A Deep Learning Approach for Flight Delay Prediction through Time-Evolving Graphs

Kaiquan Cai, Yue Li, Yiping Fang, Yanbo Zhu

A Deep Learning Approach for Flight Delay Prediction through Time-Evolving Graphs

Kaiquan Cai, *Member, IEEE,* Yue Li, Yi-Ping Fang, *Member, IEEE,* and Yanbo Zhu, *Member, IEEE*

***Abstract*—Flight delay prediction has recently gained growing popularity due to the significant role it plays in efficient airline and airport operation. Most of the previous prediction works consider the single-airport scenario, which overlooks the time-varying spatial interactions hidden in airport networks. In this paper, the flight delay prediction problem is investigated from a network perspective (i.e., multi-airport scenario). To model the time-evolving and periodic graph-structured information in the airport network, a flight delay prediction approach based on the graph convolutional neural network (GCN) is developed in this paper. More specifically, regarding that GCN cannot take both delay time-series and time-evolving graph structures as inputs, a temporal convolutional block based on the Markov property is employed to mine the time-varying patterns of flight delays through a sequence of graph snapshots. Moreover, considering that unknown occasional air routes under emergency may result in incomplete graph-structured inputs for GCN, an adaptive graph convolutional block is embedded into the proposed method to expose spatial interactions hidden in airport networks. Through extensive experiments, it has been shown that the proposed approach outperforms benchmark methods with a satisfying accuracy improvement at the cost of acceptable execution time. The obtained results reveal that deep learning approach based on graph-structured inputs have great potentials in the flight delay prediction problem.**

***Index Terms*—Flight delay prediction, time-evolving airport network, graph-structured information, graph convolutional neural network.**

I. Introduction

AIR transportation is of great significance in business and tourism, achieving a record high to accommodate near 4.5 billion passengers worldwide in 2019 [1]. According to the statistics in the same year, the average delay per flight was 13.1 minutes in Europe and 12.9 minutes in the United States, and the number in China was around 14 minutes [2]–[4]. These delays have led into inevitable consequences such as unpleasant passenger experiences, followed by economic losses of relevant airspace users. The annual cost of flight delays to the global economy was estimated to be $50 billion in 2019 [5]. Such high loss motivates the analysis of air traffic delays and the development of more advanced flight delay prediction approaches in both industry and academia [6].

In flight delay prediction, mathematical simulations and data-driven methods are currently two types of representative work. The mathematical simulation methods [7]–[9] employ mathematical tools to model air traffic operations and usually require huge computational resources. Moreover, there are typically some impractical assumptions and/or simplifications made in these methods, which make it more challenging for these methods to be used in actual situations and greatly decrease the prediction accuracy [10]. In recent decades, data-driven methods have gained extensive attention owing to the availability of massive air traffic data [11]–[13]. In detail, Hao et al. [14] developed a regression model combined with an econometric method to predict flight delays at three major commercial airports in New York. Chen et al. [15] utilized a multi-label random forest classification method combined with air traffic operation to predict sequence future delays for individual flights along their scheduled itineraries. Yu et al. [16] proposed a deep belief network method to mine the inner patterns of flight delays within a set of micro influential factors and presented a practical flight delay prediction at Beijing Capital International Airport.

However, above-mentioned works only concerned on the single-airport scenario and neglected dynamic spatial interactions hidden in airport networks. Indeed, flight delays can randomly take place within air transportation system and readily propagate throughout the tightly connected airport network owing to the large number of interconnected resources (e.g., aircraft, flight crews, passenger, and infrastructure) [17]–[19]. Therefore, it is necessary to focus on the dynamic spatial interactions among connected airports, i.e., graph-structured information.

Recent years have witnessed a growing interest in understanding the spatial dependencies in flight delay prediction problem [20]–[22]. Specifically, Pyrgiotis et al. [23] developed network decomposition models to study the complex phenomenon of the propagation of delays and provided several encouraging consequences. Similarly, Du et al. [17] applied a delay causality network to study the propagation mechanism of flight delays within a large-scale airport network based on the interdependence of delay time-series. Recently, Wu and Law [24] developed a Bayesian network in the delay-tree framework to examine complicated delay propagation effects in an airline network.

In spite of the advances in understanding the spatial dependencies among airports in flight delay prediction problem, previous works mostly concentrate on analyzing the qualitative

K. Cai and Y. Li are with the School of Electronic and Information Engineering, Beihang University, Beijing 100191, China, and also with the National Key Laboratory of CNS/ATM, Beijing 100191, China (e-mail: caikq@buaa.edu.cn; leeyue@buaa.edu.cn).

Y. Fang is with the Laboratoire Génie Industriel, CentraleSupélec, Université Paris-Saclay, 3 Rue Joliot Curie, 91190 Gif-sur-Yvette, France (e-mail: yiping.fang@centralesupelec.fr).

Y. Zhu is with the School of Electronic and Information Engineering, Beihang University, Beijing 100191, China, and also with the Aviation Data Communication Corporation, Beijing 100191, China (e-mail: zhuyanbo@buaa.edu.cn).

influence of flight delays on airport networks and may not be appropriate for quantitative prediction within dynamic spatial interactions.

Fortunately, utilizing deep learning methods to capture spatial and temporal dependency, timely and accurate traffic prediction has gained growing attention in transportation management area due to the significant benefits it might bring to traffic control and guidance [25]–[28]. More specifically, the graph convolutional neural network (GCN), with the capability of extracting complex non-linear relationships in general graphs, brings opportunities in handling complicated traffic forecasting problems with the consideration of graph-structured information [29]–[32]. Recently, based on GCN models, researchers have proposed a series of intelligent methods to provide quantified diagnostics for ground transportation [33]–[35]. Yu et al. [36] utilized a recurrent neural network to model the sequential data and developed a deep neural network based on long short term memory units for traffic forecasting. To model the non-linear temporal dependency of traffic, Pan et al., [37] proposed a meta recurrent neural network combined with an encoder-decoder architecture to predict urban traffic. Zhang et al. [38] improved GCN by applying the attention mechanism to aggregate features from the neighbours of a node and developed a deep learning framework to predict traffic speed.

Inspired by the recent success of GCN in various prediction tasks for ground transportation, this paper addresses the flight delay prediction problem by abstracting it as a time-series analysis task based on graph-structured information of an airport network. And an improved GCN is utilized to capture the highly meaningful patterns of flight delays in the multi-airport scenario. Due to the dynamic nature of airport networks, two critical problems need to be overcome:

- The structure of an airport network may evolve over time in a day, e.g., scheduled air routes for rush or slack hours; flight cancellations due to accidents or natural disasters. However, the conventional GCN cannot handle both delay time-series and time-evolving graph structures simultaneously.
- To address urgent requirements (e.g., military activity and mechanical failure) in daily operation, the air traffic management department might develop temporary air routes to mitigate the severe effects of emergencies. These temporary air routes are quite ad hoc and typically unknown to flight delay prediction models, which results in incomplete graph-structured inputs for GCN.

To address the above problems, this paper proposes a novel flight delay prediction method by jointly modeling the spatio-temporal features of an airport network. Specifically, a deep learning architecture with graph-structured inputs, named Multiscale Spatial-Temporal Adaptive Graph Convolutional Neural Network (MSTAGCN), is designed to resolve the flight delay prediction problem for multi-airport scenario in this paper. Considering the time-varying graph-structured inputs for GCN, a temporal convolutional block based on the Markov property is developed to capture the temporal dependency of air traffic through a sequence of graph snapshots. Moreover, to capture the unknown newly-formed air routes under urgent requirements, an adaptive graph convolutional block is embedded into the architecture, which parameterizes two types of graphs (namely, importance and similarity graphs) based on predefined structure of an airport network. These parameterized graphs are trained and updated jointly with convolutional parameters of the model. Overall, the proposed flight delay prediction approach can be distinguished from the prior works in the following two aspects:

- To model the dynamic spatial interactions hidden in an airport network, the flight delay prediction problem is investigated from a network perspective (i.e., multi-airport scenario).
- Considering unknown newly-formed air routes under urgent requirements, an adaptive graph convolutional network component is developed to learn complete graph-structured inputs for GCN.

