vectors. After concatenating these two feature vectors, a convolution operation is utilized to generate a spatial attention map  $M_s$ . Then, the Hadamard product is adopted to broadcast spatial attention along the channel dimension.  $E_s$  is the final output of the spatial attention mechanism.

Then CNN is used to extract the convective weather features, which takes  $E_s$  as input  $E^0$  and feeds it into  $K$  conv + max layers. In each conv + max layer, there is a convolutional layer followed by a max-pooling layer. The formulation at each layer  $k$  is,

$$E^k = \text{maxpool}(f(E^{k-1} * W^k + b^k)) \quad (9),$$

where  $*$  is a convolutional operation.  $W^k$  and  $b^k$  are learnable parameters at the  $k^{\text{th}}$  layer.  $f(\cdot)$  is a ReLU activation function and  $f(x) = \max(0, x)$ .  $\text{maxpool}$  is used to reduce the dimension of the representations after applying convolutions. After  $K$  layers, we obtain the output  $E^K$ .

Then, a flatten layer is applied to transfer the output  $E^K$  to a vector  $\hat{E}$ , which is taken as the input of the following three fully connected layers (FC) to map the vector  $\hat{E}$  to a low dimension. Finally, we obtain the output of the weather layer  $V^w$ .

### 2) Additional Factor Layer

As we mentioned, the arrival time of flights is affected by many complex factors, such as the TMA congestion degree, time of day, day of the month, and trajectory patterns. An additional factor layer is designed to incorporate these factors into our model to improve the accuracy of prediction.

We first incorporate the factors that include the TMA congestion degree, time of day, day of the month, and trajectory patterns. The embedding method [42] can reduce the dimensions of vectors and improve the training speed. The embedding method reduces the dimension  $D_1$  of each discrete variable and obtains a transformed space  $R^{D_2 \times 1}$  by multiplying

a parameter matrix  $W \in R^{D_1 \times D_2}$ .  $D_2$  is the dimension of the transformed space.

In addition to the embedded vectors, we incorporate the distance from the reference point. The center of the runway is used as the reference point, and the distance indicates flights toward the center of the TMA. Finally, the embedded vectors are concatenated with the distance as the output  $V^{\text{ext}}$  of the additional factor layer. We concatenate the weather features vector  $V^w$  with the additional factor features vector  $V^{\text{ext}}$  to obtain a time-invariant feature vector  $V^f$ .

The arrival time of flights is closely related to the TMA congestion degree [43], which can be measured by the number of flights that enter the TMA [44]. The increase in the number of flights that enter the TMA will lead to restrictions on flight entry procedures and fierce competition in airspace environment and runway occupation.

### 3) Spatial-Temporal Layer

We first capture the spatial features by a nonlinear function. The formulation is,

$$V_i^s = \tanh(W^s \cdot [\text{lat}_i; \text{lon}_i; \text{alt}_i; v_i]) \quad (10),$$

where  $[\cdot; \cdot]$  is a concentration operation that connects different features,  $W^s$  is a learnable parameter,  $\tanh$  is an activation function, the range of the output of the  $\tanh$  function is  $[-1, 1]$ , and the mean is zero. The attributes of the  $i^{\text{th}}$  track point are mapped into a vector  $V_i^s \in R^{16}$ . Finally, we obtain a sequence  $V^s \in R^{16 \times L}$ , which represents the spatial features of the trajectories.

Recurrent neural network (RNN) is a neural network that mainly processes time series. However, RNN has vanishing and exploding gradient problems due to back propagation. LSTM is a variant of RNN, and the structural design of LSTM can solve the vanishing gradient problem of RNN [45]. Compared with LSTM, Bi-LSTM utilizes an additional backward layer and thus enhances the memory capability [46]. We use Bi-LSTM to capture the temporal features from the historical trajectories in our model.

We concatenate the spatial features vector with the time-invariant features vector  $V^f$  to get the vector  $X_i = [V_i^s; V^f]$ ,  $[\cdot; \cdot]$  concatenates different features in one dimension. Then, we feed  $X_i$  to Bi-LSTM at each time step. At each time step, the LSTM contains a cell  $C_i$ , an input gate  $I_i$ , a forget gate  $F_i$  and an output gate  $O_i$ . The forget gate  $F_i$  takes  $[h_{i-1}; X_i]$  as input and controls the extent to which the information of the previous cell state  $C_{i-1}$  will be kept to cell state  $C_i$ , and the formulation is,

$$F_i = \sigma(W^F \cdot [h_{i-1}; X_i] + b^F) \quad (11),$$

where  $W^F$ ,  $b^F$  are learnable parameters,  $\sigma$  is an activation function.

