Phased Flight Trajectory Prediction with Deep Learning

KAI ZHANG, Lehigh University
BOWEN CHEN, Embry-Riddle Aeronautical University

The unprecedented increase of commercial airlines and private jets over the next ten years presents a challenge for air traffic control. Precise flight trajectory prediction is of great significance in air transportation management, which contributes to the decision-making for safe and orderly flights. Existing research and application mainly focus on the sequence generation based on historical trajectories, while the aircraft-aircraft interactions in crowded airspace especially the airspaces near busy airports have been largely ignored. On the other hand, there are distinct characteristics of aerodynamics for different flight phases, and the trajectory may be affected by various uncertainties such as weather and advisories from air traffic controllers. However, there is no literature fully considers all these issues. Therefore, we proposed a phased flight trajectory prediction framework. Multi-source and multi-modal datasets have been analyzed and mined using variants of recurrent neural network (RNN) mixture. To be specific, we first introduce spatio-temporal graphs into the low-altitude airway prediction problem, and the motion constraints of an aircraft are embedded to the inference process for reliable forecasting results. In the en-route phase, the dual-attention mechanism is employed to adaptively extract much more important features from overall datasets to learn the hidden patterns in dynamical environments. The experimental results demonstrate our proposed framework can outperform state-of-the-art methods for flight trajectory prediction for large passenger/transport airplanes.

CCS Concepts: • Information systems → Data mining; • Computing methodologies → Neural networks.

Additional Key Words and Phrases: Air traffic management, big data, flight trajectory prediction, aerial mobility

ACM Reference Format:
Kai Zhang and Bowen Chen. xx. Phased Flight Trajectory Prediction with Deep Learning. ACM Trans. Intell. Syst. Technol. 1, 1, Article 1 (January xx), 20 pages. https://doi.org/xxxx

1 INTRODUCTION

Worldwide, there is a dramatic increase in air traffic demand, which potentially results in more congested airspace and flight delay. To advance the aerial mobility and safety, some state agencies are developing air traffic management (ATM) modernization programs, for example, the Federal Aviation Administration in the United States developed Next Generation Air Transportation System (NextGen) to estimate sector capacity and enhance decision support. In Europe, EUROCONTROL adopted the PREDICT system to forecast the pre-tactical traffic load [34]. The core component of these systems is the aircraft trajectory prediction model that is being reshaped by transformative and disruptive new data-driven techniques especially machine learning in recent decades [45].

In general, an aircraft should obey regulations governing aspects of civil aviation operations and make an airway flight. Airway shown in Figure 1(a) is a defined corridor consisting of waypoints that connects one specified location to another at a specified altitude, and an aircraft should fly alone the centerline within a specific altitude block and corridor width ideally. Therefore, most of today’s practical trajectory prediction systems and
Zhang and Chen.

Fig. 1. The schematic diagram of air navigation. (a) Area navigation routes follow defined way-points (source: FAA). (b) Actual flight trajectories vs flight plan [34]. Blue lines refer to all trajectories filed the same flight plan shown by the red curve, which reveals that the actual flight trajectories may greatly deviate from their pre-departure plan.

literature [6, 31–33, 45, 46] estimate the flight trajectory in a deterministic way. To be specific, the trajectory is determined based on filed flight plan and historical routes. However, in some cases, the flight status should be adjusted appropriately according to surrounding weather, traffic conditions, and geometry environment [45] as shown in Figure 1(b). To address the problem of sensitivity of aircraft to external factors in flight trajectory forecasting, some researchers introduce weather information in modeling [4, 34, 39]. For example, [34] describes a feature cube referencing system that combines each track point of aircraft and weather conditions like wind speeds and air temperature in front of it to construct a high-dimension feature cube. This feature cube can be treated as a multi-channel image to be abstracted by the Convolutional Neural Network (CNN) and be embedded in to the training process. Although the introduction of weather information makes sense, researches usually employ a single model to estimate the whole trajectory, which conflicts with unique characteristics of flight stages because each flight phase has a definite flight mode and flight requirements [53]. Therefore, [45] divides flight trajectories into three phases – climbing, cruising, and approaching phases using clustering algorithms and proposed a Constrained Long Short-Term Memory (LSTM) network to predict future trajectories in different phases separately in terms of the input sequences. On the contrary, [6] adopted a ruled-based method to separate landing and takeoff phases.

Moreover, pilots have the innate ability to “read” one another. Certain low-altitude airspace is possibly occupied by various private jets, fixed-wing drones and commercial airplanes that take off or land, when they operate the aircraft in this crowded airspace near airports, they obey common sense rules and comply with safety conventions to avoid collisions and it could affect the trajectory. Figure 2 is an illustration of the air traffic volumes over a period of time in a day in terms of the number of flights with Orlando International Airport (MCO) – one of the busiest airports, as origin and destination. We can observe that approximately one airplane take-off or land in a minute, which intuitively presents the congestion of the airspace and indicates the necessity of aircraft-aircraft interaction modeling for flight trajectory prediction in crowded airspace near the airport.

Previous work barely considers all of above uncertainties comprehensively, hence, in the paper, we propose a phased flight trajectory prediction framework with hybrid RNNs considering dynamic spatio-temporal attributes in real air routes to bridge this gap. Hence in the paper, we present a novel architecture for short to mid-range trajectory prediction in different flight phases, namely takeoff, en-route, and approach. To capture the high-level dynamics of aircraft motion in takeoff/approach phases, we use the spatio-temporal graph (st-graph) to build a set of factor components. Each factor can be represented by a RNN that reflects the structure and interactions...
Fig. 2. The airline statistics of MCO from 4 pm to 6 pm on December 28, 2019. The orange bar represents the arrival flight while the blue represents the departure flight. (Data source: Airline On-Time Performance Data from United States Department of Transportation [36].)

of the st-graph. Motion constraints are employed in the trajectory inference process. For en-route trajectory prediction, we take the overall weather features along the planned airway as multiple "channels" to the training process with feature attention and temporal attention, which makes our model adaptable to uncertainty.

To the best of our knowledge, this is the first work proposed to systematically study phased flight trajectory prediction problem, and our work contributes to its advancements in the following ways:

- We firstly integrate the concept of aircraft-aircraft interaction into flight trajectory prediction framework. The aircraft in crowded airspace are represented by st-graphs, then the use of our developed attention-based structured-LSTM architecture well captures the structure and interactions of st-graph that yields more accurate results for trajectory prediction especially in crowded airspace.
- We design a reasonable and effective constraint trigger mechanism that introduces physical aerodynamics limitations in takeoff and approach phases for a more reliable inference process.
- We conduct extensive experiments to evaluate our proposed framework on flight trajectory prediction for all phases using our fused dataset. The experimental results demonstrate the performance of our method against some state-of-the-art approaches on our collected and incorporated aviation data.

The remainder of this paper is structured as follows. Section 2 highlights the related work. Section 3 formulates the flight trajectory prediction problem and summarizes our work as well as the used datasets. Then the detailed LSTM-based models for takeoff/approach and en-route phases are presented in Section 4 and Section 5, while Section 6 evaluates our proposed phased flight trajectory prediction framework using multiple control settings and analyzes the experiment results. Finally, we conclude this paper in Section 7.

2 RELATED WORK

In this section, we first introduce related studies on mining aviation data, followed by general methods on prediction modeling and trajectory cluster analysis. At last, we summarize past literature in the domain of modeling human-human spatio-temporal interactions.