This paper is organized as follows. In Section 2, we formulate the flight delay prediction problem and elaborate on the details of time-evolving nature of airport networks. Section 3 describes the methodology, including the deep learning architecture, the temporal convolutional block, and the adaptive graph convolutional block. In Section 4, the proposed model is compared with several benchmark approaches. Section 5 reports the case study of the Chinese airport network. The final section concludes the paper with some brief remarks.

II. Problem formulation

We define a set of nodes as $V = \{v_1, v_2, ..., v_n\}$ to represent all airports in an airport network. At time t, the airport network can be described as a directed and weighted network $G^{(t)} = (V, E^{(t)}, W^{(t)})$, where $E^{(t)}$ is a set of edges representing air routes among nodes in V. $W^{(t)} \in \mathbb{R}^{n\times n}$ denotes the weighted adjacency matrix of $G^{(t)}$, with its element $w_{ij}^{(t)}$ representing the number of flights between airport v_i and airport v_j during $(t-1,t)$. Moreover, $Y^{(t)} = \{y_1^{(t)}, y_2^{(t)}, ..., y_n^{(t)}\} \in \mathbb{R}^n$ is employed to denote an observation vector of n airports at time t, of which each element $y_i^{(t)}$ records the delay experienced at a single airport i during $(t-1,t)$. For airport i, in addition to the actual flight delays, cancellations should be considered as a delay metric to assess the on-time performance of an airport. Therefore, according to the regulations of the Federal Aviation Administration (FAA), the Eurocontrol's Network Manager (NM) Operations Centre, and the Civil Aviation Administration of China (CAAC), ρ is utilized to indicate the equivalent delay time of a cancellation. Then, the weighted value of the actual flight delays and the equivalent delays of cancellations is used to represent the average delay $y_i^{(t)}$ of an airport.

$$y_i^{(t)} = \frac{m_i^{(t)} + \rho * c_i^{(t)}}{a_i^{(t)}}, \tag{1}$$

where $m_i^{(t)}$ denotes the total delay of departure flights at airport i during $(t-1,t)$, $c_i^{(t)}$ and $a_i^{(t)}$ reveal the number of cancelled flights and scheduled departure flights at airport i during $(t-1,t)$, respectively, and $\rho = 180$ represents the equivalent delay of a cancellation [17].

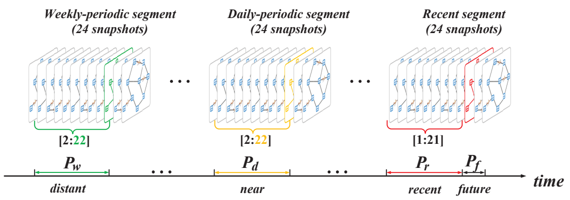


Fig. 1. Illustration of constructing the multiscale historical delay sequences based on graph snapshots.

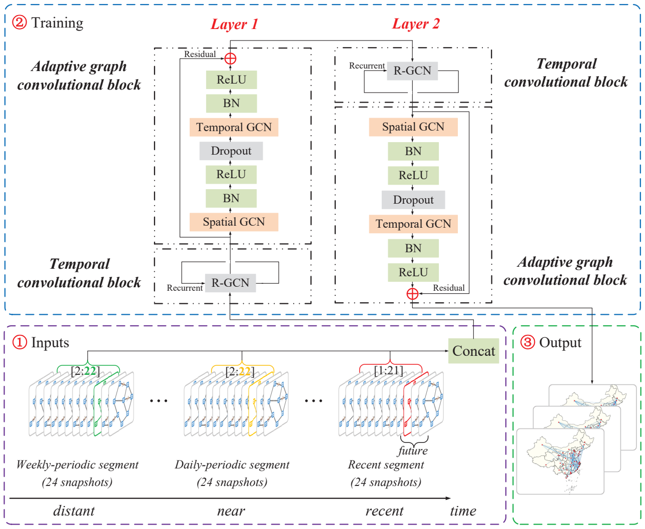


Fig. 2. System architecture of Multiscale Spatial-Temporal Adaptive Graph Convolutional Network.

We denote the adjacency matrix $W^{(t)}$ and the delay vector $Y^{(t)}$ as a graph snapshot at time t [39]. A sequence of graph snapshots $\boldsymbol{W} = \{W^{(0)}, W^{(1)}, ..., W^{(t)}\}$ and corresponding delay observations $\boldsymbol{Y} = \{Y^{(0)}, Y^{(1)}, ..., Y^{(t)}\}$ is regarded as a time-evolving graph in this paper. Note that the structure of an airport network can evolve over time in a day, and the evolution presents obvious periodic patterns. In fact, a flight schedule in air transportation is typically composed of two different flight seasons: Winter-Spring season and Summer-Autumn season. In each season, all carriers perform similar daily flight plans except for weekends, thus air traffic and flight delays at each airport show similar daily patterns. Furthermore, due to the regular flight timetables, air traffic and flight delays also exhibit weekly periodicity, e.g., air traffic pattern on the latest Tuesday has a strong similarity with the air traffic on Tuesdays in history. To take advantage of the long-term periodicity of air traffic, we formulate the multiscale historical delay sequences $X^{(t)} = \{P_w^{(t)}, P_d^{(t)}, P_r^{(t)}\}$ at time t as shown in Fig. 1, where $P_w^{(t)} \in \mathbb{R}^{N \times n}$, $P_d^{(t)} \in \mathbb{R}^{N \times n}$, and $P_r^{(t)} \in \mathbb{R}^{N \times n}$ represent the weekly-periodic segment, the daily-periodic segment and the recent segment, respectively. The details of the three delay segments are described as follows:

$$P_r^{(t)} = \{Y^{(t-N)}, Y^{(t-N+1)}, ..., Y^{(t-1)}\}, \tag{2}$$

$$P_d^{(t)} = \{Y^{(t-N-24+1)}, Y^{(t-N+1-24+1)}, ..., Y^{(t-1-24+1)}\}, \tag{3}$$

$$P_w^{(t)} = \{Y^{(t-N-7*24+1)}, Y^{(t-N+1-7*24+1)}, ..., Y^{(t-1-7*24+1)}\}, \tag{4}$$

where the time t is represented by the red graph snapshot in Fig. 1. The recent delay sequence $P_r^{(t)}$, including N graph snapshots before time t during the same day and the day before if $t \leq (N-1)h$, records the neighboring graph snapshots which have inevitable influences on future traffic. The daily-periodic delay sequence $P_d^{(t)}$ is consisted of N graph snapshots tracing back from (and including) the predicted period in the day before, representing the daily periodicity of air traffic. The weekly-periodic delay sequence $P_w^{(t)}$, including N graph snapshots tracing back from (and including) the predicted period in the same day of the previous week, exposes the weekly periodicity of air traffic. For example, assuming that $N = 21$ and the time t denotes a period from 16:00h to 17:00h on a Wednesday. Then, the recent delay sequence records the historical delays of all airports between 19:00h (Tuesday) and 16:00h (Wednesday); The historical delays of all airports between 20:00h (Monday) and 17:00h (Tuesday) are poured into the daily-periodic delay sequence; The weekly-periodic delay sequence consists of delays of all airports between 20:00h (the last Tuesday) and 17:00h (the last Wednesday). Through extensive experiments, $N = 21$ is chosen in our case study.