The input gate  $I_i$  determines the extent to which the input of the current networks  $X_i$  flows into cell state  $C_i$ , and the formulation is,

$$I_i = \sigma(W^I \cdot [h_{i-1}; X_i] + b^I) \quad (12),$$

$$\tilde{C}_i = \tanh(W^C \cdot [h_{i-1}; X_i] + b^C) \quad (13),$$

where  $W^I$ ,  $b^I$ ,  $W^C$ ,  $b^C$  are learnable parameters,  $\tanh$  is an activation function.

After the two steps, the cell state can be updated, and the process can be expressed as follow,

$$C_i = F_i \otimes C_{i-1} + I_i \otimes \tilde{C}_i \quad (14),$$

where  $\otimes$  is the Hadamard product.

The output gate  $O_i$  controls the extent to which the information of the cell state  $C_i$  is used to compute the output  $h_i$ , and the formulation is,

$$O_i = \sigma(W^o \cdot [h_{i-1}; X_i] + b^o) \quad (15),$$

$$h_i = O_i \otimes \tanh(C_i) \quad (16),$$

where  $W^o$  and  $b^o$  are learnable parameters,  $h_i$  is the hidden state, which is the output of a cell unit of the LSTM.

We get the  $i^{th}$  hidden states  $h_i^f$  and  $h_i^b$  of the forward and backward layer respectively. Two hidden states are concatenated to get the  $i^{th}$  hidden state  $h_i = [h_i^f; h_i^b]$ . Finally, we obtain a hidden state sequence  $\{h_i\}$ .

#### 4) Prediction Layer

Flights may turn or accelerate in some areas. Paying more attention to these parts can improve the accuracy of the prediction. Therefore, we apply the temporal attention mechanism to learn the importance of trajectory features at different time steps in an adaptive way to predict ETA accurately. The essence of the temporal attention mechanism is to assign a weight to each input passed to it and then output the weighted sum of all inputs. In the prediction layer, the hidden state sequence  $\{h_i\}$  from recurrent units is taken into as the input. Then, the attention weights of hidden states are calculated as

$$u_i = Q \cdot \tanh(W^u \cdot h_i + b^u) \quad (17),$$

where  $Q$ ,  $W^u$  and  $b^u$  are learnable parameters.  $u_i$  is the attention weight, which represents the correlation between the  $i^{th}$  hidden state with the arrival time of the flight. Then, the softmax function is utilized to normalize all the  $u_i$ , the formulation is,

$$\beta_i = \frac{\exp(u_i)}{\sum_{i=1}^L \exp(u_i)} \quad (18),$$

where  $0 \leq \beta_i \leq 1$  and  $\sum_{i=1}^L \beta_i = 1$ . Once we obtain the attention weights, the output vector  $H$  of the temporal attention mechanism is computed with

$$H = \sum_{i=1}^L \beta_i h_i \quad (19).$$

Finally, the vector  $H$  is passed to several residual fully connected layers, which consist of fully connected layers and residual connections [47]. The residual connection adds the residual term to the fully connected layer. It has been shown that the introduction of the residual term can prevent network degradation. Finally, we obtain the estimated time of arrival.

During the training phase, the mean absolute percentage error (MAPE) is applied as our loss function.

## IV. RESULTS AND DISCUSSION

### A. Data Preparation

To study this problem, the Beijing TMA is selected as the study case, which is one of the busiest TMAs in China. Beijing Capital International Airport (BCIA) is one of the busiest airports in the world and contains three parallel runways: 18R/36L, 18L/36R, and 19/01 (Wang et al., 2018). We only take the arrival flights landing on runway 36R/36L/01 into account. The datasets in this study include Automatic Dependent Surveillance-Broadcast (ADS-B) data and radar echo maps from August 1st, 2019 to October 9th, 2019 over the TMA of Beijing Capital International Airport.

Each record of the ADS-B data contains the aircraft ID, type of operation (departure/arrival), longitude, latitude, altitude, horizontal ground velocity, coordinated universal time (UTC) timestamp, heading, etc. After data filtering and cleaning, we extract 17,596 trajectories from the dataset. A total of 12,348 trajectories are used for training, and 5,248 trajectories are used for testing. As only the flights in the Beijing TMA are considered, and we crop the original images and only keep the spatially nearby region part. The longitude and latitude of the four corners are (115.32E, 40.47N), (117.17E, 40.47N), (115.31E, 38.58N), and (117.17E, 38.58N), respectively. The horizontal resolution of images after cropping is 189 by 186. According to the experiment, the number of all landing flights within 2100s after the target flight enters the TMA is most related to the ETA, so we select this number as the TMA congestion.

The distribution of the transit time from flights entering the Beijing TMA to landing is shown in Fig. 5. The time that most flights spend in the TMA ranges from 900 to 1500 seconds, with an average of 1229.4 seconds.

