

Received July 16, 2020, accepted August 10, 2020, date of publication August 13, 2020, date of current version August 27, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3016289

A Deep Learning Approach for Aircraft Trajectory Prediction in Terminal Airspace

WEILI ZENG¹, ZHIBIN QUAN^{1,2,4}, (Member, IEEE), ZIYU ZHAO¹, CHAO XIE³, AND XIAOBO LU⁴

¹College of Civil Aviation, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China

²Department of Computer and Information Science, Faculty of Science and Technology, University of Macau, Macau 999078, China

³College of Mechanical and Electronic Engineering, Nanjing Forestry University, Nanjing 210037, China

⁴School of Automation, Southeast University, Nanjing 210096, China

Corresponding author: Zhibin Quan (zbquan@um.edu.mo)

This work was supported in part by the Fundamental Research Funds for the Central Universities under Grant NS2018044 and Grant 2242020R20014, in part by the National Natural Science Foundation of China under Grant 61901221, and in part by the Macao Young Scholars Program under Grant AM201914.

ABSTRACT Current state-of-the-art trajectory methods do not perform well in the terminal airspace that surrounds an airport due to its complex airspace structure and the frequently changing flight postures of aircraft. Since an aircraft that takes off or lands in an airport must follow a specified procedure, this paper will learn a data-driven trajectory prediction model from many historical trajectories to improve the accuracy and robustness of trajectory prediction in the terminal airspace. A regularization method is utilized to reconstruct each aircraft trajectory to obtain a high-quality trajectory with equal time intervals and no noise. Furthermore, we formulate the 4D trajectory prediction problem as a sequence-to-sequence learning problem, and we propose a sequence-to-sequence deep long short-term memory network (SS-DLSTM) for trajectory prediction, which can effectively capture the long and short temporal dependencies and the repetitive nature among trajectories. The proposed model is composed of an encoding module and a decoding module, where the encoding module realizes the feature representation of historical trajectories, while the decoding module accepts the output of the encoding module as its initial input and recursively outputs the predicted trajectory sequence. The proposed method is applied to a dataset for the terminal airspace in Guangzhou, China. The experimental results demonstrate that our approach has relatively high robustness and outperforms mainstream data-driven trajectory prediction methods in terms of accuracy.

INDEX TERMS Aircraft trajectory prediction, terminal airspace, trajectory reconstruction, regularization method, long short-term memory network.

I. INTRODUCTION

The terminal airspace that surrounds an airport is the area with the highest flight density and the most complex structure. According to statistical data that were released by Boeing, 60% of the fatal accidents of the worldwide commercial jet fleet from 2007 to 2016 occurred in the phases of takeoff, initial climb, final approach and landing [1]. Even though these four phases comprise only 6% of the flight time, they pose substantial threats to the safety of air transportation. Therefore, most research and development of decision support tools (DSTs) focus on the terminal airspace, with the objective of providing controllers with automated tools for detecting and resolving aircraft conflicts, sequencing the

arrivals and departures of flights, and monitoring anomalous behaviors such as the low-altitude alert and flight path deviation alert. However, the effective operation of these automated DSTs almost completely depends on the prediction of aircraft trajectories with high accuracy and reliability.

Many approaches for aircraft trajectory forecasting have been proposed and can be categorized as the state estimation method, aerodynamic-model-based method, data-driven method, and combination method. The state estimation methods regard the aircraft trajectory prediction problem as a tracking problem and use tracking methods such as Kalman filtering and the hidden Markov model to predict the future trajectory of an aircraft [2]–[4]. In contrast to most previous state estimation prediction approaches, which simply map the past and the current flight states into the future, Liu and Hwang [5] combined the intent information of

The associate editor coordinating the review of this manuscript and approving it for publication was Zhenbao Liu.

aircraft with the dynamics of the aircraft to obtain more accurate prediction results. A stochastic linear hybrid system that utilizes a state-dependent transition model and a Markov transition model is presented for describing the aircraft dynamics with changing flight modes. Based on an alignment and fusion process, Ayhan and Samet [6] employed a set of 4D cubes to represent aircraft trajectories. They learned a stochastic model, namely, a hidden Markov model (HMM), from the historical trajectories, aircraft specifications, and related weather information.