2.1 Aviation Data Mining

The aviation industry and related fields such as air traffic management systems and weather forecasting generate large-scale complex data, requiring specialized analysis and modeling to extract useful knowledge and insights. In
the field of aviation big data analytics and mining, most of researchers focused on operation problems [14, 15, 25] and revenue management [3, 55]. In general, the forecasting and analysis of flight delay, flight flow and flight trajectory is the mainstream research for advanced air traffic management.

ADS-B system plays an important role in the safe navigation and air traffic control management, and its messages can be received by everyone when airplanes are flying over. Therefore, ADS-B data has been widely used in the field of aviation data mining. However, considering the lack of security mechanisms in the ADS-B system, researchers applied multiple algorithms to detect anomalies and intrusions of ADS-B messages [20, 30, 49] based on flight routes or sensors’ clocks.

2.2 Trajectory Forecasting

Trajectory forecasting is typically taken as a sequence prediction problem whether it is pedestrian trajectory, vehicle trajectory or flight trajectory. The methods applied in trajectory forecasting are summarized as parametric and non-parametric models. Parametric models are built on prior known parameters including Origin-Destination matrix [54], Kalman filter [24, 41], and Markov model [13, 42], while non-parametric models contain support-vector machine [56] and neural networks [35].

Recently, RNN and its variants such as LSTM [22] and Gated Recurrent Units [10] have been proven to be effective for sequence prediction tasks such as machine translation and speech recognition. Therefore, they are also widely employed to predict trajectory and obtain the state-of-the-art results, for example, [9] used a CNN-LSTM hybrid network to predict the near-term trajectories of agents in dense traffic video. [40] proposed a RNN approach for aircraft trajectory prediction with weather features, which aims to calibrate the flight plan.

2.3 Trajectory Clustering

Aircraft trajectory clustering aims to identify patterns of air traffic by grouping similar trajectories, which in turn contributes to the airspace planning, air traffic flow management, and flight time estimation [18]. K-means and DBSCAN are commonly used clustering algorithms for point-based data including aircraft waypoints. Some clustering methods exist in the literature for flow identification, where the objective is to determine the set of clusters that best fit the operational air traffic flows within a airspace [37]. For example, [7] proposed a en-route flow analysis framework based on HDBSCAN algorithm [8], an extended version of DBSCAN, which extracts a simplified hierarchy consisted only of the most significant clusters from the generated complete density-based clustering hierarchy. [38] focused on clustering flows around airport with DBSCAN and conducted risk assessment in air traffic safety. As for aircraft ground handling, [11] took DBSCAN as a core in its algorithmic approach to cluster aircraft taxiing trajectories.

2.4 Human-human Spatio-Temporal Interactions

A pioneering work, namely Social Force model [21], used attractive and repulsive forces in terms of relative distances to capture the interactions between agents and achieves good performance. This idea was later followed by researchers in the field of robotics and [50] pointed out that Social Force can not model the complex crowd behavior like cooperation, then a Gaussian process-based way was proposed.

Unlike previous work using hand-crafted proximity relationships, [2] introduced a social pooling layer which allows agents share their spatially proximal information with each other. Besides, the interactions that could occur in a more distant future are expected using LSTM. Inspired by this work, [29] took into account scene context, and employed a variational autoencoder to obtain a diverse set of hypothetical trajectory samples, which are then ranked by a scoring module based on RNN. [19] presented Social GAN by combining recurrent sequence-to-sequence model and generative adversarial networks (GAN). [44] introduced an GAN-based approach with a
Fig. 3. Data visualization. (a) Airspace condition near Daytona Beach International Airport from the ERAU Live Traffic application developed by ERAU’s NextGen Applied Research Laboratory (NEAR Lab) [16]. (b) Vertical profile of relative humidity, red showing high relative humidity and blue showing drier air. (Data source: the rapid refresh numerical weather model [27].)

physical attention and a social attention that incorporate the influence of all agents in the scene as well as the scene context.

To represent high-level spatio-temporal structures of human motions, [23] used spatio-temporal graph [5, 26] and transformed it into a feedforward mixture of RNNs that can be jointly trained to model dynamics in spatio-temporal tasks. In this paper, we use a similar structure but with attention and constraint modules.

3 PRELIMINARIES

In this section, we present the problem statement for traffic trajectory prediction. The detailed information of datasets and data preprocessing are described next, and the overall structure of our proposed framework is illustrated in Figure 4.

3.1 Problem Statement

In this paper, we deal with the problem of flight trajectory prediction in three phases – climbing, cruising, and approaching phases. In general, the problem of flight trajectory prediction can be viewed as a sequence generation task. Let \((x_i^t, y_i^t, z_i^t)\) represents the (latitude-longitude-altitude)-coordinate of aircraft \(i\) at time-instant \(t\) and let \(F_i^t\) represents other features such as weather and aircraft types, our problem can be formulated as:

Given spatial locations \(\{(x_i^t, y_i^t, z_i^t)\}\) and corresponding features \(F_i^t\) for aircraft \(i\) from time 1 to \(T_{obs}\), predict their future positions for time instants \(T_{obs+1}\) to \(T_{pred}\). Note that we focus on short to short-term trajectory prediction, the forecast period is typically less than 10 minutes.

3.2 Data Sources and Preprocessing

In this paper, we use three types of data – trajectory data, weather data, and flight profile. The Apache NiFi is utilized to parse and fuse data in a parallel manner. Figure 3 is the graphical representation of some of our data. We will describe their attributes and the corresponding techniques of preprocessing in the following.

3.2.1 Trajectory data: ADS-B data is the periodically broadcasted information by the aircraft, ground stations and other aircraft flying in the near airspace to avoid collision. The ADS-B data contains all flights installed with ADS-B equipment which covers most commercial aviation as well as partial general aviation [57]. ADS-B data used in this paper comes from ADS-B receivers of NEAR Lab at the Daytona Beach campus. In other words, the

ACM Trans. Intell. Syst. Technol., Vol. 1, No. 1, Article 1. Publication date: January xx.
maximal spatial coverage in the data can only cover the airspace over Florida and some parts of neighboring states. The original data collected by NEAR are ASTERIX Category 033 messages that contain 17 attributes such as latitude/longitude, velocity, aircraft ID and flight number. There are lots of trajectories of trainer aircraft in our data because of many flight schools in Florida. Such trajectories are always unpredictable since student pilots do different actions randomly. Therefore, we filter these trajectories in terms of aircraft ID and aircraft type but keep other private jet’s trajectories because they are much more deterministic.

3.2.2 Weather data: The weather data used in this paper is High-Resolution Rapid Refresh (HRRR) which is the newest version of the Rapid Refresh (RAP) numerical weather model running by the National Centers for Environmental Prediction (NCEP). HRRR generates data down to a 3-km resolution grid every hour so users can analyze the weather condition for quite small regions of interest. The data is composed of various attributes at different height level.

3.2.3 Flight profile: Flight profile consists of aircraft profile such as aircraft’s type and flight plan. Flight plans indicate the planned air route prior to departure. This information is extracted from the records of Traffic Flow Management System (TFMS) system which is also provided by NEAR lab. Original flight plan data is the set of variable-length strings or fixed way-point IDs, so we convert them as pairs of latitude and longitude using the official lookup table [12]. We employ the approximate trajectory partitioning algorithm [28] to identify characteristic points in flight plans with varying lengths. The final flight plan dataset includes 36 unique flight plans with a fixed length of 10 including the first and last waypoints. It is worth noting that almost all flight plans of general aviation do not exist in the database, we set them as all 0 by default.