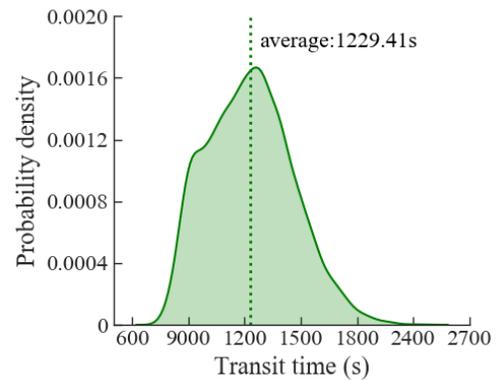


Fig. 5. Transit time probability distribution of flights in Beijing TMA

### B. Clustering Performance and Trajectory Prediction Performance

The parameters  $\epsilon$  and MinPts have a great influence on the clustering performance of DBSCAN. The parameter grids used

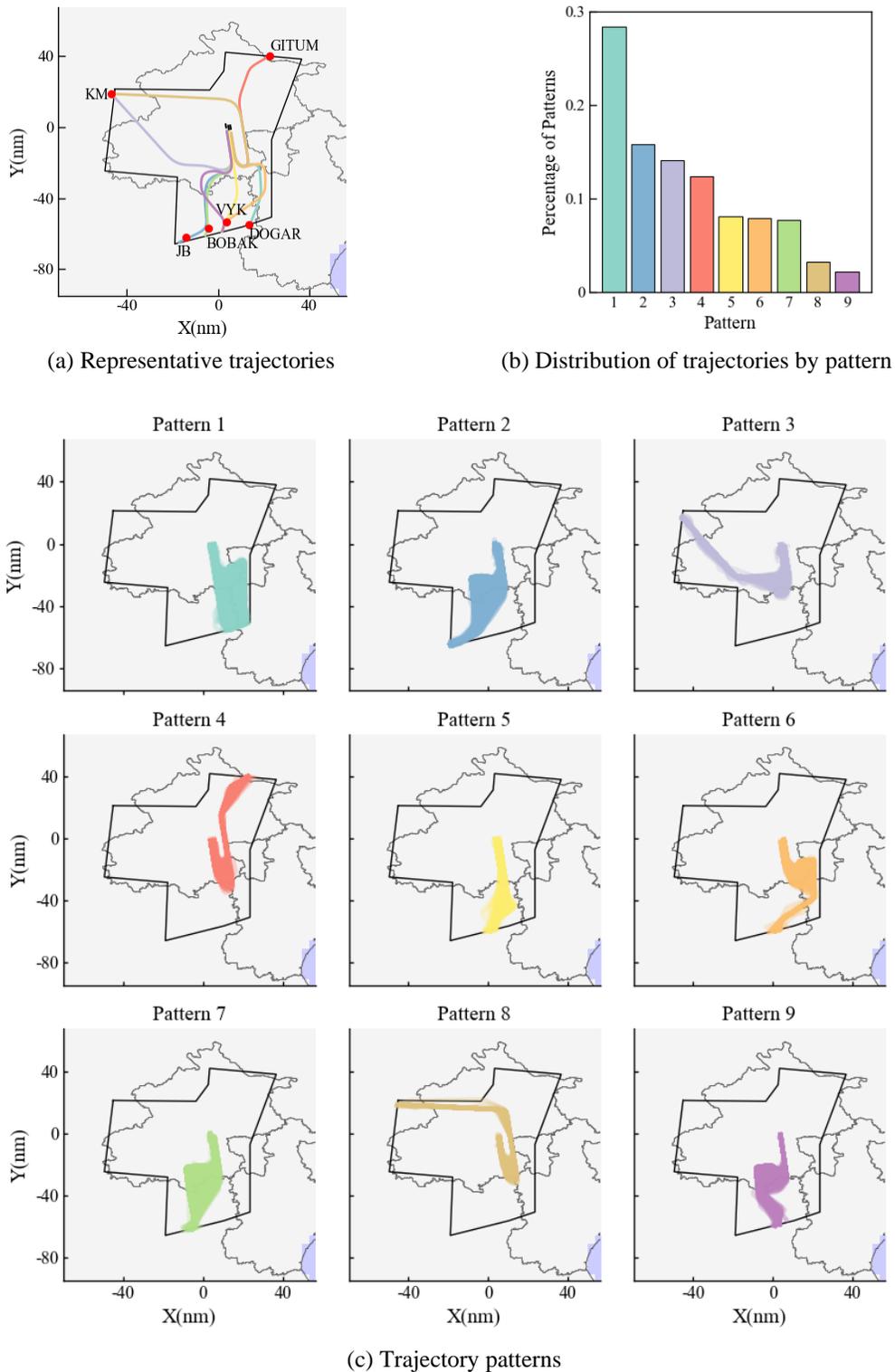


Fig. 6. Results of trajectory clustering. nm represents nautical mile.