The aerodynamic-model-based method can be seen as a kind of physical model, which is based on the forces that act on the aircraft to predict the successive points of the future trajectory [7], [8]. Schuster *et al.* [9] proposed a 4D trajectory prediction model for the entire *en route* phases by fully utilizing the aircraft state and intent information, which are composed of three-level units: basic data (initial aircraft state, wind field, and aircraft performance data), the aircraft dynamics system and the flight management system. However, the accuracy of this approach is relatively low in the climbing and descent phases, as this method is unable to capture the change in the aircraft configuration between climb-modes effectively. Baklacioglu and Cavcar [10] developed a new aero-propulsive trajectory prediction model for the climbing and descent phases. This technique took into account the effects of the compressible drag above the critical Mach number and the compressibility and profile camber. Zhang *et al.* [11] proposed an online 4D trajectory forecasting method on the tactical level that was based on updating the aircraft intention, which consisted of the flight plan and adaptation data (the controlled airspace, air routes, standard instrument departure, and standard terminal arrival route). As we know, the aerodynamic-model-based method performs poorly on real scenarios, as it depends on many unknown or partially known parameters.

The data-driven method uses large amounts of historical data related to the aircraft trajectory to train an aircraft predictor, which does not use any aircraft aerodynamic parameters. Le Fablec and Alliot [12] focused on aircraft trajectory prediction in the vertical plane. Their method is based on the back-propagation (BP) neural network, which is trained using a set of historical trajectories. However, this algorithm does not perform well because it uses only one hidden layer and a small amount of trajectory data. Tastambekov *et al.* [13] proposed a data-driven method, namely, local linear functional regression, for predicting short- to mid-term aircraft trajectories. The dataset used in this study contains only the past radar tracks and does not include any aeronautical or physical parameters. According to the authors, their method performs similarly to traditional neural networks. Wang *et al.* [14] introduced a model for solving the short-term trajectory prediction forecasting problem in the terminal airspace via a machine learning technique. It is composed of two parts: trajectory clustering and a BP neural network. It is difficult for this algorithm to capture the flight pattern in complex terminal airspace due to its use of a neural network that

has only one hidden layer. Hernández *et al.* [15] formulated the aircraft trajectory prediction problem as a regression problem and used traditional ensemble machine learning algorithms to solve it. Several sophisticated approaches, which include a regression method that uses the wavelet decomposition, density-based spatial clustering of applications with noise (DBSCAN) clustering, and the Kalman filter algorithm, have been used to generate a complete trajectory. Barratt *et al.* [16] developed two probabilistic generative models for takeoffs and landings in the terminal airspace of an airport. The method uses a clustering method to mine flight modes from a historical dataset of radar-based position measures. Given partial samples of a trajectory, the model produces a prediction that is generated from the posterior distribution.

The combination methods combine two or more of the previously discussed methods. Alligier *et al.* [17] focused on trajectory prediction in the climbing phase and learned parameters such as the mass and thrust from past observations to improve the accuracy of aircraft trajectory forecasting. The thrust law and the weight are learned from historical data and previous trajectory points, respectively. Furthermore, Alligier and Gianazza [18] utilized additional data to determine aircraft operational factors, such as the mass and the speed intent. They studied 11 frequently used aircraft types and a sizable ADS-B dataset that included 1520 airports. Guan *et al.* [19] used the historical flight data and the BADA dataset for a specified aircraft type as the training sample for the learning machine to build an initial predictive model. Then, the current state information and the output of the initial predictive model were integrated with a physical model to output the predicted trajectory.

Though many trajectory prediction methods for aircraft have been proposed, the current state-of-the-art methods do not perform well in the terminal airspace. The main reason for the low accuracy and reliability of these prediction methods is that they cannot accurately capture the features of nonmodeled behaviors, which are hidden in the historical flight trajectories. The development of a prediction model that can learn the nonmodeled behaviors from historical data would facilitate the improvement of accuracy and reliability. It could enable the accuracy of trajectory prediction to reach an unprecedented level due to the advancement of big data analytics and machine learning algorithms. This paper focuses on the construction of a data-driven trajectory forecasting model for the terminal airspace from a large amount of ADS-B surveillance data.