3.2.4 Phase Identification: There are unique characteristics in each flight phase, however, there is no such a feature in our aerial datasets. Previous work [45] applied clustering algorithms two times to create sub-clusters based on the characteristics of trajectory sequence, but the consistency of generated sub-clusters can not be guaranteed because each entry in the trajectory dataset is relatively close to its neighbors [47]. To solve this issue, [48] proposed to extract continuous flights using DBSCAN and then apply fuzzy logic to segment flight trajectory. This method is employed in our work because it can handle large-scale data efficiently. To be specific, we use the following relationships to identify the correct flight phase given altitude $H$, speed $V$, vertical rate $RoC$:

$$\begin{align}
\text{if } H_{lo} \land V_{mid} \land RoC_+ \text{ then Takeoff} \\
\text{if } H_{hi} \land V_{hi} \land RoC_0 \text{ then En-Route} \\
\text{if } H_{lo} \land V_{mid} \land (RoC_- \lor RoC_0) \text{ then Approach}
\end{align}$$

where for a commercial flight, $H_{lo}(\eta) = G(\eta, 10000, 10000)$, $H_{hi} = G(\eta, 35000, 20000)$, $V_{mid}(v) = G(v, 300, 100)$, $V_{hi}(v) = G(v, 600, 100)$, $RoC_0(\tau) = G(\tau, 0, 100)$, $RoC_+(\tau) = S(\tau, 10, 1000)$, $RoC_-(\tau) = Z(\tau, -1000, -10)$. $G(x; \mu, \sigma)$, $S(x; a, b)$, $Z(x; a, b)$ are Gaussian function, S-shaped membership functions, Z-shaped membership functions, respectively. Fuzzy logic takes such relations between inputs and output to identify the three different flight phase (takeoff, en-route, approach) given data point $(\eta_i, \tau_i, v_i)$. For more details on phase identification, we refer the reader to [48].

3.3 Framework Architecture

Figure 4 shows the architecture of our approach, which is comprised of two components. First, in the takeoff or approach phase, the aircraft-aircraft interactions are considered to build a spatio-temporal graph where each node represents an aircraft, edges capture the interactions between them. Then, a variant of structural RNN combined with the motion constraints is proposed. We transform the st-graph as a factor graph and represent each factor with a LSTM. The relative importance of each aircraft is captured using a soft-attention module. The second stage is to predict en-route flight trajectory. It is not reasonable to take only local (current position of an
aircraft) weather information to predict en-route flight because the air route may change when weather along the flight plan changes. Hence, we leverage CNN to extract compact representation of global (the southeastern region of the United States in this paper) weather information. To appropriately select the most relevant features and capture long-term temporal dependencies of a sequence, the dual-stage attention [43] is used. As the last step, the long-term trajectory can be predicted in a multi-step way.

4 MODELING AIRCRAFT-AIRCRAFT INTERACTIONS IN CROWDED AIRSPACES

Predicting the motion of aircraft targets while taking into account the complex and often subtle interactions that occur between aircraft in crowded airspaces, is an extremely challenging problem. This problem is inherently comprised of factors in space and time. To capture the dynamics in such a spatio-temporal task, we use factor graph of st-graph representation in which the nodes represent the problem components, and the edges capture their spatio-temporal interactions [23].

4.1 Spatio-Temporal Graph Representation

Let $G = (\mathcal{V}, \mathcal{E}_S, \mathcal{E}_T)$ denote an st-graph, where $\mathcal{V}$ is the node set, $\mathcal{E}_S$ and $\mathcal{E}_T$ represent as the set of spatial edges and temporal edges, respectively. The graph structure $(\mathcal{V}, \mathcal{E}_S)$ is unrolled over time through $\mathcal{E}_T$ to form $G$. In the unrolled st-graph, different nodes $v \in \mathcal{V}$ at the same time instant are connected by spatial edges $e = (u, v) \in \mathcal{E}_S$, whereas same nodes at adjacent time instants are connected by temporal edges $e = (u, u) \in \mathcal{E}_T$.

In our work, we formulate the problem of flight trajectory prediction in takeoff and descending phases as an st-graph. Hence, the nodes of st-graph represent the aircraft, the spatial edges link two different aircraft at the same time instant, and temporal edges link the identical aircraft at neighbouring time instants. Note that the st-graph is constructed using nodes within a certain spatial scene. The dynamics of relative orientation and distance between aircraft is extracted using spatial edges, while the dynamics of the aircraft’s own trajectory is extracted using temporal edges. On the basis of the above definitions, we let $x^t_v$ and $x^t_e$ denote the feature vectors associated with nodes and edges, respectively as shown in Figure 5, then our goal is to predict the node labels or position vectors in the problem formulation at each time instant $t$. For example, in aircraft-aircraft interaction, the node features can represent their current positions with aircraft attributes and corresponding weather information, and edge features represent their relative orientation; the node labels represent the aircraft future motion that is affected by both its node and edges, leading to a complex system. Such interactions are
commonly parameterized with a factor graph that transports how a complex function over st-graph factorizes into a set of simpler functions including a factor function for each node and a pairwise factor function for each edge and the parameters of these factors are needed to be learned [52]. In our formulation, we let all the spatial edges share the identical factor function while temporal edges share the other one common factor to reduce the computational complexity and improve the model scalability with more nodes without increasing the number of parameters.

4.2 Attention-based Structural LSTM Based on Spatio-Temporal Graphs

After the st-graph representation is decomposed into a set of factor components as discussed in Section 4.1, we represent each factor using a LSTM similar to [23] but a soft attention mechanism is introduced to compute the attention score over hidden states of adjacent spatial edgeLSTMs for each node, which is summarized in Figure 6. In detail, we have edgeLSTMs $R_\psi$ (the LSTMs related to edge factor $\psi_V$) to model the dynamics of aircraft-aircraft interations and individual motions in crowded airspace. Then nodeLSTMs $R_\psi$ (the LSTMs related to node factor $\psi_V$) can predict the aircraft’s future position at the next time step using the node features and hidden states of the adjacent edgeLSTMs. Besides, the model parameters are shared across all nodes and edges, therefore, the number of parameters is independent of the number of aircraft at any given time.

4.2.1 EdgeLSTM: For each spatial edgeLSTM $R_{uv}$ at each time instant $t$, we embed the corresponding edge’s feature vector $x_{uv}^t$ into a fixed-length vector $e_{uv}^t$ that is the input to the LSTM cell in the following:

$$e_{uv}^t = \phi(x_{uv}^t; W_{Spatial}^e)$$

(2)

$$h_{uv}^t = \text{LSTM}(h_{uv}^{t-1}; e_{uv}^t; W_{Spatial}^r)$$

(3)

where $\phi(\cdot)$ is an embedding function, $W_{Spatial}^e$ is the embedding weight. The edgeLSTM weights are denoted by $W_{Spatial}^r$ and the hidden state is denoted by $h_{uv}$. Similarly, the LSTM cell for temporal edgeLSTM $R_{uu}$ is built as:

$$e_{uu}^t = \phi(x_{uu}^t; W_{Temporal}^e)$$

(4)

$$h_{uu}^t = \text{LSTM}(h_{uu}^{t-1}; e_{uu}^t; W_{Temporal}^r)$$

(5)

these weights are the trainable parameters for edgeLSTMs.