in DBSCAN are shown in Table I. In the clustering process, the distance threshold  $\epsilon$  is set to 0.18, and the minimum number of points MinPts is set to 10, the noise accounts for a small proportion of datasets. DBSCAN clusters the historical

trajectories into nine main patterns and noise. The noise is mainly composed of holding patterns and trajectories with large vectoring, which will have an interference for the ETA prediction training stage. Therefore, we delete the noise from

the dataset. Fig. 6c is the 2D plot of the trajectories in the Beijing TMA, where different colors denote trajectories in different patterns. We find that the patterns of the trajectories are related to the entry points and runways. The representative trajectories of the patterns in Fig. 6a show that more than one pattern exists for some entry points. As an example, the flights categorized as pattern 5, pattern 6, and pattern 9 enter the Beijing TMA at entry point VYK. Fig. 6b shows the distribution of the patterns. Pattern 1 is the dominant pattern and concentrates more than 25% of the trajectories.

TABLE I  
THE GRIDS OF HYPERPARAMETERS

Parameters	Parameter grid
$\varepsilon$	{0.06, 0.12, 0.18, 0.24, 0.30, 0.36, 0.42, 0.48}
MinPts	{5, 10, 20, 30, 40, 50, 60}

To evaluate the performance of trajectory prediction, we analyze the distribution of the prediction errors. Horizontal error represents the distance between the predicted 2D coordinate (latitude and longitude) and the actual one. Vertical error is the distance between the predicted altitude and the actual altitude. The horizontal errors of points are mostly in the range of 0 nm to 9 nm, with an average of 2 nm. The vertical errors of points are mostly in the range of 0 nm to 0.6nm. Moreover, the horizontal and vertical errors of trajectories have similar distribution with that of points.

### C. ETA Prediction Performance

#### 1) Methods for Comparison

To evaluate the performance of our model, we compare our model with the following methods.

Random Forest Regressor (RF) [25]: a tree-based algorithm. The ETA is formulated as a regression problem, and the input includes the features of a track point, the features extracted from the radar echo maps and additional factors.

Extreme Gradient Boosting (XGBoost) [48]: one of the boosting algorithms. Regularization is introduced in the loss function, which can reduce overfitting during the training phase. XGBoost learns column subsampling from RF. The input of XGBoost is the same as that of RF.

Adaptive Boosting (AdaBoost) [49]: an iterative algorithm. The core of AdaBoost is to find the weight of the present classifier and update the error of the samples. It is adaptive and very sensitive to noise and abnormal data.

DBSCAN+MCNN [33]: a model that combines clustering with a machine learning predictor. The historical trajectories are identified as different clusters by DBSCAN, and each cluster has an individual NN-based predictor.

RNN-ETA: a simplified model of STNN-ETA. We use RNN to capture temporal features instead of Bi-LSTM.

LSTM-ETA: a simplified model of STNN-ETA. We use an LSTM to capture temporal features instead of Bi-LSTM.

GRU-ETA: a simplified model of STNN-ETA. We use a GRU to capture temporal features instead of Bi-LSTM.

#### 2) Performance Evaluation

We use the mean absolute percentage error (MAPE) and the mean absolute error (MAE) to evaluate the proposed methods,

$$MAPE = \frac{1}{n} \sum_{r=1}^n \left| \frac{\hat{y}_r - y_r}{y_r} \right| \times 100\% \quad (20),$$

$$MAE = \frac{1}{n} \sum_{r=1}^n |\hat{y}_r - y_r| \quad (21),$$

where  $\hat{y}_r$  is the  $r^{th}$  predicted value of the ETA of the target flights and  $y_r$  is the  $r^{th}$  observed value of ETA.

For hyperparameters in the arrival time prediction component, the batch size of each epoch is 64 and the learning rate is 0.5. The hidden units in the Bi-LSTM, RNN, LSTM, GRU are set to 128. The model is trained by Adam.

Considering that different time horizons of prediction may have different prediction errors, we compare the performance of these methods at different time horizons. The results are shown in Table II.  $t(s)=0$  indicates that the flight is at the entry point of the TMA;  $t(s)=240$  indicates that the flight has been flying in TMA for 240 seconds;  $t(s)=480$  indicates that the flight has been flying in TMA for 480 seconds. As shown, STNN-ETA outperforms all the compared methods in each group. At the entry points,  $t(s)=0$ , STNN-ETA improves the MAPE by 3.45% and the MAE by 37.94 seconds in the dataset compared with those of the best performance of the existing methods. We also find that the MAE tends to be smaller when the flight is close to its destination. Another observation is that MAPE shows similar trends with MAE, and with less time to