In this paper, the flight delay prediction problem is converted to a time-series analysis task on time-evolving airport networks and we aim to predict the most likely delays for each airport in the next M time steps given the air traffic observations in the historical $3 * N$ time steps (Note: the past $3*N$ hours are chosen no matter they are in the same calendar day or not). This procedure can be formulated as

$$[X^{(t)}] \rightarrow [Y^{(t)}, ..., Y^{(t+M-1)}], \tag{5}$$

III. Methodology

A. Overview of the proposed model

The proposed model (i.e., MSTAGCN) is composed of two multiscale spatial-temporal adaptive graph convolutional layers and a residual connection is added for each layer to stabilize the training procedure, as shown in Fig. 2. Each multiscale spatial-temporal adaptive graph convolutional layers is consisted of a temporal convolutional block and an adaptive spatial convolutional block.

MSTAGCN takes a time-evolving airport network (i.e., a sequence of graph snapshots) and the multiscale historical delay sequences as inputs. After stacking two multiscale spatial-temporal adaptive graph convolutional layers, we apply a fully connected output layer at the final step and the prediction results are generated. Moreover, the Mean Squared Error is employed as the loss function to evaluate the performance of the proposed approach.

B. A temporal convolutional block for mining the time-varying patterns of flight delays

An airport network (a directed graph) can be considered as a multi-relational graph with incoming and outgoing relations, and the relational GCN (R-GCN) is capable of handling multi-relational graphs. So, we first employ R-GCN to model the spatial interactions of each airport in a single graph snapshot. Then, the temporal dynamics of flight delays between two adjacent graph snapshots is investigated in accordance with the Markov property. Based on above observations, we generalize R-GCN to process the time-evolving graph.

Considering a single graph snapshot, the adjacency matrix $W^{(t)}$ and the multiscale historical delay sequences $X^{(t)}$ are taken as inputs. We define the notion of graph convolution operator $\star\mathcal{G}$ based on spectral graph convolution, as the multiplication of the multiscale delay sequences $X^{(t)}$ with a kernel Θ_F:

$$\Theta_F \star \mathcal{G} X^{(t)} = \sigma(\sum_{r\in\mathcal{R}} (D_r^{(t)})^{-1} W_r^{(t)} X^{(t)} M_r + X^{(t)} M_0), \tag{6}$$

where Θ_F denotes the parameter set used in a single graph snapshot, $\sigma(\cdot)$ represents the activation function. $\mathcal{R} = \{in, out\}$ reveals a set of relation type, $W_{in}^{(t)} = W^{(t)}$ denotes the incoming relation and $D_r^{(t)}$ is the diagonal degree matrix with $(D_r^{(t)})_{ii} = \sum_j (W_r^{(t)})_{ij}$, $W_{out}^{(t)} = (W^{(t)})^T$ represents the outgoing relation. And M_{in}, M_{out} reveal the weight matrix for incoming and outgoing relations, respectively. M_0 represents the self-connection weight matrix [40]. Moreover, to further generalize R-GCN and prevent overfitting, a linear combination of incoming and outgoing relations is utilized to simplify the self-connection weight matrix [39]. The Eq. (6) can then be rewritten as

$$\Theta_F \star \mathcal{G} X^{(t)} = \sigma(\sum_{r\in\mathcal{R}} \widetilde{W}_r^{(t)} X^{(t)} M_r), \tag{7}$$

where $\widetilde{W}_r^{(t)} = (\widetilde{D}_r^{(t)})^{-1}(W_r^{(t)} + I_n)$, $(\widetilde{D}_r^{(t)})_{ii} = \sum_j (W_r^{(t)} + I_n)_{ij}$, and I_n represents a n-dimension identity matrix.

Considering two adjacent graph snapshots at time $t-1$ and t, we take the processed adjacency matrix $\widetilde{W}_r^{(t-1)}, \widetilde{W}_r^{(t)}$ and the multiscale historical delay sequences $X^{(t-1)}, X^{(t)}$ as inputs. Then, the graph convolution operator $\star\mathcal{G}$ for two adjacent graph snapshots can be generalized as follow:

$$\Theta_B \star \mathcal{G}[X^{(t-1)}, X^{(t)}] = \sigma(\Theta_H \star \mathcal{G} X^{(t-1)} + \Theta_F \star \mathcal{G} X^{(t)}), \tag{8}$$

where Θ_B denotes the parameter set used in two adjacent graph snapshots. $\Theta_H \star \mathcal{G} X^{(t-1)}$ represents the graph convolution operation at time $t-1$, and it take $\widetilde{W}_r^{(t-1)}$ and $X^{(t-1)}$ as inputs. Note that the parameter set Θ_H does not change over time. Overall, the graph convolution operation for two adjacent graph snapshots can be viewed as a combination of both current and accumulating previous graph snapshots.

Finally, a hidden state $H^{(t-1)}$ is utilized to memorize the accumulating previous graph snapshots, and a combination of the hidden state and the current input is employed to generate a new hidden state:

$$H^{(t)} = \sigma(\Theta_B \star \mathcal{G}[H^{(t-1)}, X^{(t)}]), \tag{9}$$

where Θ_B includes Θ_H and Θ_F. In these settings, we can handle a time-evolving graph through applying Eq. (9) sequentially.

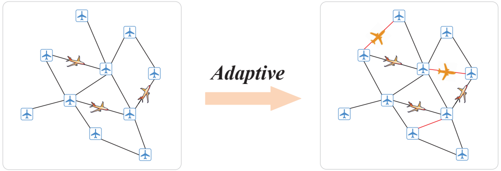


Fig. 3. Illustration of the evolution of the graph convolution operator. Blue point: civil airport. Black line: the scheduled air routes. Red line: the unknown temporary air routes.

C. An adaptive graph convolutional block for capturing unknown spatial interactions hidden in airport networks

The adaptive graph convolutional block is composed of a spatial GCN (Convs), a temporal GCN (Convt), a dropout layer, and a residual connection. Each GCN-based layer is followed by a batch normalization layer (BN) and a ReLU layer (ReLU). Here, the temporal GCN is the same as the model proposed by Shi et al. [41], i.e., it applies a $K_t \times 1$ convolution on the $C_t \times N \times n$ feature map to capture temporal information through an airport network. The spatial GCN for the space domain is described as follows.

The adjacency matrix is defined as A corresponding to the last hidden state $H^{(t_{last})}$ of the temporal convolutional block. As shown in Fig. 2, the adaptive graph convolutional block employs a spatial GCN in the middle to concatenate the temporal convolutional block. Therefore, we can take the adjacency matrix A and the last hidden state $H^{(t_{last})}$ as inputs to the spatial GCN. Then, the graph convolution operator $\star\mathcal{G}$ is defined as the multiplication of the last hidden state $H^{(t_{last})}$ with a kernel Θ_I:

$$\begin{aligned} \Theta_I \star \mathcal{G} H^{(t_{last})} &= \Theta_I(L) H^{(t_{last})} \\ &= \Theta_I(U \Lambda U^T) H^{(t_{last})} \\ &= \Theta_I(I_n - Q^{-1}A) H^{(t_{last})}, \end{aligned} \tag{10}$$