As we know, the flight environment in the terminal airspace is more complex than that in other maneuvering airspaces. However, aircraft in the terminal airspace surrounding an airport must comply with the arrival and departure modes, which are repeatedly observed in the historical trajectories due to the interrelationships among the trajectories. To a large extent, trajectories of aircraft have clear temporal and spatial patterns, demonstrating relatively high predictability. Hence, a useful predictor should be able to capture not only

the short-term dependencies, which reflect the local changes of the aircraft trajectory, but also the repetition among the trajectories.

The long short-term memory (LSTM) neural network is an extension of the traditional recurrent neural network (RNN), which has become the mainstream framework for sequence modeling [20]. Since the proposal of LSTM by Hochreiter and Schmidhuber [21], many variants of LSTM have been developed, such as gated LSTM [22], convolution LSTM [23], deep LSTM [24], and sequence-to-sequence LSTM [25]. Since LSTM and its variants perform well in capturing the long-temporal dependencies of sequential data, they have been widely applied in many sectors, such as the energy sector [26], [27], economic industry [28], [29], medical sector [30], [31], and transportation sector [32]–[34]. Inspired by the excellent performance of LSTM and its variants, and considering the operational characteristics of the terminal airspace that surrounds an airport, we extend the LSTM architecture to aircraft trajectory prediction.

This paper focuses on a short-time trajectory prediction model that employs a RNN framework in the terminal airspace. Our objective is to forecast multistep trajectory points, which can be formulated as a sequence-to-sequence learning problem, in which the input sequence is the previous states of multiple trajectory points of an aircraft, and the output is the remaining flight trajectory points. Taking a departure flight as an example, the states of some trajectory points of an aircraft after taking off from the airport are used as the input sequence, while the output sequence consists of all 4D trajectory points of an aircraft between the starting prediction time and the final time prior to leaving the terminal airspace. To solve this sequential learning problem, a sequence-to-sequence deep LSTM (SS-DLSTM) neural network is used; it contains an encoding module and a decoding module, where the encoding module learns from the past aircraft states, and the output of the encoding module is used as the input of the decoding module. The main contributions of this paper as follows:

(a) To address the unequal time intervals and the outliers in the original trajectories, a regularization model is proposed for reconstructing the original trajectory into a trajectory with equal time intervals and no outliers, which yields relatively high-quality data for subsequent trajectory prediction.

(b) In consideration of the satisfactory performance of SS-DLSTM in characterizing the long and short temporal dependencies and the repetitiveness among sequences, this paper extends SS-DLSTM to aircraft trajectory prediction for the first time. It constructs a model that is more in line with the flying features of an aircraft in terminal airspace, as aircraft must comply with takeoff / landing procedures, which exhibits repetitive patterns among trajectories.

(c) Unlike the state-of-the-art methods that use a large variety of data such as trajectory data, aircraft performance data, and meteorological data, the proposed method takes into account only trajectory-related information, such as the 3D position, course, velocity and aircraft type. The aircraft

performance can be indirectly learned from a large amount of historical data by using the aircraft type as an input attribute, which facilitates the improvement of the accuracy of trajectory prediction.

The remainder of this paper is organized as follows. Section II describes the data processing and feature selection. Section III elaborates on aircraft prediction. Section IV describes the regularization reconstruction and the trajectory prediction proposed in this paper. Section V compares the proposed model with the mainstream models via experiments. Finally, this study is summarized in Section VI.