TABLE II  
PERFORMANCE COMPARISON

Model	$t(s) = 0$		$t(s) = 240$		$t(s) = 480$	
	MAE(s)	MAPE (%)	MAE (s)	MAPE (%)	MAE (s)	MAPE (%)
DBSCAN+MCNN	127.33	10.51	98.24	9.98	70.12	9.06
RF	134.96	11.37	97.59	9.79	69.87	8.98
XGBoost	130.51	10.98	94.81	9.58	70.44	9.15
AdaBoost	131.15	11.01	94.19	9.42	69.57	8.93
RNN-ETA	99.94	7.90	87.39	8.93	61.10	8.04
LSTM-ETA	91.53	7.29	73.29	7.52	54.58	7.26
GRU-ETA	93.73	7.56	72.14	7.33	56.21	7.42
STNN-ETA	<b>89.39</b>	<b>7.06</b>	<b>70.06</b>	<b>6.91</b>	<b>51.42</b>	<b>6.87</b>

the destination, the prediction performance of all methods continues to improve. When  $t(s)=240$  and  $t(s)=480$ , the MAPE of STNN-ETA is 6.91%, 6.87% respectively. RNN-ETA, LSTM-ETA, GRU-ETA use RNN, LSTM, GRU respectively to capture the temporal features in the spatial-temporal layer instead of Bi-LSTM, which are much better than all existing methods.

To analyze the effectiveness of the temporal attention mechanism, we replace the temporal attention mechanism with mean pooling, which equally treats the hidden state sequence. Then, the output vector of the mean pooling is passed to the residual fully connected layers to obtain the predicted time of arrival. The results are shown in Table III ( $t(s)=240$ ). The MAPE increases to 7.84%. Thus, the temporal attention mechanism can enhance predictive performance.

We eliminate the weather layer to further analyze the effect of convective weather, the results are shown in Table III. The MAPE under this setting is 7.47%, compared with the 6.91% of the original model ( $t(s)=240$ ). Thus, the convective weather features extracted from the radar echo maps can improve the prediction performance. In order to analyze the impact of the spatial attention mechanism, we extract features from the radar echo maps only by CNN. The MAPE is 7.34%, which indicates that the spatial attention mechanism can improve the performance by 0.43%.

We eliminate exactly one additional factor from the model to analyze its impact on the prediction performance. The results are shown in Table III ( $t(s)=240$ ). We find that eliminating the distance to the center of the TMA, the TMA congestion degree, and the pattern affects the MAPE growth by 0.83%, 0.66%, and 0.52% respectively. In addition, eliminating the day of month and time of day results in MAPE growth of 0.10%. Therefore, among all the additional factors, the distance to the center of the TMA, the TMA congestion degree, and the pattern are the dominant factors.

TABLE III  
PERFORMANCE OF STNN-ETA

Model	MAE (s)	MAPE (%)
STNN-ETA	70.06	6.91
STNN-ETA with mean pooling	77.77	7.84
STNN-ETA without spatial attention	73.58	7.34
STNN-ETA without convective weather	73.96	7.47
STNN-ETA without distance	76.25	7.74
STNN-ETA without TMA congestion degree	74.92	7.57
STNN-ETA without pattern	73.78	7.43

In order to measure the performance of the model under different conditions, we divide the congestion degree into four group : 1)  $< 25\%C_{max}$  , 2)  $25\%C_{max} - 50\%C_{max}$  , 3)  $50\%C_{max} - 75\%C_{max}$ , 4)  $> 75\%C_{max}$ ,  $C_{max}$  is the maximum of the congestion degree. The percentage of flights in the four groups are 7.7%, 23.1%, 61.0%, 8.2%. Fig. 7 shows that DBSCAN+MCNN, RF, XGBoost, AdaBoost have similar prediction performance under different conditions. However,

the model we proposed can make a better prediction under the congested condition.

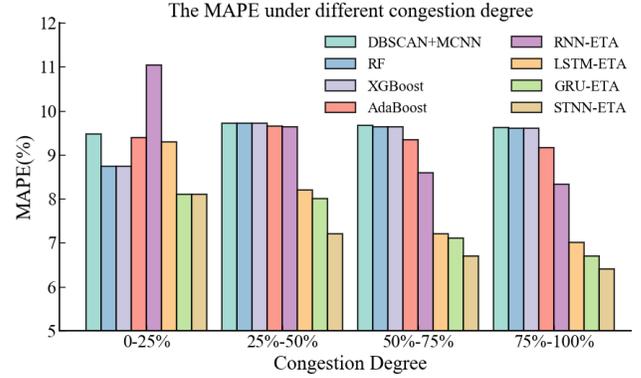


Fig. 7. The MAPE under different congestion degree

## V. CONCLUSION

In this paper, we propose a spatial-temporal neural network model for estimating the time of arrival (STNN-ETA). Three components are discussed. In the trajectory pattern recognition component, DBSCAN is applied to cluster the historical trajectories into several main patterns. Given the target flight, the trajectory from the current position to the runway is predicted by the trajectory prediction component. In the arrival time prediction component, a nonlinear function and Bi-LSTM are proposed to extract the spatial and temporal features from trajectories. In addition, we employ a temporal attention mechanism to adaptively learn the weights for hidden state sequence output from the Bi-LSTM to predict the ETA. Moreover, STNN-ETA considers the effects of convective weather and additional factors. We evaluate our model on the Beijing TMA dataset, and the experimental results show that STNN-ETA significantly outperforms the existing models. We compare the performance of STNN-ETA at different time horizons and find that the mean absolute error tends to be smaller when the time to the destination is shorter. In addition, we analyze the effect of convective weather and additional factors.