where Θ_I denotes the parameter set used in the spatial GCN modeling. The graph Fourier basis $U \in \mathbb{R}^{n\times n}$ is the eigenmatrix of the random walk Laplacian $L = I_n - Q^{-1}A = U\Lambda U^T \in \mathbb{R}^{n\times n}$, where I_n represents a n-dimension identity matrix, $A \in \mathbb{R}^{n\times n}$ is the adjacency matrix and $Q \in \mathbb{R}^{n\times n}$ is the diagonal degree matrix with $Q_{ii} = \sum_j A_{ij}$, Q^{-1} denotes the inverse matrix of Q, and $\Lambda \in \mathbb{R}^{n\times n}$ indicates the diagonal matrix of eigenvalues of L. To efficiently study flight delays at the system level, the 1^{st}-order Approximation method [42] is adopted in this paper due to its simplicity and proven performance. The Eq. (10) can then be rewritten as

$$\begin{aligned} \Theta_I \star \mathcal{G} H^{(t_{last})} &\approx \theta_0 H^{(t_{last})} + \theta_1(\frac{2}{\lambda_{max}} L - I_n) H^{(t_{last})} \\ &\approx \theta_0 H^{(t_{last})} - \theta_1(Q^{-1}A) H^{(t_{last})}, \end{aligned} \tag{11}$$

where θ_0, θ_1 represent two shared parameters in the kernel Θ_I. To ensure a stabilize numerical performance [10], here we assume that $\theta = \theta_0 = -\theta_1$. The Eq. (11) can then be expressed by

$$\Theta_I \star \mathcal{G} H^{(t_{last})} = \theta(I_n + Q^{-1}A) H^{(t_{last})}, \tag{12}$$

note that the graph convolution operator designed by Eq. (12) concentrates on the scheduled air routes of an airport network, while the unknown temporary air routes for urgent requirements (as shown in Fig. 3) are neglected. To address this problem, an adaptive graph convolutional layer is employed to model the complicated spatial interactions in airport networks under urgent situations. More specifically, the importance and similarity matrixes based on the scheduled structure of an airport network are added and parameterized, and these matrixes are trained and updated jointly with the convolutional parameters of the model. These matrixes are unique for different layers and reveal the critical airports in flight delay propagation. The Eq. (12) can then be rewritten as

$$\Theta_A \star \mathcal{G} H^{(t_{last})} = \theta^*(I_n + Q^{-1}(A + Z + S))H^{(t_{last})}, \quad (13)$$

where Θ_A denotes the parameter set used in the spatial GCN, θ^* represents the shared parameter in the kernel Θ_A. A represents the scheduled structure of an airport network, $Z \in \mathbb{R}^{n\times n}$ shows the importance of each airport in an airport network and plays the same role as the attention mechanism [43]. Moreover, elements in Z are updated and trained together with other parameters based on the training inputs. Therefore, there is no constraint on the value of each element, which allows the generation of temporary connections not existing in the scheduled structure of an airport network. To determine whether there is a connection between two airports and how strong the connection is, the normalized embedded Gaussian function (i.e., Eq. (14)) is employed to measure the similarity of two airports (Note that the similarity function is not symmetric because the airport network is described as a directed network in this paper):

$$f(v_i, v_j) = \frac{\exp(\varphi(v_i)^T)\psi(v_j)}{\sum_{j=1}^{n}\exp(\varphi(v_i)^T)\psi(v_j)}, \quad (14)$$

where $\varphi(\cdot)$ and $\psi(\cdot)$ represent two embedding functions designed to map the original feature map $H^{(t_{last})}$ into two embedded feature maps $S^1 \in \mathbb{R}^{n\times C_e N}, S^2 \in \mathbb{R}^{C_e N\times n}$. Specifically, we firstly embed the original feature map $H^{(t_{last})}$ into $H^{(t_{last})^*} \in \mathbb{R}^{N\times n\times C_e}$ with the two embedding functions. Here, we employ a 1×1 convolutional layer as the embedding function based on extensive experiments. Then, the two embedded feature maps are rearranged to S^1 and S^2, so the similarity matrix S can be defined as the multiplication of S^1 with S^2. Since the normalized embedded Gaussian is equipped with a softmax operation, we can obtain the similarity matrix S based on Eq. (14) as follow:

$$S = softmax(H^{(t_{last})^T}\Theta_\varphi^T\Theta_\psi H^{(t_{last})}), \quad (15)$$

where $\Theta_\varphi, \Theta_\psi$ reveal the parameter set of the embedding function $\varphi(\cdot)$ and $\psi(\cdot)$, respectively.

IV. Experimental results of the Chinese airport network

A. Dataset

The data used in this paper are provided by CAAC, comprising all domestic flights from April 1, 2018 to October 31, 2018 (i.e., Summer-Autumn flight season). Considering the sparse flight schedules at spoke airports, 224 civil airports are ranked according to the handling capacity and the top 74 busy airports are poured into the experimental dataset. The dataset contains 2.19 million scheduled flights connecting 74 critical airports, which serve more than 90% of the air traffic in China. All raw data are normalized by Z-Score method. 70% of the dataset is employed for training, 15% are used for testing while the remaining 15% for validation. All experiments are tested on a Linux cluster (i.e., CPU: Intel (R) Xeon (R) Gold 6126 CPU @ 2.60GHz, GPU: NVIDIA TITAN RTX).

B. Baselines

We compare our model with two types of baseline models: single-airport scenario model and multi-airport scenario model.

(1) Single-airport scenario models

- ARIMA [44]: Auto-Regressive Integrated Moving Average, which is one of the most popular methods in time-series prediction task.
- SVR [45]: Support Vector Regression with Radial Basis Function Kernel, the penalty term is set to 0.1, the number of historical observation is 21.

(2) Multi-airport scenario models

- DCRNN [46]: Diffusion Convolutional Recurrent Neural Network. Both encoder and decoder contain two recurrent layers and there are 64 units in each recurrent layer. The initial learning rate is $1e^{-2}$ with a decay rate of 0.1 after every 10 epochs and early stopping is utilized on the validation dataset. The maximum step of random walk is set to 3. We train our models by minimizing the mean square error using Adam (Adaptive Moment Estimation) optimizer for 50 epochs with batch size 64.
- STGCN [10]: Spatial-temporal Graph Convolutional Network. The channels of three layers in ST-Conv block are 64, 16, 64 respectively. Both the graph convolution kernel size and temporal convolution kernel size are set to 3. The initial learning rate is $1e^{-4}$ with a decay rate of 0.6 after every 5 epochs. We train our models by minimizing the mean square error using Adam (Adaptive Moment Estimation) optimizer for 50 epochs with batch size 64.

C. Evaluation Metrics

The Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE) are employed to evaluate the performance of the models:

$$RMSE = \sqrt{\frac{1}{k}\sum_{i=1}^{k}(\hat{x_i} - x_i)^2}, \quad (16)$$

$$MAE = \frac{1}{k}\sum_{i=1}^{k}|\hat{x_i} - x_i|, \quad (17)$$

$$MAPE = \frac{100\%}{k}\sum_{i=1}^{k}|\frac{\hat{x_i} - x_i}{x_i}|, \quad (18)$$

where k is the number of testing samples. $\hat{x_i}$ and x_i denote the real traffic information and predicted flight delay, respectively.

TABLE I
PERFORMANCE COMPARISON OF DIFFERENT METHODS ON THE CHINESE AIRPORT NETWORK.