II. PROBLEM FORMULATION

The aim of this paper is to learn a data-driven prediction model from a large amount of ADS-B data for arrival and departure flights in the terminal airspace that surrounds an airport. Given the previous state information of an aircraft, the trained model can output the 3D position of each timestamp in the subsequent few minutes simultaneously. As introduced in Section III. B, six attributes are utilized to characterize the state of each timestamp of an aircraft. Therefore, the input and output of the prediction model can be described mathematically as two sequences. Without loss of generality, we assume that the aircraft state contains K attributes. Let $X^{(t)} = (x_1^{(t)}, x_2^{(t)}, \dots, x_K^{(t)}) \in \mathbb{R}^K$ and $Y^{(t)} = (y_1^{(t)}, y_2^{(t)}, y_3^{(t)}) \in \mathbb{R}^3$ denote the aircraft state vector and the 3D position vector, respectively, at timestamp t . To predict the trajectory after time t , a function $F(\cdot)$ that maps the state sequence of length t to the future 3D position sequence of length S is learned:

$$\left[X^{(1)}, \dots, X^{(t)} \right] \xrightarrow{F(\cdot)} \left[Y^{(t+1)}, \dots, Y^{(t+S)} \right] \quad (1)$$

According to (1), our objective is to learn a sequence-to-sequence forecasting model. Compared to other airspaces, the primary challenge of aircraft trajectory prediction for the terminal airspace is that the flight attitude and course change frequently. However, fortunately, aircraft in the terminal airspace must follow various arrival and departure procedures. Aircraft that are executed with the same procedure have similar trajectories under similar flight conditions. Thus, the accuracy of the prediction model depends on whether the model can capture sufficiently accurately the long- and short-term dependencies and the repetitiveness among trajectory sequences.

Our algorithm for trajectory prediction is composed of three parts: data preparation, trajectory reconstruction, and trajectory prediction, as illustrated in Fig. 1. The data preparation process, which includes data quality analysis, data cleaning, coordinate transformation, feature construction, and normalization, is described in Section III.A. Trajectory reconstruction aims at transforming the original trajectory into an equal time-interval trajectory without noise, thereby guaranteeing that the subsequent trajectory prediction has high-quality input data. The trajectory prediction model contains an encoding module and a decoding module.

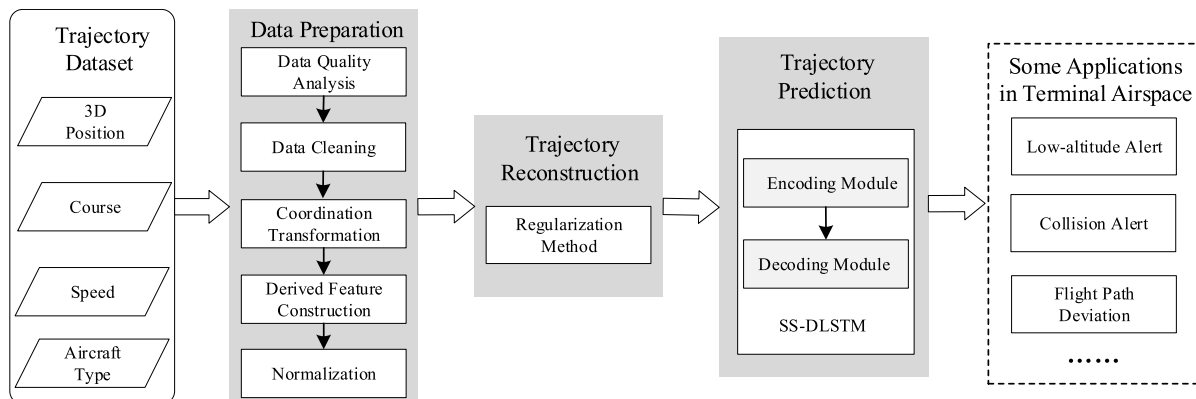


FIGURE 1. Overview of the proposed SS-DLSTM method for trajectory prediction.

The encoding module inputs the historical aircraft state and outputs a feature vector. The decoding module accepts the output of the encoding module as the initial input and outputs the 3D position sequence recursively until the end timestamp. In Section IV, we will describe the trajectory reconstruction and trajectory prediction processes in detail.