In the future, we will further optimize the approach for better prediction performance and model interpretability. Particularly, additional efforts will be made to improve the trajectory prediction. In addition, other important features, e.g., the aircraft types, and the sequencing of arrivals and departures at the airports have an effect on the arrival time of flights, we will incorporate these features into model to enhance the prediction accuracy. What's more, this work focuses on presenting a new model for short-term ETA prediction, and we will extend our model to solve the long-term ETA prediction problem.

## ACKNOWLEDGEMENT

This paper is supported by National Key R&D Program of China (Grant No.2019YFF0301400) and National Natural Science Foundation of China (Grant Nos.61961146005 and 62088101).

## REFERENCES

- [1] M. S. Ryerson, M. Hansen and J. Bonn. "Time to burn: Flight delay, terminal efficiency, and fuel consumption in the National Airspace System," *Transp. Res. Pt. A-Policy Pract.*, vol. 69, pp. 286-298, Nov 2014.
- [2] B. Yu, Z. Guo, S. Asian, et al. "Flight delay prediction for commercial air transport: A deep learning approach," *Transp. Res. Pt. E-Log*, vol. 125, pp. 203-221, 2019.
- [3] L. Jin, Y. Cao and D. Sun. "Investigation of potential fuel savings due to continuous-descent approach," *J. Aircr.*, vol. 50, no. 3, pp. 807-816, 2013.
- [4] Y. Jie, C. Hui, L. Xingyu, et al. "Research on estimated time of arrival prediction based upon ADS-B and spatiotemporal analysis," in *Proc. IEEE 1st Int. Conf. Civ. Aviat. Saf. Inf. Technol., ICCASIT*, 2019: IEEE, pp. 630-634.
- [5] A. Akincilar. "A methodology for shuttle scheduling in airports that ensures mitigating arriving passenger congestion under uncertain demand," *IEEE Intell. Transp. Syst. Mag.*, 2021. DOI: 10.1109/MITS.2021.3049359.
- [6] M. Chen, W. Zeng, Z. Xu, et al. "Delay prediction based on deep stacked autoencoder networks," in *Proc. Asia-Pacific Conf. Intell. Med. Int. Conf. Transp. Traffic Eng*, 2018, pp. 238-242.
- [7] L. L. Pan, K. P. Bowman, E. L. Atlas, et al. "The stratosphere-troposphere analyses of regional transport 2008 experiment," *Bull. Amer. Meteorol. Soc.*, vol. 91, no. 3, pp. 327-342, 2010.
- [8] M. Zhang. "Analysis of traffic delay in terminal area under short-term weather changes," M.S. thesis, Mar., NUAU, Nanjing, China, 2019.
- [9] A. Mukherjee and M. Hansen. "A dynamic rerouting model for air traffic flow management," *Transp. Res. Pt. B-Methodol.*, vol. 43, no. 1, pp. 159-171, 2009
- [10] J. Chen, L. Chen and D. Sun. "Air traffic flow management under uncertainty using chance-constrained optimization," *Transp. Res. Pt. B-Methodol.*, vol. 102, pp. 124-141, 2017.
- [11] Y. Su, K. Xie, H. Wang, et al. "Airline disruption management: A review of models and solution methods," *Engineering*, 2021.
- [12] P. T. Wang, C. R. Wanke and F. P. Wieland. "Modeling time and space metering of flights in the national airspace system," in *Proc. Winter Simul. Conf.*, Washington, DC, USA, 2004: IEEE, pp. 1299-1304.
- [13] K. Roy, B. Levy and C. Tomlin. "Target tracking and estimated time of arrival (ETA) prediction for arrival aircraft," in *AIAA Guid., Navig., Control Conf. Exhib.*, 2006.
- [14] J. Wei, J. Lee and I. Hwang. "Estimated time of arrival prediction based on state-dependent transition hybrid estimation algorithm," in *AIAA Guid., Navig., Control Conf.*, Kissimmee, FL, USA, Jan. 5-9, 2015.
- [15] X. Bai, L. A. Weitz and S. Priess. "Evaluating the impact of estimated time of arrival accuracy on interval management performance," in *AIAA Guid. Navig. Control. Conf.*, San Diego, CA, Jan. 4-8, 2016.
- [16] Z. Wang, M. Liang and D. Delahaye. "Automated data-driven prediction on aircraft estimated time of arrival," *J. Air Transp. Manag.*, vol. 88, pp. 101840, Sept. 2020.
- [17] V. P. Thai, W. Zhong, T. Pham, et al. "Detection, tracking and classification of aircraft and drones in digital towers using machine learning on motion patterns," in *Integr. Commun., Navig. Surveill. Conf., ICNS*, Herndon, VA, USA, 2019: IEEE, pp. 1-8.
- [18] M. Zhang, S. Chen, L. Sun, et al. "Characterizing flight delay profiles with a tensor factorization framework," *Engineering*, 2021.
- [19] Q. Cai, S. Alam and V. N. Duong. "A spatial-temporal network perspective for the propagation dynamics of air traffic delays," *Engineering*, vol. 7, no. 4, pp. 452-464, 2021.
- [20] G. Zhong, T. Yin, L. Li, et al. "Bus travel time prediction based on ensemble learning methods," *IEEE Intell. Transp. Syst. Mag.*, May. 2020. DOI: 10.1109/MITS.2020.2990175.
- [21] T. Pamula. "Road traffic conditions classification based on multilevel filtering of image content using convolutional neural networks," *IEEE Intell. Transp. Syst. Mag.*, vol. 10, no. 3, pp. 11-21, 2018.
- [22] T. Zhang, G. Guo. "Graph attention LSTM: A spatio-temporal approach for traffic flow forecasting," *IEEE Intell. Transp. Syst. Mag.*, Jun. 2020, DOI: 10.1109/MITS.2020.2990165.