M	Metric	ARIMA	SVR	DCRNN	STGCN	**MSTAGCN**
1 hour	RMSE	18.89	12.201±0.471	11.313±0.187	11.288±0.133	**10.371±0.168**
	MAE	12.79	7.154±0.128	6.605±0.095	6.564±0.049	**5.884±0.072**
	MAPE	32.72%	19.31±0.95%	16.68±0.58%	16.42±0.27%	**13.39±0.45%**
2 hours	RMSE	21.841	12.601±0.489	11.554±0.193	11.506±0.151	**10.613±0.174**
	MAE	15.33	7.421±0.143	6.824±0.094	6.773±0.047	**6.123±0.089**
	MAPE	41.51%	20.19±0.98%	17.57±0.61%	17.24±0.32%	**14.90±0.52%**
3 hours	RMSE	24.547	12.761±0.514	11.684±0.191	11.641±0.159	**10.787±0.181**
	MAE	17.811	7.493±0.142	6.963±0.095	6.924±0.054	**6.252±0.093**
	MAPE	47.62%	20.92±0.97%	18.28±0.62%	17.97±0.41%	**15.11±0.56%**

Note: we run each experiment 10 times independently and report the mean and standard deviation.

TABLE II
RESULTS OF COMPARISON AMONG DIFFERENT VARIANTS IN MSTAGCN ON THE CHINESE AIRPORT NETWORK.

M	Metric	MSTAGCN-NS	MSTAGCN-NE	MSTAGCN-NA	**MSTAGCN**
1 hour	RMSE	11.981±0.195	11.182±0.176	10.817±0.124	**10.371±0.168**
	MAE	6.927±0.098	6.467±0.069	6.142±0.041	**5.884±0.072**
	MAPE	18.02±0.52%	16.03±0.47%	14.46±0.22%	**13.39±0.45%**
2 hours	RMSE	12.205±0.198	11.379±0.184	11.006±0.136	**10.613±0.174**
	MAE	7.294±0.095	6.692±0.076	6.331±0.047	**6.123±0.089**
	MAPE	19.17±0.57%	16.92±0.50%	15.57±0.28%	**14.90±0.52%**
3 hours	RMSE	12.563±0.194	11.604±0.197	11.289±0.129	**10.787±0.181**
	MAE	7.581±0.096	6.823±0.087	6.506±0.051	**6.252±0.093**
	MAPE	19.98±0.58%	17.87±0.54%	16.30±0.29%	**15.11±0.56%**

Note: we run each experiment 10 times independently and report the mean and standard deviation.

D. Experimental Setting

MSTAGCN is composed of two multiscale spatial-temporal adaptive graph convolutional layers and a residual connection is added for each layer. Both the graph convolution kernel size and temporal convolution kernel size are set to 5. The time-window is set to 24. The initial learning rate is $1e^{-4}$ with a decay rate of 0.6 after every 5 epochs. We train our models by minimizing the mean square error using Adam (Adaptive Moment Estimation) optimizer for 50 epochs with batch size 64.

Here, we vary the size of graph convolution kernel from 1 to 7 and test the impacts on the prediction performance (MAE) for three different future time periods, as shown in the left panel of Fig. 4. When the size of graph convolution kernel grows from 1 to 5, MAE decreases slightly on three future time periods. However, with increasing from 5 to 7, MAE increases dramatically on all time periods. Therefore, we set graph convolution kernel size as 5, and MSTAGCN can achieve better and stable performance.

In addition, we also vary the size of the time-window from 12 to 24 and test the effect on the prediction performance (MAE) for three different future time periods, as shown in the right panel of Fig. 4. When the size of time-window grows from 12 to 24, MAE declines gradually on three future time periods. Therefore, we set the time-window as 24, and MSTAGCN can achieve better and stable performance.

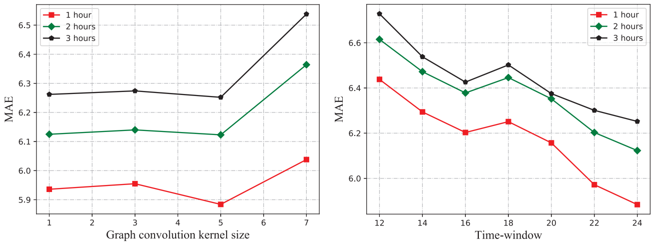


Fig. 4. Impacts of the graph convolution kernel size and time-window size on the prediction performance (MAE) for three different future time periods.

E. Experimental Results

(1) Model Comparison

Table I reports the performance results of the proposed approach and benchmark methods on the testing dataset. We run each experiment 10 times independently and report the mean and standard deviation. Specifically, the first row (i.e., $M = 1$ hour) in Table I shows the short-term prediction results and the long-term forecasting is illustrated by the second and third rows (i.e., $M = 2, 3$ hours). It is shown that our model achieves the best performance in terms of all the statistical metrics both in short-term and long-term predictions. Additionally, as validated by the analysis of variance (ANOVA) [47], there is no significant difference among the predicted results for three different future time periods. Such superior performance of the proposed approach is mainly attributed to the following two points. Firstly, considering the time-

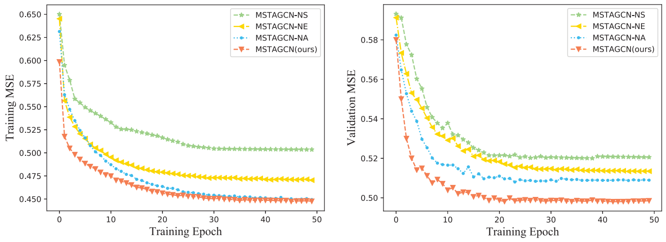


Fig. 5. Training MSE and validation MSE of MSTAGCN variants during the training procedure.

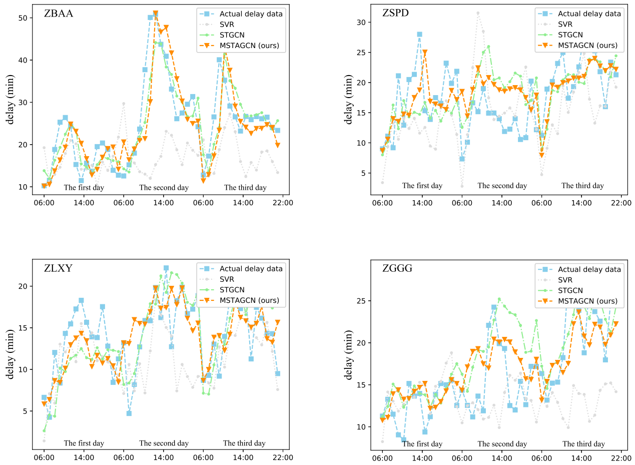


Fig. 6. Predictions of Beijing Capital International Airport (ZBAA), Shanghai Pudong International Airport (ZSPD), Xian Xianyang International Airport (ZLXY), and Guangzhou Baiyun International Airport (ZGGG) for three consecutive days (from $06:00$ to $22:00$).

evolving structure of the airport network, this approach employs the Markov property to capture temporal dependency of air traffic through a sequence of graph snapshots. Furthermore, by parameterizing the additional two graphs based on the scheduled structure of an airport network, the proposed method can effectively explore the latent spatial relationships among airports even if there is no scheduled air route connection.

(2) Variant Comparison

Because the proposed MSTAGCN contains multiple key components, we additionally compare variants of MSTAGCN with respect to the following perspectives to demonstrate the performance of MSTAGCN: 1) the effect of the spatial interactions, 2) the effect of the time-evolving component, and 3) the impact of the adaptive component. The following MSTAGCN variants are designed for comparison.

- MSTAGCN-NS: A variant of MSTAGCN with the spatial interactions component being removed.
- MSTAGCN-NE: A variant of MSTAGCN with the time-evolving component being removed.
- MSTAGCN-NA: A variant of MSTAGCN with the adaptive spatial interactions component being removed.

The ablation study results are shown in Table II and the training MSE and validation MSE of MSTAGCN variants during the training procedure are shown in Fig. 5.

1) Effects of the spatial interactions component: We compare the performance of MSTAGCN with MSTAGCN-NS on a real dataset (descried in Section IV.A) to investigate the effectiveness of the spatial interactions component. From the result, we observe that the proposed MSTAGCN performs better than MSTAGCN-NS which confirms the superiority of introducing the spatial interaction to our model.