4.2.2 Soft Attention Module: We compute a scaled dot-product attention [51] given in Equation (6) over the hidden states of edgeLSTMs of all spatial edges $h_{uv}^t, i \in [1, m]$ which each node associates with.

$$\text{Attention}(h_{uv}^t, h_{v_i}^t) = \sigma\left(\frac{m}{\sqrt{d_e}} (W_{uv}h_{uv}^t, W_{v_i}h_{v_i}^t)\right)h_{v_i}^t$$

(6)
Fig. 6. Architecture of Attention-based Structural LSTM that is composed of EdgeLSTM (left), Soft Attention Module (middle), and NodeLSTM (right).

where $m$ is the number of spatial edges the node connects, $W_{eo}$ and $W_e$ are weights that are used to project the hidden states into vectors with a fixed-dimension $d_e$. $\sigma$ is the softmax function. Note that $m$ depends on the number of aircraft and may change in every time frame.

Therefore, we can obtain the output vector of the attention module, a weighted sum of hidden states at each time step $t$,

$$H_v^t = \sum_{i=1}^{m} \text{Attention}(h_{eo}^t, h_v^t)$$ (7)

4.2.3 NodeLSTM: The aim of nodeLSTMs is the position estimation, to be specific, is to predict the distribution of the flight trajectory $(\hat{x}, \hat{y}, \hat{z})$ at the next time step based on concatenated information at the previous time frame. Inspired by the previous work [2, 17], we assume a multivariate Gaussian distribution and we denote $(\mu_v^t, \sigma_v^t, \rho_v^t)$ as the mean, standard deviation, and correlation coefficient of it. These parameters are predicted by a linear layer $W_p$ and the predicted coordinates $(\hat{x}, \hat{y}, \hat{z})_v^t$ are given by

$$(\hat{x}, \hat{y}, \hat{z})_v^t \sim N(\mu_v^t, \sigma_v^t, \rho_v^t)$$ (8)

$$\mu_v^t = (\mu_x^t, \mu_y^t, \mu_z^t), \sigma_v^t = (\sigma_x, \sigma_y, \sigma_z), \rho_v^t = (\rho_x, \rho_y, \rho_z)$$ (9)

$$[\mu_v^t, \sigma_v^t, \rho_v^t] = W_p h_v^t$$ (10)

The input to nodeLSTM consists of the embedded node’s features $x_v^t$, the hidden state of the corresponding temporal edgeLSTM $h_{eo}^t$, and the attention output $H_v^t$ as follows:

$$e_v^t = \phi(x_v^t; W_{node}^e)$$ (11)

$$a_v^t = \phi(\text{concat}(h_{eo}^t, H_v^t); W_{attention}^e)$$ (12)

$$h_v^t = \text{LSTM}(h_{eo}^{t-1}, \text{concat}(e_v^t, a_v^t); W_{node}^e)$$ (13)

The trainable parameters of the LSTM model are jointly learned by minimizing the negative log-likelihood loss ($L_v$ for the trajectory of node $v$):

$$L_v(W_{Spatial}^e, W_{Temporal}^e, W_{node}^e, W_{attention}^e, W_{node}^r, W_p) = - \sum_{t=1}^{T_{pred}} \log(P(x_v^t, y_v^t, z_v^t | \mu_v^t, \sigma_v^t, \rho_v^t))$$ (14)
4.3 Motion Constraints and Model Inference
During test time, the trained model is fitted to sample the predicted multivariate Gaussian distribution to obtain the future position \( \{ (\hat{x}_t', \hat{y}_t', \hat{z}_t') \} \) using the observation at previous time steps from \( t \) to \( T_{obs} \). From time step \( T_{obs+1} \) to \( T_{pred} \), the predicted positions \( \{ (\hat{x}_t', \hat{y}_t', \hat{z}_t') \} \) are used to compute node features and spatial edge features in place of actual positions for each time.

To mitigate the generalization error especially the effect of noise and variance of the model, we propose knowledge-guided motion constraints for takeoff and approach phases independently and incorporate them into the inference process as shown in Algorithm 1. The sets of parameters \( \alpha \) and \( \beta \) in the formulations of motion constraints are achieved and explained in Section 4.3.1 and 4.3.2. In addition, the phase identification has been discussed in Section 3.3. Note that some information such as heading may be missing in the trajectory data so that we cannot calculate the constraint parameters. Therefore, the motion constraints module will not be applied in such a case.

**Algorithm 1 Flight Trajectory Inference Based on Constraints**

**Input:** Trajectory \( \{ (x_t, y_t, z_t) \} \) with associated features \( F_t, t = 1, ..., T_{obs} \), a set of constraints \( \alpha, \beta \) for takeoff and approach phases, respectively.

**Output:** Future trajectory \( \{ (x_t, y_t, z_t) \} \), \( t = T_{obs+1}, ..., T_{pred} \)

1: \( \mu'_o, \sigma'_o, \rho'_o \leftarrow \) fit the trained LSTM model with test input \( (x_t, y_t, z_t, F_t), t \in [1, T_{obs}] \)

2: \( \text{phase} \leftarrow \) phase indicator function with test input

3: \( \text{LOOP Process} \)

4: \( \text{for} \ t = T_{obs+1} \text{ to } T_{pred} \text{ do} \)

5: \( \text{while phase = takeoff or approach do} \)

6: \( \text{if} \ (x_t, y_t, z_t) \text{ satisfies limitation } \alpha \text{ or } \beta \text{ then} \)

7: \( \text{Append} \ (x_t, y_t, z_t) \text{ to the estimated future trajectory sequence } \{ (x, y, z) \} \)

8: \( t = t + 1 \)

9: \( \text{break} \)

10: \( \text{else} \)

11: \( \text{continue} \)

12: \( \text{return} \ \text{estimated trajectory } \{ (x, y, z) \} \)

4.3.1 Climbing and Descending Constraints. Figure 7 depicts a schematic diagram of climbing and descending phases. The constraints are considered in the paper to apply the angle limitation. We firstly extract all trajectories that are identified as in the takeoff phase from the entire data, and set the points with maximum altitudes as their Top of Climb (TOC) points \( \{ (x_{toc}, y_{toc}, z_{toc}) \} \) to calculate the climb angle \( \{ \theta'_c \} \) based on the first way-points \( \{ (x_o, y_o, z_o) \} \) above 1500 feet of the trajectory. Then the maximum climb angle \( \theta_c = \max \{ \theta'_c \} \) can be known using Equation (15). Similarly, we can also obtain the Top of Descend (TOD) points \( \{ (x_{tod}, y_{tod}, z_{tod}) \} \) and the maximum descend angle \( \theta_d \).

\[
\begin{align*}
\theta \leq \theta_c &= \max \{ \theta'_c = \tan^{-1}(\Delta h/\Delta d) \}, \\
\Delta h &= |z_{top} - z_o|, \\
\Delta d &= 2r \sin^{-1}(\sqrt{\sin^2 (\Delta y/2) + \cos (\gamma') \sin^2 (\Delta x/2)}), \\
\Delta x &= |x_{top} - x|, \Delta y = |y_{top} - y|, \\
\cos (\gamma') &= \cos (y'), \cos (\gamma') = \cos (y) 
\end{align*}
\]
where $\Delta d$ is the great-circle distance between two way-points on a sphere with latitude and longitude $(x, y)$ and $r$ is the radius of the Earth.

### 4.3.2 Rate of Turn Constraint.

The rate of turn (ROT) is the number of degrees (expressed in degrees per second) of heading change that an aircraft makes, the ROT can be determined by taking the constant of 1,091, multiplying it by the tangent of any bank angle $\theta_b$ and dividing that product by a given airspeed $v_a$ in knots [1]. Suppose the maximum ROT is $\omega_{\text{ROT}}$, then the ROT constraint can be expressed as Equation (16) that also describes the relationship between bank angle and airspeed as they affect the ROT.

$$
\omega \leq \omega_{\text{ROT}} = \max\{\omega_{\text{ROT}}^*\}, \\
\omega_{\text{ROT}}^* = \left(\frac{1091 \times \tan \theta_b}{v_a}\right)
$$

(16)

where $\{\omega_{\text{ROT}}^*\}$ can be obtained from our data in the same way as described in Section 4.3.1.