III. DATA PREPARATION

A. ADS-B DATASET

The ADS-B dataset used in this paper is derived from data from the VariFlight technology company (<http://flightadsb.variflight.com/tracker/>). It contains all monitoring data of landing and takeoff aircraft at Baiyun International Airport (GBIA) of China from August 1 to November 30, 2018. Each item of data consists of several attributes, which include the flight ID, latitude, longitude, altitude, velocity, course, departure location, arrival location, aircraft type, and monitoring time. The center of the airport in GBIA is defined in earth-centered, earth-fixed coordinates. The original measurements are transformed to east-north-up coordinates via a stereographic projection. We keep only the position measurements that are less than 20 km in the “east” and “north” dimensions and less than 2000 meters in the “up” dimension. A trajectory is deleted if there are two adjacent track points between which the time interval exceeds 20 seconds. In terms of the departure location and the arrival location, all trajectories in the dataset are divided into two subdatasets: a takeoff aircraft dataset and a landing aircraft dataset.

B. ATTRIBUTES

This paper aims to train a short-time aircraft prediction model that is based on a neural network architecture from a large amount of historical data that are related to trajectories. In contrast to the aerodynamic-model-based method, we do not require knowledge of the flight dynamics equations or the onboard flight intent. This paper considers the basic attributes that are related to the flight path and their corresponding derived attributes. The basic attributes are obtained directly

from the surveillance information system and include the aircraft type and five state attributes: aircraft longitude, latitude, altitude, velocity, and course. The derived attributes are the changing speed and acceleration according to the five state attributes.

1) BASIC ATTRIBUTES

- Three-dimensional (3D) position (longitude, latitude, altitude): Four-dimensional trajectory prediction aims at estimating the future trajectory by using the current and historical states of an aircraft and its relevant information, which is defined as a time sequence of 3D position (longitude, latitude, height) and time. Because the position information has short-term and long-term correlations, the use of historical position information as the input attributes of a model will facilitate the improvement of the accuracy of trajectory prediction.
- Course: The course of an aircraft is the main direction of control of the aircraft, and it is measured by the angle between the projection of the vertical axis of the aircraft and a baseline of the horizon plane. A flight is typically required to follow a prespecified flight route that consists of straight lines between waypoints. An aircraft theoretically adopts the same course between two neighboring waypoints. Therefore, the change of the course indirectly characterizes the flight intention of an aircraft by reflecting the flight trend.
- Velocity: The velocity of an aircraft represents its motion feature, which is easily affected by differences in pilot operation, restrictions by air traffic rules, impacts of the atmospheric environment, and the state of the aircraft. However, the velocity directly affects the position information at the next time point. Similar to the course of an aircraft, it can indirectly reflect intentions and trends.
- Aircraft type: Aircraft types differ in terms of performance parameters, such as the engine thrust, flight resistance, lift, and weight of the aircraft, which also change dynamically with the flight phases. Currently, the mainstream aerodynamic-model-based method

considers various performance parameters of aircraft in predicting trajectories and is a standard general model. It is difficult to obtain accurate values for the parameters of an aircraft during a flight because they are easily affected by a variety of factors, thereby leading to low accuracy and poor robustness of the prediction model. In this paper, the number of aircraft types is used as an input feature to capture the dynamic performance modes in various flight phases from many historical trajectories. In contrast to other features, the type of aircraft is a character attribute, which is coded via the one-hot encoding method [35].

2) DERIVED ATTRIBUTES

An aircraft trajectory is a time-related sequence. For the trajectory time series, the position of an aircraft at the next timestamp is strongly correlated with the changing speeds of the features of previous timestamps. The change of each feature can be characterized by speed and acceleration, and the first derivative and the second derivative are leveraged to construct the speed and acceleration time sequences, respectively. Section III. A introduced six basic attributes. Apart from the aircraft type, the attributes are numerical attributes, which can be used to calculate the speed and acceleration. The central difference method is widely used in the literature to calculate the speed and the acceleration of a trajectory sequence [36].

IV. METHODOLOGY

This section will introduce a regularized method for trajectory reconstruction for obtaining high-quality trajectory data. Then, it will describe the construction of a SS-DLSTM model for trajectory forecasting.