2) Effects of the time-evolving component: We compare the performance of MSTAGCN with MSTAGCN-NE on a real dataset (descried in Section IV.A) to investigate the effectiveness of the time-evolving component. According to the result, it is easy to observe that the proposed MSTAGCN achieves better performance in terms of all the evaluation metrics and it also accomplish much faster training and easier convergences in the dataset. Thanks to the presence of the time-evolving component, the proposed architecture achieves a better performance in the long-term predictions.

3) Effects of the adaptive spatial interactions component: We compare the performance of MSTAGCN with MSTAGCN-NA on a real dataset (descried in Section IV.A) to investigate the effectiveness of the adaptive spatial interactions component. According to the result, it is easy to observe that the proposed MSTAGCN outperforms MSTAGCN-NA, which denotes that the adaptive spatial interaction can consistently provide supplementary information to benefit our model.

V. Case Study of the Chinese Airport Network

A. Case Description

We select four representative airports from the Chinese airport network and show their prediction results for three consecutive days, as shown in Fig. 6. More specifically, the SVR model (the gray curve) represents the traditional data-driven methods that overlook the spatial information in an airport network. The STGCN model (the green curve) denotes the deep learning architectures based on static graph-structured inputs. And the proposed MSTAGCN model (the orange curve) represents the deep learning architectures based on evolving graph-structured inputs. In addition, four typical airports are examined in detail including: 1). Beijing Capital International Airport (ZBAA): an aviation hub in northern China which served for a record high of near 100.98 million people in 2018, ranking the first in the Chinese mainland; 2). Shanghai Pudong International Airport (ZSPD): an aviation hub in eastern China. The handling capacity of ZSPD reached 74.05 million passengers in 2018 and it ranked the second in the Chinese mainland; 3). Xian Xianyang International Airport (ZLXY): an aviation hub in western China. It handled 329,700 flights connecting 161 domestic cities and 50 international cities in 2018, and its accessibility of air routes network ranked the second in China; 4). Guangzhou Baiyun International Airport (ZGGG): an aviation hub in southern China. The handling capacity of ZGGG achieved 69.73 million in 2018 and it ranked the third in the Chinese mainland.

B. Prediction Results

From Fig. 6, it is easy to observe that STGCN and MSTAGCN generally perform better than SVR, being able

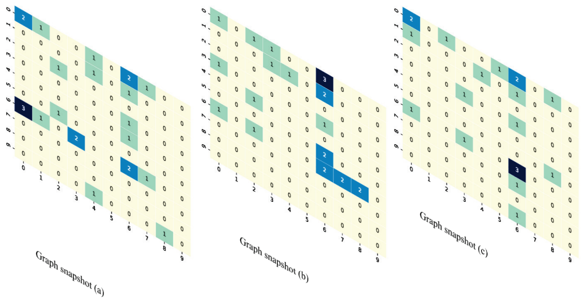


Fig. 7. Illustration of the temporal correlation among adjacent graph snapshots.

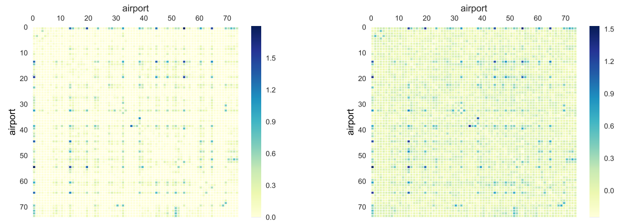


Fig. 8. Illustration of the graph-structured input of MSTAGCN and MSTAGCN-NA. The left matrix represents the scheduled structure of an airport network (MSTAGCN-NA). An example of the corresponding adaptive matrix learned by our model is represented in the right matrix (MSTAGCN).

to effectively capture the dynamic changes from 07:00h to 09:00h and 17:00h to 20:00h. By considering (ignoring) the specific "spatial interactions" in airport networks, the deep learning models (SVR) make more (less) accurate predictions, especially in the morning peak and evening rush hours.

More specifically, the proposed model can meet the practical requirement for long-term prediction and has the potential for short-term forecasting.

- One hour ahead delay predictions: the average error between actual delay and predicted delay is 5.884 ± 0.072 minutes.
- Two hours ahead delay predictions: the average error between actual delay and predicted delay is 6.123 ± 0.089 minutes.
- Three hours ahead delay predictions: the average error between actual delay and predicted delay is 6.252 ± 0.093 minutes.

In practice, the errors of the Airport-Collaborative Decision Making System (A-CDM) powered by Beijing Capital International Airport for one hour, two hours and three hours ahead flight delay predictions are 7.84 minutes, 11.86 minutes and 17.26 minutes, respectively.

Unfortunately, the results of the proposed model present a lag phenomenon compared to the actual delay data in the ZBAA subfigure of Fig. 6. Note that it is the most common phenomenon in time-series analysis tasks based on deep learning approaches [48]–[50]. Moreover, each spot in Fig. 6 represents the delay of an airport during an hour. The lag phenomenon produces a minute-level error (less than 10 minutes), which brings few negative effects to practical flight delay prediction problem.

C. Analysis of Temporal Correlation among Graph Snapshots

To show the advantages of temporal correlation among graph snapshots in greater detail, we compare the graph-structured input of MSTAGCN with MSTAGCN-NE in the graph convolutional layer. We randomly select three adjacent graph snapshots from a time-evolving graph and randomly intercept a portion of them to form a 10×10 adjacency matrix, as shown in Fig. 7. Each element in these matrices denotes the number of flights per unit time between the corresponding origin-destination pair. The key connections among hub airports are represented by the mazarine dots (larger value) and the yellowish dots reveal the weak air routes connections in the airport network. Specifically, if there is no scheduled air route connection between origin-destination pairs, the element would be assigned as zero.

As shown in Fig. 7, there are significant differences in the graph snapshots at different time slots, namely, the graph-structured input of the model changes dramatically over time. Indeed, an airport at time t is not only correlated with other airports at the same time, but also depends on airports at the previous time step due to the large number of interconnected resources. MSTAGCN-NE takes a static graph-structured information as input (i.e., maybe one of the subplots in Fig. 7) and ignores the temporal correlation among graph snapshots. However, the proposed MSTAGCN utilizes a sequence of graph snapshots to model the temporal correlation, which finely captures the changes in an airport network and greatly improves the performance of deep learning models.

D. Analysis of Adaptive Spatial Interactions

To show the advantages of adaptive spatial interactions in greater detail, we compare the graph-structured input of MSTAGCN with MSTAGCN-NA in the graph convolutional layer. As shown in Fig. 8, the adaptive adjacency matrix (in the right panel) not only presents the inherent scheduled air routes but also exposes the unknown temporary relationships among airports under urgent requirements. From a macroscopic perspective, the adaptive matrix reduces the importance gap between hub airports and spoke airports. Namely, it decreases the difference among all elements in the adjacency matrix, which brings more comparability for various inputs and greatly improves the performance of deep learning models.

From a microscopic perspective, the change of each element in the adjacency matrix is calculated to evaluate the alterations of air route strength. Table III reports the top 10 air routes with the greatest reduction of the strength of connections in the Chinese airport network. Although these air routes accommodate a great number of flights in scheduled structure of the airport network, they do not insert flight delays into the air transportation system and play an essential role in flight delay absorption based on superior infrastructures. Take ZGSZ→ZSSS as an example, both airports are capable of handling flight delays based on advanced infrastructures, so the flight delays from the upstream airports may be absorbed by the air route connection between ZGSZ and ZSSS.