## 5 DUAL-STAGE ATTENTION-BASED LSTM FOR EN-ROUTE TRAJECTORY PREDICTION

En-route is a flight phase between climbing and descending which usually consumes the majority of a flight. Typically, aircraft in the en-route phases fly at a constant velocity and altitude but their heading or direction of flight may change. Due to the vast airspace and flight level allocation by ATC in the en-route phase, an airplane has relatively little impact on others. Intuitively, it seems much easier to predict flight trajectory in en-route than other phases. However, because of the requirement of global observations instead of local observations of weather uncertainty, it is a big challenge to predict the abnormal but actual trajectory that deviates from original flight plan. Hence, the dual-stage attention mechanism [43], in which the relevant input features and relevant hidden states are adaptively and sequentially extracted, can be deployed to our en-route trajectory prediction model. For example, each feature vector in the first attention layer will be weighted so that more information of more important features (higher weights) can be introduced into the decoder. Then, in the second stage, important time steps such as the one that exists a sudden thunderstorm are captured by the temporal attention layer.

### 5.1 Weather Feature Representation

The weather information in the Rapid Refresh model is represented by a set of reference 3D points including latitude, longitude, and altitude. The latitude and longitude of reference points are fixed. Therefore, we build the boundary in the latitude-longitude plane using the reference points and the middle points between each neighboring reference point, and construct a series of cubes in terms of altitude. A cube contains the weather information of a 3D airspace such as humidity, wind speed and direction. We assume that an aircraft flies at a fixed altitude in the en-route phase, in other words, this aircraft passes through cubes on a horizontal plane. Therefore, to extract the high-dimensional global weather information, each feature can be thought as a channel and be filtered into a fixed-size compact representation using three 2-D convolutional layers with dense. There
Fig. 8. Overview of En-route trajectory prediction. At time-step $t$, input attention module (left) computes attention weights $\{\alpha_t\}$ for each input feature $\{x_1, x_2, ..., x_n\}$ according to the hidden state $h_e$ of the encoder at previous time-step $t-1$. The weighted input is then fed into the encoder. Temporal attention module (right) computes attention weights $\{\beta_t\}$ according to the hidden state $h_d^{t-1}$ of the decoder. The weighted sum of encoder hidden states over all time-steps $c_t$ is represented as the input to decoder that outputs predicted way point $\hat{wp}_T$.

are 7 channels or weather features in total, they are HGT (geopotential height), TMP (temperature), RH (relative humidity), VVEL (vertical velocity), UGRD (u-component of wind), VGRD (v-component of wind), and ABSV (absolute vorticity). The output at time step $t$ is denoted by:

$$C_t = \text{conv}(F^t_W; W_{cnn})$$

(17)

where $F^t_W$ represents weather information of all channels, $W_{cnn}$ represents trainable parameters of convolutional neural networks. Note that the dimension of weather feature representation should be equal to the input sequence.

5.2 Dual-stage Attention Mechanism

Figure 8 shows the feedward-pass of our proposed en-route trajectory prediction model based on dual-stage attention-based LSTM. The input attention and temporal attention blocks have the same score function as we used in Section 4.2.2.

Different from the structural LSTM for low-altitude airspace, normal en-route trajectory usually follows the flight plan without influence caused by other aviation, consequently, we need to focus more on the aerodynamics of flight. We thus take the velocity components in each direction $\vec{V}$ and the track angle $\Theta$ into account to design the Location-Velocity-Angle (LVA) loss function given by:

$$L(W_{cnn}, W_{rnn}, W_e) = \frac{1}{n} \sum_{i=1}^{n} [(\vec{V}_i - \hat{\vec{V}}_i)^2 + (\Theta_i - \hat{\Theta}_i)^2 + (T_i - \hat{T}_i)^2]$$

(18)

where $W_{rnn}$ refers to the parameters of LSTMs (in both encoder and decoder) to learn, $W_e$ is the weight of embedding layer that is used to get a fixed-dimension representation of the flight profile. $T_i - \hat{T}_i$ denote the distance between two trajectories, which can be calculated by the Equation (15).

6 EXPERIMENTS

In this section, we firstly describe the experimental settings, and then compare the performance of our proposed framework with other methods. Next, we will use two example to explain why our method can achieve good performance.
6.1 Evaluation Metrics

We use the aerial data within the southeastern region around Florida state of the United States in 2019 to assess the performance of our proposed framework. The data is segmented into three sub-datasets for different flight phases. For each sub-dataset, the training set and test set are partitioned according to specific dates which are sampled randomly. Specifically, the data of 219 days (%60 of 365) is taken as the training set, the data of 73 days (%20 of 365) is taken as the validation set, while the remaining data of other days is taken as the test set. We also use the same training and testing dataset for other baseline and state-of-the-art methods for performance comparison. Three commonly-used evaluation metrics are applied to report the prediction error:

- **Average displacement error (ADE):** the mean euclidean distance (in meter) over all predicted way-points and actual trajectory.
- **Final displacement error (FDE):** the distance (in meter) between final way-point of the estimated flight trajectory and the actual destination.
- **Mean absolute error (MAE):** in this paper, this metrics is mainly employed to measure the prediction error of a single parameter of geo-locations. The latitude and longitude values are decimal degrees and altitude is in meters.

6.2 Implementation Details

The dimension of hidden states of all LSTMs in our framework is set to 256, while the dimension of embedding layers in structural LSTM is set to 64 with ReLU activation and dimension of embedding layers in en-route depends on input sequence. The ADS-B system records data per 0.5 second so that the distance between two adjacent way-points is too close. Hence, we scale the sampling rate to let one time-step corresponds to 10 seconds. During test time, we observe a trajectory for 3 minutes ($T_{obs} = 18$) and predict is future trajectory for the next 5 minutes ($T_{pred} = 48$). We use a learning rate of 0.005 with Adam optimizer to train our framework on a single GPU with Pytorch.

6.3 Results and Analysis

Because of the limitation of our dataset, there are not too many complete trajectories (from origin to destination). Therefore, the results of our experiments mainly show the medium-term or short-term prediction error, but the longer trajectory forecasting can also be obtained by the mean of multi-step forecasting and the jointed outputs from each model. We compare the performance of our model with off-the-shelf methods as well as multiple control settings:

- Kalman Filter (KF) [2]: we extrapolate trajectories with assumption of linear acceleration in takeoff/approach phases and constant speed in en-route phase. KF is applied in the case where the input only contains waypoints (C1).
- Hidden Markov Model (HMM) [4]: HMM is applied in both cases where the input only contains waypoints (C1) and the input contains all information including flight plans and weather (C2).
- Our Vanilla LSTM (V-LSTM): this is simplified setting of our model where we remove the st-graph, motion constraints, and dual-attention modules and treat all the trajectories to be independent of each other. This model is applied in both cases (C1 and C2).
- Social LSTM (S-LSTM) [2]: it is a state-of-the-art method named Social LSTM (S-LSTM), which takes into account influence of interaction. This model is applied in both cases (C1 and C2).