- [23] Y. Glina, R. Jordan and M. Ishutkina. "A tree-based ensemble method for the prediction and uncertainty quantification of aircraft landing times," in *Amer. Meteorol. Soc. Conf. Arti. Intell. Appl. Environ. Sci., New Orleans, LA*, 2012
- [24] G. Takacs. "Predicting flight arrival times with a multistage model," in *Proc. - IEEE Int. Conf. Big Data, IEEE Big Data*, Washington, DC, USA, Oct. 27-30, 2014, pp. 78-84.
- [25] C. S. Kern, I. P. de Medeiros and T. Yoneyama. "Data-driven aircraft estimated time of arrival prediction," in *Annu. IEEE Int. Syst. Conf., SysCon - Proc.*, Vancouver, BC, Canada, Apr. 13-16, 2015, pp. 727-733.
- [26] A. De Leege, M. van Paassen and M. Mulder. "A machine learning approach to trajectory prediction," in *AIAA Guidance, Navigation, Control (GNC) Conf.*, Boston, MA, USA, Aug. 19-22, 2013.
- [27] G. Zhu, C. Matthews, P. Wei, et al. "En route flight time prediction under convective weather events," in *Aviat. Technol., Integr., Op. Conf.*, Atlanta, GA, USA, Jun. 25-29, 2018.
- [28] S. Ayhan, P. Costas and H. Samet. "Predicting estimated time of arrival for commercial flights," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, London, U.K., Jul. 19-23, 2018, pp. 33-42.
- [29] M. H. Nguyen and S. Alam. "Airspace collision risk hot-spot identification using clustering models," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 1, pp. 48-57, 2017.
- [30] D. Delahaye, S. Puechmorel, S. Alam, et al. "Trajectory mathematical distance applied to airspace major flows extraction," in *ENRI Int. Workshop ATM/CNS*, Springer, Singap., 2017, pp. 51-66.
- [31] H. Arneson, A. Bombelli, A. Segarra-Torné, et al. "Analysis of convective-weather impact on pre-departure routing decisions for flights traveling between Fort Worth Center and New York Air Center," in *Aviat. Technol., Integr., Oper. Conf.*, Denver, CO, USA, Jun. 5-9, 2017, pp. 3593.
- [32] N. Alisoltani, M. Zargayouna, L. Leclercq. "A sequential clustering method for the taxi-dispatching problem considering traffic dynamics," *IEEE Intell. Transp. Syst. Mag.*, vol. 12, no. 4, pp. 169-181, 2020.
- [33] Z. Wang, M. Liang and D. Delahaye. "A hybrid machine learning model for short-term estimated time of arrival prediction in terminal manoeuvring area," *Transp. Res. Part C Emerg. Technol.*, vol. 95, pp. 280-294, 2018.
- [34] S. Trivedi, Z. A. Pardos and N. T. Heffernan. "The utility of clustering in prediction tasks," *ArXiv preprint ArXiv*, 2015.
- [35] S. Hong and K. Lee. "Trajectory prediction for vectored area navigation arrivals," *J. Aerosp. Inf. Syst.*, vol. 12, no. 7, pp. 490-502, 2015.
- [36] Electronic Aeronautical Information Publication of People's Republic of China, Aeronautical Information Center, ATMB, CAAC, 2017.
- [37] M. C. R. Murça, R. J. Hansman, L. Li, et al. "Flight trajectory data analytics for characterization of air traffic flows: A comparative analysis of terminal area operations between New York, Hong Kong and Sao Paulo," *Transp. Res. Part C Emerg. Technol.*, vol. 97, pp. 324-347, 2018.
- [38] M. Ester, H. P. Kriegel, J. Sander et al. "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, 1996, pp. 226-231
- [39] H. Duan, X. Wang, Y. Bai, et al. "Integrated approach to density-based spatial clustering of applications with noise and dynamic time warping for breakout prediction in slab continuous casting," *Metall. Trans. B*, vol. 50, no. 5, pp. 2343-2353, 2019.
- [40] P. Senin. "Dynamic time warping algorithm review," *Inf. Comput. Sci. Dept. UHM, USA*, 2008.
- [41] Y. Liu and M. Hansen. "Predicting aircraft trajectories: A deep generative convolutional recurrent neural networks approach," *ArXiv preprint ArXiv*, 2018.
- [42] Y. Gal and Z. Ghahramani. "A theoretically grounded application of dropout in recurrent neural networks," *Advances Neural Inf. Process. Syst.*, vol. 29, pp.1019-1027, 2016.
- [43] V. Deshpande, and M. Arikan. "The impact of airline flight schedules on flight delays," *Manuf. Serv. Oper. Manag.*, vol. 14, no. 3, pp. 423-440, 2012.
- [44] H. Wang. "Study on flight efficiency of arrivals of Beijing Capital Airport," M.S. thesis, May., CAUC, Tianjin, China, 2018.
- [45] S. Hochreiter, Y. Bengio, P. Frasconi, et al. "Gradient flow in recurrent nets: The difficulty of learning long-term dependencies," 2001.
- [46] A. Graves, and J. Schmidhuber. "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," *Neural Netw.*, vol. 18, no. 5-6, pp. 602-610, 2005.
- [47] K. He, X. Zhang, S. Ren, et al. "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2016, pp. 770-778.
- [48] T. Chen and C. Guestrin. "Xgboost: A scalable tree boosting system," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, San Francisco, CA, USA, Aug. 13-17, 2016.
- [49] J. Friedman, T. Hastie and R. Tibshirani. "Additive logistic regression: A statistical view of boosting (with discussion and a rejoinder by the authors)," *Ann. Statist.*, vol. 28, no. 2, pp. 337-407, 2000.