TABLE III
TOP 10 AIR ROUTES WITH THE GREATEST REDUCTION OF THE STRENGTH OF CONNECTIONS IN THE CHINESE AIRPORT NETWORK.

Ranking	Air route	The scheduled strength	The strength learned by model	The strength reduction
1	ZGSZ→ZSSS	1.458	1.042	0.417
2	ZBAA→ZSSS	1.792	1.413	0.379
3	ZGGG→ZSSS	1.417	1.054	0.363
4	ZUCK→ZSSS	0.625	0.337	0.288
5	ZGSZ→ZHCC	0.458	0.174	0.284
6	ZBTJ→ZGGG	0.5	0.22	0.28
7	ZSSS→ZBAA	1.792	1.518	0.273
8	ZSSS→ZGSZ	1.292	1.021	0.271
9	ZSSS→ZLXY	0.667	0.397	0.269
10	ZBAA→ZUCK	1	0.734	0.266

Note: the strength reduction represents the difference of connection strength between the scheduled adjacency matrix and the adaptive matrix.

TABLE IV
TOP 10 AIR ROUTES WITH THE GREATEST IMPROVEMENT OF THE STRENGTH OF CONNECTIONS IN THE CHINESE AIRPORT NETWORK.

Ranking	Air route	The scheduled strength	The strength learned by model	The strength improvement
1	ZGZH→ZSQD	0	0.241	0.241
2	ZGDY→ZSWX	0	0.223	0.223
3	ZWWW→ZWKL	0.25	0.47	0.22
4	ZPDL→ZSYT	0	0.215	0.215
5	ZULS→ZSQZ	0	0.21	0.21
6	ZPDL→ZGZJ	0	0.206	0.206
7	ZBNY→ZPDL	0	0.205	0.205
8	ZWSH→ZBYN	0	0.2	0.2
9	ZSYA→ZUGY	0.042	0.241	0.199
10	ZGDY→ZHYC	0	0.198	0.198

Note: the strength improvement represents the difference of connection strength between the scheduled adjacency matrix and the adaptive matrix.

Moreover, Table IV reveals the top 10 air routes with the greatest improvement of the strength of connections in the Chinese airport network. On the one hand, although the air routes in Table IV accommodate few flights in scheduled structure of the airport network (e.g., ZWWW→ZWKL), their poor infrastructures fail to handle flight delays from upstream airports and lead to an increased strength of air route connection. On the other hand, although there is no scheduled connection between these origin-destination pairs (e.g., ZGZH→ZSQD), our model learns a series of non-zero values representing increased strength of these connections. The increased strength may illustrate the importance of neighboring airports in absorbing flights requiring diversion when severe delays or emergency breaks out.

VI. CONCLUSION AND FUTURE WORK

In this paper, we investigate the flight delay prediction problem from a novel perspective and develop a flight delay prediction method in which the time-evolving nature of an airport network is considered. Specifically, based on a sequence of graph snapshots, a temporal convolutional block in accordance with the Markov property is employed to mine the time-varying patterns of flight delays. Moreover, an adaptive graph convolutional block is embedded into the proposed approach to explore spatial interactions hidden in airport networks. Using operational data obtained from CAAC, the proposed model is compared with several benchmark approaches in a study case involving 74 civil airports in China in 2018. The results show that our model achieves the best performance with all the evaluation metrics both in the short-term and the long-term predictions. Additionally, by considering interactions among airports, the deep learning approaches generally outperform the classical methods in which the interactions among airports are ignored.

The proposed model could be extended further if relevant data is available. For example, a more comprehensive scenario can be modeled by considering the overall landside and airside operation, such as boarding/de-boarding, security checks, ground holding, and taxiing. Moreover, considering the fact that most of the interactions and operations in air transportation are subject to highly regulated terms and conditions, it would be interesting to integrate data-driven methods with the operating rules of air traffic management.

ACKNOWLEDGMENT

This work is supported by the Funds of the National Natural Science Foundation of China (Grant Nos. 61822102, U2033215, U1833125).

REFERENCES

[1] International Air Transport Association (IATA), “Annual review 2019,” *https://www.iata.org/contentassets/*, Accessed February, 2020.

[2] EUROCONTROL, “Annual network operations report 2019,” *https://www.eurocontrol.int/publication/annual-network-operations-report-2019*, Accessed April, 2020.

[3] United States Department of Transportation, “Bureau of transportation statistics, airline service quality performance,” *https://transtats.bts.gov/*, Accessed October, 2020.

[4] Civil Aviation Administration of China, “Bulletin on the development of the civil aviation industry in 2019,” *http://www.caac.gov.cn/*, Accessed June 5, 2020.

[5] International Air Transport Association (IATA), “Iata economics,” *https://www.iata.org/en/publications/economics/*, Accessed March, 2020.

[6] J. J. Rebollo and H. Balakrishnan, “Characterization and prediction of air traffic delays,” *Transportation Research Part C: Emerging Technologies*, vol. 44, pp. 231–241, 2014.

[7] Y. Tu, M. O. Ball, and W. S. Jank, “Estimating flight departure delay distributions—a statistical approach with long-term trend and short-term pattern,” *Journal of the American Statistical Association*, vol. 103, no. 481, pp. 112–125, 2008.

[8] Y. Zhang and N. Nayak, “Macroscopic tool for measuring delay performance in national airspace system,” *Transportation Research Record*, vol. 2177, no. 1, pp. 88–97, 2010.

[9] R. D. Windhorst, “Towards a fast-time simulation analysis of benefits of the spot and runway departure advisor,” *AIAA Guidance, Navigation, and Control Conference*, 2012.

[10] B. Yu, H. Yin, and Z. Zhu, “Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting,” pp. 3634–3640, 7 2018.

[11] Y. J. Kim, S. Choi, S. Briceno, and D. Mavris, “A deep learning approach to flight delay prediction,” *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*, pp. 1–6, 2016.

[12] Y. Guleria, Q. Cai, S. Alam, and L. Li, “A multi-agent approach for reactionary delay prediction of flights,” *IEEE Access*, vol. 7, pp. 181 565–181 579, 2019.

[13] G. Gui, F. Liu, J. Sun, J. Yang, Z. Zhou, and D. Zhao, "Flight delay prediction based on aviation big data and machine learning," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 1, pp. 140–150, 2020.

[14] L. Hao, M. Hansen, Y. Zhang, and J. Post, "New york, new york: Two ways of estimating the delay impact of new york airports," *Transportation Research Part E: Logistics and Transportation Review*, vol. 70, pp. 245 – 260, 2014.

[15] J. Chen and M. Li, "Chained predictions of flight delay using machine learning," *AIAA Science and Technology Forum and Exposition*, 01 2019.

[16] B. Yu, Z. Guo, S. Asian, H. Wang, and G. Chen, "Flight delay prediction for commercial air transport: A deep learning approach," *Transportation Research Part E: Logistics and Transportation Review*, vol. 125, pp. 203–221, 2019.

[17] W.-B. Du, M.-Y. Zhang, Y. Zhang, X.-B. Cao, and J. Zhang, "Delay causality network in air transport systems," *Transportation Research Part E: Logistics and Transportation Review*, vol. 118, pp. 466 – 476, 2018.

[18] K. Cai, J. Zhang, M. Xiao, K. Tang, and W. Du, "Simultaneous optimization of airspace congestion and flight delay in air traffic network flow management," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 11, pp. 3072–3082, 2017.

[19] C. Yang, J. Mao, X. Qian, and P. Wei, "Designing robust air transportation networks via minimizing total effective resistance," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 6, pp. 2353–2366, 2019.