6.3.1 Comparison between Models for Takeoff and Approach Phases. Table 1 and Table 2 present the prediction errors of climbing and descending flights for all methods in Case 1 during test time, respectively. The models under comparison includes baseline methods without considering the aircraft-aircraft interactions such as V-LSTM,
Table 1. Quantitative Results (C1) of All the Methods for Takeoff Phase

| Model  | ADE    | FDE    | MAE    |
|--------|--------|--------|--------|
|        | longitude | latitude | altitude |
| V-LSTM | 1842.9849 | 3465.8422 | 0.0188 | 0.0075 | 195.2549 |
| KF     | 2375.4481 | 4331.8186 | 0.0214 | 0.0093 | 231.7216 |
| HMM    | 1981.8992 | 2994.1124 | 0.0186 | 0.0074 | 182.4849 |
| S-LSTM | 688.6658  | 1114.6335 | 0.0056 | 0.0048 | 147.5498 |
| Our    | 357.4578  | 512.4812  | 0.0024 | 0.0022 | 68.2185  |

Table 2. Quantitative Results (C1) of All the Methods for Approach Phase

| Model  | ADE    | FDE    | MAE    |
|--------|--------|--------|--------|
|        | longitude | latitude | altitude |
| V-LSTM | 2155.1877 | 4311.8799 | 0.0199 | 0.0184 | 222.8441 |
| KF     | 4572.4678 | 6248.5177 | 0.0374 | 0.0266 | 219.5478 |
| HMM    | 2949.7379 | 4712.5413 | 0.0214 | 0.0193 | 194.5148 |
| S-LSTM | 1436.8421 | 3248.2154 | 0.0083 | 0.0057 | 243.5582 |
| Our    | 725.7452  | 2811.9688 | 0.0053 | 0.0034 | 151.7838 |

Table 3. Quantitative Results (C2) of All the Methods for Takeoff Phase

| Model  | ADE    | FDE    | MAE    |
|--------|--------|--------|--------|
|        | longitude | latitude | altitude |
| V-LSTM | 573.3794  | 1643.6625 | 0.0042 | 0.0035 | 116.2396 |
| HMM    | 612.4712  | 1385.1277 | 0.0044 | 0.0038 | 130.8003 |
| S-LSTM | 488.9801  | 914.8549  | 0.0036 | 0.0029 | 85.7734  |
| Our    | 341.8263  | 437.8141  | 0.0024 | 0.0022 | 60.7959  |

KF, and HMM. Besides, S-LSTM, which takes into account influence of interaction, is also used for comparison. In general, our model far exceeds others, especially in the approach phase where there are much more complex motion patterns because of the air traffic control. In Case 2, as shown in Table 3 and Table 4, we obtain the similar conclusion. Besides, we can demonstrate that more information leads to better predictive trajectories when we compare results of different phases in a same case (e.g. Table 1 vs. 3 and Table 2 vs. Table 4).

To further explain how the air traffic control and aircraft operations influence flight trajectory as well as what patterns corresponding to ATM learned by models, we then take a case where two airplanes (UPS1328 and FDX1074) approach MCO airport on August 1, 2019 as an example. Figure 9 shows this classical ATM case in a crowded airspace. The process is described in detail in its caption. Besides, we applied our model and vanilla LSTM to predict these two trajectories, separately. The result shows that our model learns how multiple agents cooperate but vanilla LSTM fails. Figure 9(a) highlights three Points of Interests (POIs) $I_1$, $I_2$ and $I_3$, in other words, the way-points that one aircraft yield the other one to ensure the flight safety. From Figure 9(b), we observe that the trajectory predicted by the Vanilla LSTM $\text{lst}m_2$ exceeds the final point or destination $D_1$ in this time.
Table 4. Quantitative Results (C2) of All the Methods for Approach Phase

| Model  | ADE   | FDE   | MAE (longitude) | MAE (latitude) | MAE (altitude) |
|--------|-------|-------|----------------|----------------|---------------|
| V-LSTM | 1048.5811 | 2183.5482 | 0.0085          | 0.0054          | 234.2214      |
| HMM    | 862.1238  | 1284.5943  | 0.0073          | 0.0046          | 193.4833      |
| S-LSTM | 801.6282  | 1422.5314  | 0.0062          | 0.0046          | 158.7524      |
| Our    | 450.9431  | 766.2355  | **0.0036**      | **0.0024**      | **92.8776**   |

Table 5. Quantitative Results (C1) of All the Methods for En-route Phase

| Model  | ADE   | FDE   | MAE (longitude) | MAE (latitude) | MAE (altitude) |
|--------|-------|-------|----------------|----------------|---------------|
| V-LSTM | 842.7844  | 1477.2425 | 0.0061          | 0.0031          | 131.5484      |
| KF     | 874.8100  | 1589.2196  | 0.0065          | 0.0035          | 95.4739       |
| HMM    | **825.1741**  | 1354.3221  | **0.0058**      | **0.0029**      | 113.8461      |
| Our    | 1157.4851 | 2948.7744  | 0.0079          | 0.0059          | **91.6638**   |

Table 6. Quantitative Results (C2) of All the Methods for En-route Phase

| Model  | ADE   | FDE   | MAE (longitude) | MAE (latitude) | MAE (altitude) |
|--------|-------|-------|----------------|----------------|---------------|
| V-LSTM | 400.5694  | 769.4510  | 0.0030          | 0.0024          | 57.9083       |
| HMM    | 572.8794  | 985.4811  | 0.0049          | 0.0028          | 97.5547       |
| Our    | **228.6174**  | 359.3068  | **0.0018**      | **0.0013**      | **23.4769**   |

frame, and reaches the other aircraft. The similar result can also be found in Figure 9(d). These phenomena demonstrate that the interaction modeling makes sense in the flight trajectory prediction problem in low-altitude and crowded airspace.

6.3.2 Comparison between Models for En-route Phase. Table 5 shows the performance of each model in En-route phase in Case 1 during test time. Generally speaking, the prediction errors of en-route phase are much less than the prediction errors of other two phases because of the relative stationarity of trajectories in en-route phases in practice. HMM models can predict the trajectory more accurately than others while our model performs badly to forecast longitude and latitude. This phenomenon may be the overfitting because of the high model complexity.

From Table 6, we observe that LSTM-based models outperform the Markov process-based model in Case 2. This result is reasonable since Markov process makes an assumption that future states depend only on the current state, not on the events that occurred before it. In other words, the state transition probability is independent of time which leads to prediction bias since Markov model can not capture the temporal patterns of uncertainties from weather.
Fig. 9. Illustration of our proposed framework predicting trajectories of two aircraft descend simultaneously. (a) Overview of descending trajectories. (b) Descend stage 1. One aircraft $g_1$ as depicted in black yields the other one $g_2$ which is marked in blue line. (c) Descend stage 2. Two aircraft descend almost together, but the distance between them is increased gradually for collision avoidance. (d) The last descend stage. It is obvious that $g_1$ fly at a higher altitude than $g_2$ and prepares for descending until $g_2$ lands.

6.4 Ablation Study

To comprehensive evaluate the effectiveness of our proposed modules, we further perform ablation studies. We remove a single component in an ablation study to construct new structure. We investigate four variants of our models as follows.

- **Structural LSTM without Spatial Attention (SWSA):** This variant does not have the spatial attention module. We concatenate hidden states of edgeLSTMs and the feature vector of the node directly as the input to nodeLSTMs.

- **Structural LSTM without Motion Constraints (SWMC):** This variant does have the motion constraints in the inference process. The predicted result depends on on-time sampling.

- **Encoder-Decoder without Dual-stage Attention (EDWDA):** This variant can be thought as a naive sequence-to-sequence structure. The intermediate (encoder) vector is directly fed into decoder.
• **Encoder-Decoder with Position-only Loss Function (EDWP):** The loss function used in this variant is to minimize the distance between predicted way-points and groundtruth without taking into account the aircraft’s aerodynamics such as speed and acceleration.