**Yan Ma** received the B.S. degree from the College of Electronic and Information Engineering, Beihang University, Beijing, China, in 2019. She is currently pursuing the M.S. degree in electronic and communication engineering at Beihang University, Beijing, China. Her current research interests include ETA prediction and machine learning.



**Yisheng Lv** is an Associate Professor at the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Science. He is also with the University of Chinese Academy of Science. His research interests are artificial intelligence for transportation, intelligent vehicles, and parallel traffic management and control systems.



**Wenbo Du** (M'17) received the B.S. and Ph.D. degrees from the School of Computer Science and Technology, University of Science and Technology of China, Hefei, China, in 2005 and 2010, respectively.

He is a Professor with the School of Electronic and Information Engineering, Beihang University, Beijing, China. His current research interests include data science and intelligent transportation.



**Xianbin Cao** (M'08-SM'10) received the B.Eng and M.Eng degrees in computer applications and information science from Anhui University, Hefei, China, in 1990 and 1993, respectively, and the Ph.D. degree in information science from the University of Science and Technology of China, Hefei, in 1996.

He is currently a Professor with the School of Electronic and Information Engineering, Beihang University, Beijing, China. His current research interests include intelligent transportation systems, air traffic management, and intelligent computation.



**Jun Chen** received the B.Sc. degree in electrical engineering and automation from the Nanjing University of Science and Technology, Nanjing, China, and the M.Sc. degree in software engineering from Tongji University, Shanghai, China. He received the second M.Sc. (with distinction) and Ph.D. degrees in systems engineering and control from The University of Sheffield, Sheffield, U.K.

He is currently a Senior Lecturer in Engineering Science at Queen Mary University of London, London, U.K. He has published more than 60 scientific papers in areas of multi-objective optimization, interpretable fuzzy systems, data-driven modelling, and intelligent transportation systems. From 2020, he serves as a full member of the EPSRC Peer Review College. He is also a Turing Fellow at Alan Turing Institute.



**Yu Zhang** received the B.S. degree in transportation engineering from Southeast University, Nanjing, China, the M.S. and Ph.D. degrees in civil and environmental engineering from the University of California Berkeley, Berkeley, CA, USA.

She is currently an Associate Professor with the Department of Civil and Environmental Engineering at the University of South Florida, Tampa, FL, USA. Her current research interests are: Transportation system modeling, analysis, and simulation; Resilient system design and operations; Air transportation and global airline industry; Multimodal transportation planning and sustainable transportation.