[20] C. Ciruelos, A. Arranz, I. Etxebarria, S. Peces, B. Campanelli, P. Fleurquin, V. Eguíluz, and J. J. Ramasco, "Modelling delay propagation trees for scheduled flights," *the 11th USA/Europe Air Traffic Management Research and Development Seminar*, 06 2015.

[21] Y. Xiao, Y. Zhao, G. Wu, and Y. Jing, "Study on delay propagation relations among airports based on transfer entropy," *IEEE Access*, vol. 8, pp. 97 103–97 113, 2020.

[22] C. Chen, C. Li, J. Chen, and C. Wang, "Vfdp: Visual analysis of flight delay and propagation on a geographical map," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–12, 2020.

[23] N. Pyrgiotis, K. M. Malone, and A. Odoni, "Modelling delay propagation within an airport network," *Transportation Research Part C: Emerging Technologies*, vol. 27, pp. 60 – 75, 2013.

[24] C.-L. Wu and K. Law, "Modelling the delay propagation effects of multiple resource connections in an airline network using a bayesian network model," *Transportation Research Part E: Logistics and Transportation Review*, vol. 122, pp. 62 – 77, 2019.

[25] G. Mehr and A. Eskandarian, "Estimating the probability that a vehicle reaches a near-term goal state using multiple lane changes," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–12, 2021.

[26] A. Eskandarian, C. Wu, and C. Sun, "Research advances and challenges of autonomous and connected ground vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 2, pp. 683–711, 2021.

[27] H. Zhao, Y. Li, W. Hao, S. Peeta, and Y. Wang, "Evaluating the effects of switching period of communication topologies and delays on electric connected vehicles stream with car-following theory," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–11, 2020.

[28] Y. Kan, Y. Wang, D. Wang, J. Sun, C. Shao, and M. Papageorgiou, "A novel approach to estimating missing pairs of on/off ramp flows," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 2, pp. 1287–1305, 2021.

[29] X. Ma, Z. Tao, Y. Wang, H. Yu, and Y. Wang, "Long short-term memory neural network for traffic speed prediction using remote microwave sensor data," *Transportation Research Part C: Emerging Technologies*, vol. 54, pp. 187 – 197, 2015.

[30] X. Shi, H. Qi, Y. Shen, G. Wu, and B. Yin, "A spatial-temporal attention approach for traffic prediction," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–10, 2020.

[31] Z. Cui, K. Henrickson, R. Ke, and Y. Wang, "Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 11, pp. 4883–4894, 2020.

[32] K. Guo, Y. Hu, Z. Qian, H. Liu, K. Zhang, Y. Sun, J. Gao, and B. Yin, "Optimized graph convolution recurrent neural network for traffic prediction," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–12, 2020.

[33] C. Park, C. Lee, H. Bahng, T. won, K. Kim, S. Jin, S. Ko, and J. Choo, "Stgrat: A spatio-temporal graph attention network for traffic forecasting," *arXiv*, vol. abs/1911.13181, 2019.

[34] X. Yin, G. Wu, J. Wei, Y. Shen, H. Qi, and B. Yin, "A comprehensive survey on traffic prediction," *arXiv*, vol. abs/2004.08555, 2020.

[35] J. Ye, J. Zhao, K. Ye, and C. Xu, "How to build a graph-based deep learning architecture in traffic domain: A survey," *arXiv*, vol. abs/2005.11691, 2020.

[36] R. Yu, Y. Li, C. Shahabi, U. Demiryurek, and Y. Liu, "Deep learning: A generic approach for extreme condition traffic forecasting," *Proceedings of the 2017 SIAM International Conference on Data Mining*, pp. 777–785, 06 2017.

[37] Z. Pan, Y. Liang, W. Wang, Y. Yu, Y. Zheng, and J. Zhang, "Urban traffic prediction from spatio-temporal data using deep meta learning," *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data*, p. 1720–1730, 2019.

[38] J. Zhang, X. Shi, J. Xie, H. Ma, I. King, and D. Yeung, "Gaan: Gated attention networks for learning on large and spatiotemporal graphs," *arXiv*, vol. abs/1803.07294, 2018.

[39] J. Li, Z. Han, H. Cheng, J. Su, P. Wang, J. Zhang, and L. Pan, "Predicting path failure in time-evolving graphs," *arXiv*, vol. abs/1905.03994, 2019.

[40] M. Schlichtkrull, T. Kipf, P. Bloem, R. V. Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks," *arXiv*, vol. abs/1703.06103, 2017.

[41] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Two-stream adaptive graph convolutional networks for skeleton-based action recognition," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12 018–12 027, 06 2019.

[42] T. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv*, vol. abs/1609.02907, 2017.

[43] V. Mnih, N. Heess, A. Graves, and K. Kavukcuoglu, "Recurrent models of visual attention," *arXiv*, vol. abs/1406.6247, 2014.

[44] M. S. Ahmed and A. R. Cook, "Analysis of freeway traffic time-series data by using box-jenkins techniques," *Transp. Res. Rec.*, vol. 722, pp. 1 – 9, 1979.

[45] J.-M. H. C.-H. Wu and D. T. Lee, "Travel-time prediction with support vector regression," *IEEE Transactions on Intelligent Transportation Systems*, vol. 5, pp. 276 – 281, 2004.

[46] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Graph convolutional recurrent neural network: Data-driven traffic forecasting," *arXiv*, vol. abs/1707.01926, 2017. [Online]. Available: http://arxiv.org/abs/1707.01926

[47] H.Scheffe, "The analysis of variance," *John Wiley & Sons*, vol. 72, 1999.

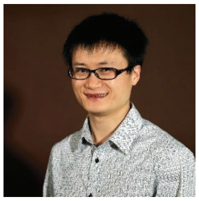
[48] Z. Hou and X. Li, "Repeatability and similarity of freeway traffic flow and long-term prediction under big data," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 6, pp. 1786–1796, 2016.

[49] J. Mackenzie, J. F. Roddick, and R. Zito, "An evaluation of htm and lstm for short-term arterial traffic flow prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 5, pp. 1847–1857, 2019.

[50] H. Zheng, F. Lin, X. Feng, and Y. Chen, "A hybrid deep learning model with attention-based conv-lstm networks for short-term traffic flow prediction," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–11, 2020.

Kaiquan Cai received the B.S. and Ph.D. degrees from Beihang University in 2004 and 2013, respectively. He is currently a professor with School of Electronic and Information Engineering, Beihang University, and deputy director of National Key Laboratory of CNS/ATM. His research interests include intelligent air navigation, networked collaborative air traffic management.

Yue Li received the B.S. degree in mathematics and applied mathematics from Xidian University, Xian, China, in 2017. He is currently pursuing the Ph.D. degree with the School of Electronic and Information Engineering, Beihang University, China. His research interests include deep learning and networked air traffic management systems.



Yi-Ping Fang received his Ph.D. degree in Industriel Engineering from École Centrale Paris (ECP), France. He is currently an assistant professor in the Chair Risk and Resilience of Complex Systems, Laboratoire Gene Industriel, CentraleSupélec, Université Paris-Saclay, France. He has been the Postdoc research fellow at ETH Zurich, Switzerland from March 2015 to January 2017. His primary research interests focus on the study and development of advanced computational methods for risk, reliability and resilience analytics of critical cyber-physical systems (e.g., power grids and electrified transportation systems). Particularly, He is interested in resilience assessment, stochastic and robust optimization, and machine learning.

Yanbo Zhu received the B.S. and Ph.D. degrees from Beihang University, in 1995 and 2009, respectively. He is currently vice president of Aviation Data Communication Corporation, China, and he is also a part-time tutors with School of Electronic and Information Engineering, Beihang University. His research interests include intelligent air navigation, aeronautical datalink communication, and collaborative air traffic management.