In Table 7 and Table 8, we observe our proposed motion constraint mechanism plays a more important role than other components because of the dramatically increased prediction errors. Besides, the introduction of attention module also greatly improves prediction accuracy but it seems that weather information does not make sense in the takeoff phase. We think the reason is that departure flights will be canceled or delayed when the weather is pretty bad but these ground trajectories are not considered in our prediction framework. In other words, the trajectories in takeoff phase, which are used to be trained or predicted, are collected when aircraft operate under normal weather conditions.

| Model | ADE       | FDE       | MAE  |
|-------|-----------|-----------|------|
|       | longitude | latitude  | altitude |
| SWSA  | 840.8854  | 1956.4782 | 0.0069 | 0.0044 | 180.3561 |
| SWMC  | 1453.55  | 2561.5495 | 0.0124 | 0.0073 | 203.7228 |
| Our   | 312.3486  | 437.8141  | **0.0023** | **0.0019** | 54.8213 |

| Model | ADE       | FDE       | MAE  |
|-------|-----------|-----------|------|
|       | longitude | latitude  | altitude |
| SWSA  | 1165.7009 | 2241.7800 | 0.0094 | 0.0065 | 197.4589 |
| SWMC  | 2264.8614 | 4158.4823 | 0.0187 | 0.0123 | 224.2156 |
| Our   | **450.9431** | **766.2355** | **0.0036** | **0.0024** | **92.8776** |

Table 9 shows the results of ablation study for en-route phase. Different from takeoff and approach, weather information is essential in the en-route trajectory prediction. The lack of aerodynamic information in loss function mainly affects the prediction accuracy of altitude and then affects the ADE and FED. Moreover, although we observe that the dual-stage attention module does not contribute a lot to improve the prediction accuracy of coordinates in terms of MSE of latitude, longitude, and altitude, it helps reduce the prediction error of the entire predicted trajectory.

7 CONCLUSION

In this article, we present the first flight trajectory prediction framework considering aircraft-aircraft interactions and flight phase characteristics by leveraging real-world big aviation data. Compared with existing flight trajectory prediction research based on historical trajectories, this paper takes uncertainties and aerodynamics into account. We first propose a structural LSTM with physical constraints, to learn latent spatio-temporal dynamics of aircraft itself and their interactions. Then we propose a method of representation of high-dimensional weather. The compact representation of all information is tweaked to learn by the dual-attention mechanism by minimizing aerodynamics-related loss function LVA. From the experimental results on our dataset, we show the superiority...
of our proposed approach and the effectiveness of each of the components. Our approach can facilitate effective air traffic management, which is of great importance to advance aerial mobility.

ACKNOWLEDGMENTS

This research was done when authors (Zhang and Chen) were Research Assistant at Embry-Riddle Aeronautical University. We thank faculty working in the Next-Generation Advanced Research (NEAR) Lab at Embry-Riddle Aeronautical University for providing collected aerial data.

REFERENCES

[1] Federal Aviation Administration. 2009. Pilot’s handbook of aeronautical knowledge. Skyhorse Publishing Inc.
[2] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. 2016. Social lstm: Human trajectory prediction in crowded walks. In Proceedings of the IEEE conference on computer vision and pattern recognition. 961–971.
[3] Bo An, Haipeng Chen, Noseong Park, and VS Subrahmanian. 2016. MAP: Frequency-based maximization of airline profits based on an ensemble forecasting approach. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 421–430.
[4] Samet Ayhan and Hanan Samet. 2016. Aircraft trajectory prediction made easy with predictive analytics. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 21–30.
[5] Lei Bai, Lina Yao, Salll S Kanhere, Xianzhi Wang, Wei Liu, and Zheng Yang. 2019. Spatio-temporal graph convolutional and recurrent networks for citywide passenger demand prediction. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management. 2293–2296.
[6] Shane T Barratt, Mykel J Kochenderfer, and Stephen P Boyd. 2018. Learning probabilistic trajectory models of aircraft in terminal airspace from position data. IEEE Transactions on Intelligent Transportation Systems 20, 9 (2018), 3536–3545.
[7] Luis Basora, Jérôme Morio, and Corentin Mailhot. 2017. A trajectory clustering framework to analyse air traffic flows. 7th SESAR Innovation Days (2017), 1–8.
[8] Ricardo JGB Campello, Davoud Moulavi, and Jörg Sander. 2013. Density-based clustering based on hierarchical density estimates. In Pacific-Asia conference on knowledge discovery and data mining. Springer, 160–172.
[9] Rohan Chandra, Uttaran Bhattacharya, Aniket Bera, and Dinesh Manocha. 2019. Traphic: Trajectory prediction in dense and heterogeneous traffic using weighted interactions. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 8483–8492.
[10] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555 (2014).
[11] Andrew M Churchill and Michael Bloem. 2019. Clustering aircraft trajectories on the airport surface. In Proceedings of the 13th USA/Europe Air Traffic Management Research and Development Seminar, Chicago, IL, USA. 10–13.
[12] FAA. 2021. Fixes/Waypoints. Retrieved October 4, 2020 from https://www.faa.gov/air_traffic/flight_info/aeronav/aero_data/loc_id_search/Fixes_Waypoints/
[13] Shih-Hau Fang, Larry Lin, Yi-Ting Yang, Xiaotong Yu, and Zhezhuan Xu. 2019. CityTracker: Citywide Individual and Crowd Trajectory Analysis Using Hidden Markov Model. IEEE Sensors Journal 19, 17 (2019), 7693–7701.
[14] Christian Eduardo Verdonk Gallego, Víctor Fernando Gómez Comendador, Francisco Javier Sáez Nieto, Guillermo Orenge Imaz, and Rosa Maria Arnaldo Valdés. 2018. Analysis of air traffic control operational impact on aircraft vertical profiles supported by machine learning. Transportation research part C: emerging technologies 95 (2018), 883–903.
Maxime Gariel, Ashok N Srivastava, and Eric Feron. 2011. Trajectory clustering and an application to airspace monitoring. *IEEE Transactions on Intelligent Transportation Systems* 12, 4 (2011), 1511–1524.

Next generation ERAU Applied Research Lab. 2020. *ERAU Traffic Live*. Retrieved October 1, 2020 from https://www.near.aero/erau-livetraffic

Alex Graves. 2013. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850* (2013).

Xuhao Gui, Junfeng Zhang, and Zihan Peng. 2021. Trajectory clustering for arrival aircraft via new trajectory representation. *Journal of Systems Engineering and Electronics* 32, 2 (2021), 473–486.

Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. 2018. Social gan: Socially acceptable trajectories with generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2255–2264.

Edan Habler and Asaf Shabtai. 2018. Using LSTM encoder-decoder algorithm for detecting anomalous ADS-B messages. *Computers & Security* 78 (2018), 155–173.

Dirk Helbing and Peter Molnar. 1995. Social force model for pedestrian dynamics. *Physical review E* 51, 5 (1995), 4282.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

Mauro Leonardi. 2018. ADS-B anomalies and intrusions detection by sensor clocks tracking. *IEEE Trans. Aerospace Electron. Systems* 55, 5 (2018), 2370–2381.

Yi Lin, Jian-wei Zhang, and Hong Liu. 2018. An algorithm for trajectory prediction of flight plan based on relative motion between positions. *Frontiers of Information Technology & Electronic Engineering* 19, 7 (2018), 905–916.

Yi Lin, Jian-wei Zhang, and Hong Liu. 2019. Deep learning based short-term air traffic flow prediction considering temporal–spatial correlation. *Aerospace Science and Technology* 93 (2019), 105113.

Weiye Liu and Inseok Hwang. 2011. Probabilistic trajectory prediction and conflict detection for air traffic control. *Journal of Guidance, Control, and Dynamics* 34, 6 (2011), 1779–1789.

Yulan Liu and Mark Hansen. 2018. Predicting aircraft trajectories: a deep generative convolutional recurrent neural networks approach. *arXiv preprint arXiv:1812.11670* (2018).

Nishant Nikhil and Brendan Tran Morris. 2018. Convolutional neural network for trajectory prediction. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*. 0–0.

Mauro Leonardi. 2018. ADS-B anomalies and intrusions detection by sensor clocks tracking. *IEEE Trans. Aerospace Electron. Systems* 55, 5 (2018), 2370–2381.

Yi Lin, Jian-wei Zhang, and Hong Liu. 2018. An algorithm for trajectory prediction of flight plan based on relative motion between positions. *Frontiers of Information Technology & Electronic Engineering* 19, 7 (2018), 905–916.

Yi Lin, Jian-wei Zhang, and Hong Liu. 2019. Deep learning based short-term air traffic flow prediction considering temporal–spatial correlation. *Aerospace Science and Technology* 93 (2019), 105113.

Weiye Liu and Inseok Hwang. 2011. Probabilistic trajectory prediction and conflict detection for air traffic control. *Journal of Guidance, Control, and Dynamics* 34, 6 (2011), 1779–1789.

Yulan Liu and Mark Hansen. 2018. Predicting aircraft trajectories: a deep generative convolutional recurrent neural networks approach. *arXiv preprint arXiv:1812.11670* (2018).

Nishant Nikhil and Brendan Tran Morris. 2018. Convolutional neural network for trajectory prediction. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*. 0–0.

Mauro Leonardi. 2018. ADS-B anomalies and intrusions detection by sensor clocks tracking. *IEEE Trans. Aerospace Electron. Systems* 55, 5 (2018), 2370–2381.

Yi Lin, Jian-wei Zhang, and Hong Liu. 2018. An algorithm for trajectory prediction of flight plan based on relative motion between positions. *Frontiers of Information Technology & Electronic Engineering* 19, 7 (2018), 905–916.

Yi Lin, Jian-wei Zhang, and Hong Liu. 2019. Deep learning based short-term air traffic flow prediction considering temporal–spatial correlation. *Aerospace Science and Technology* 93 (2019), 105113.

Weiye Liu and Inseok Hwang. 2011. Probabilistic trajectory prediction and conflict detection for air traffic control. *Journal of Guidance, Control, and Dynamics* 34, 6 (2011), 1779–1789.

Yulan Liu and Mark Hansen. 2018. Predicting aircraft trajectories: a deep generative convolutional recurrent neural networks approach. *arXiv preprint arXiv:1812.11670* (2018).

Nishant Nikhil and Brendan Tran Morris. 2018. Convolutional neural network for trajectory prediction. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*. 0–0.

Mauro Leonardi. 2018. ADS-B anomalies and intrusions detection by sensor clocks tracking. *IEEE Trans. Aerospace Electron. Systems* 55, 5 (2018), 2370–2381.

Yi Lin, Jian-wei Zhang, and Hong Liu. 2018. An algorithm for trajectory prediction of flight plan based on relative motion between positions. *Frontiers of Information Technology & Electronic Engineering* 19, 7 (2018), 905–916.

Yi Lin, Jian-wei Zhang, and Hong Liu. 2019. Deep learning based short-term air traffic flow prediction considering temporal–spatial correlation. *Aerospace Science and Technology* 93 (2019), 105113.

Weiye Liu and Inseok Hwang. 2011. Probabilistic trajectory prediction and conflict detection for air traffic control. *Journal of Guidance, Control, and Dynamics* 34, 6 (2011), 1779–1789.

Yulan Liu and Mark Hansen. 2018. Predicting aircraft trajectories: a deep generative convolutional recurrent neural networks approach. *arXiv preprint arXiv:1812.11670* (2018).

Nishant Nikhil and Brendan Tran Morris. 2018. Convolutional neural network for trajectory prediction. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*. 0–0.

Mauro Leonardi. 2018. ADS-B anomalies and intrusions detection by sensor clocks tracking. *IEEE Trans. Aerospace Electron. Systems* 55, 5 (2018), 2370–2381.
[44] Amir Sadeghian, Vineet Kosaraju, Ali Sadeghian, Noriaki Hirose, Hamid Rezaatofghig, and Silvio Savarese. 2019. Sophie: An attentive gan for predicting paths compliant to social and physical constraints. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 1349–1358.

[45] Zhiyuan Shi, Min Xu, and Quan Pan. 2020. 4-D Flight Trajectory Prediction With Constrained LSTM Network. IEEE Transactions on Intelligent Transportation Systems (2020).

[46] Zhiyuan Shi, Min Xu, Quan Pan, Bing Yan, and Haimin Zhang. 2018. LSTM-based flight trajectory prediction. In 2018 International Joint Conference on Neural Networks (IJCNN). IEEE, 1–8.

[47] Junzi Sun, Joost Ellerbroek, and Jacco Hoekstra. 2016. Large-scale flight phase identification from ads-b data using machine learning methods. In 7th International Conference on Research in Air Transportation.

[48] Junzi Sun, Joost Ellerbroek, and Jacco Hoekstra. 2017. Flight extraction and phase identification for large automatic dependent surveillance–broadcast datasets. Journal of Aerospace Information Systems 14, 10 (2017), 566–572.

[49] Asma Tabassum, Nicholas Allen, and William Semke. 2017. ADS-B message contents evaluation and breakdown of anomalies. In 2017 IEEE/AIAA 36th Digital Avionics Systems Conference (DASC). IEEE, 1–8.

[50] Peter Trautman and Andreas Krause. 2010. Unfreezing the robot: Navigation in dense, interacting crowds. In 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems.

[51] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In Advances in neural information processing systems. 5998–6008.

[52] Anirudh Vemula, Katharina Muelling, and Jean Oh. 2018. Social attention: Modeling attention in human crowds. In 2018 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 4601–4607.

[53] Guoqing Wang and Wenhao Zhao. 2020. Chapter 3 - The requirement organization of the avionics system. In The Principles of Integrated Technology in Avionics Systems, Guoqing Wang and Wenhao Zhao (Eds.). Academic Press, 103 – 167. https://doi.org/10.1016/B978-0-12-816651-2.00003-4

[54] Yuandong Wang, Hongzhi Yin, Hongxu Chen, Tianyu Wo, Jie Xu, and Kai Zheng. 2019. Origin-destination matrix prediction via graph convolution: a new perspective of passenger demand modeling. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 1227–1235.

[55] David Warnock-Smith, John F O’Connell, and Mahnaz Maleki. 2017. An analysis of ongoing trends in airline ancillary revenues. Journal of Air Transport Management 64 (2017), 42–54.

[56] Hanwool Woo, Yonghoon Ji, Hitoshi Kono, Yusuke Tamura, Yasuhide Kuroda, Takashi Sugano, Yasunori Yamamoto, Atsushi Yamashita, and Hajime Asama. 2017. Lane-change detection based on vehicle-trajectory prediction. IEEE Robotics and Automation Letters 2, 2 (2017), 1109–1116.

[57] Kai Zhang, Yushan Jiang, Dahai Liu, and Houbing Song. 2020. Spatio-Temporal Data Mining for Aviation Delay Prediction. In 2020 IEEE 39th International Performance Computing and Communications Conference (IPCCC). IEEE, 1–7.