A. TRAJECTORY RECONSTRUCTION

Various quality problems are encountered with the ADS-B data, such as errors in the position data and loss of data, which have been analyzed by many researchers [37]–[41]. Various quality analyses of the ADS-B data are utilized in this work, and we identified two main problems with the original trajectories in the dataset. First, some trajectories may contain outliers (such as noises) at a specified timestamp (Fig. 2). Second, the time intervals are not of equal length.

An effective trajectory predictor relies strongly on high-quality data. Trajectory reconstruction is the process of using a low-quality trajectory to produce a higher quality trajectory. Many researchers solved the two problems that are discussed above with ADS-B or Radar data separately, namely, by removing noises with a low-pass filter and by using an interpolation method to fill in the null values. Motivated by the exemplary results of the regularization model for image reconstruction [10], a regularized reconstruction model is proposed for acquiring a high-quality trajectory in a unified model. The reconstructed trajectory has three properties: (a) the time intervals are of equal lengths; (b) the time resolution is higher than that of the original trajectory; and (c) all

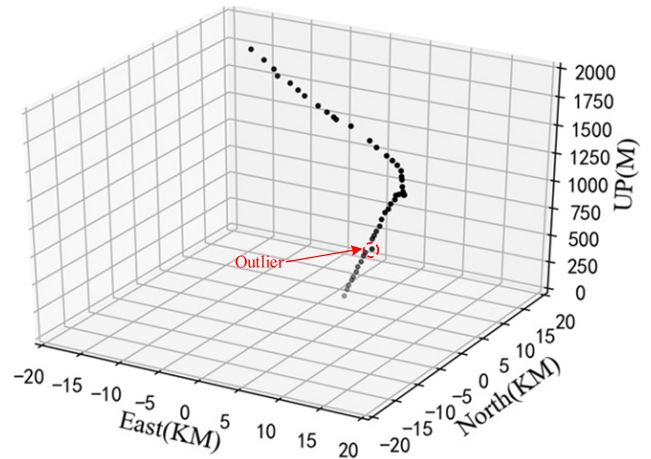


FIGURE 2. A representative trajectory with outliers.

outliers have been discarded, and the missing data have been reconstructed based on their neighboring points.

Let $\mathbf{P} \in \mathbb{R}^{N \times K}$ be the high-quality trajectory with N equal-time-gap points, and let $\mathbf{P}_0 \in \mathbb{R}^{M \times K}$ be the corresponding observed trajectory with M unequal-time-interval points. The following linear equation expresses the relationship between the high-quality trajectory and the observed trajectory:

$$\mathbf{P}_0 = \mathbf{A}\mathbf{P} + \mathbf{n} \tag{2}$$

where $\mathbf{A} \in \mathbb{R}^{M \times N}$ is a sampling matrix and \mathbf{n} denotes the random noises. The objective of trajectory reconstruction is to recover the high-quality trajectory \mathbf{P} from the observed trajectory \mathbf{P}_0 . However, this problem is an ill-posed problem, and we cannot obtain the solution directly from (2).

A regularization method is an effective approach for solve ill-posed problems, which has been proven effective in a wide range of applications, such as image reconstruction, denoising, and compressed sensing. Its objective is to improve the signal quality by imposing prior knowledge on the desired signals. We will apply and extend the regularization method to aircraft trajectory reconstruction, which is equivalent to solving the following optimization problem:

$$\bar{\mathbf{P}} = \underset{\mathbf{P}}{\text{ArgMin}} \left\{ \|\mathbf{A}\mathbf{P} - \mathbf{P}_0\|_F^2 + \lambda \rho(\mathbf{P}) \right\} \tag{3}$$

where $\|\mathbf{A}\mathbf{P} - \mathbf{P}_0\|_F^2$ represents a measure of the fit; $\rho(\cdot)$ is the regularization cost function; λ is the regularization parameter, which is deployed to control the trade-off between the fitting term and the regularization term; and $\|\cdot\|_F$ is the Frobenius norm of a matrix. The first term of (3) encourages the reconstructed trajectory to be close to the measured points, and the second term encourages it to smooth the measured points, which is especially useful for suppressing noises.

Most importantly, a regularization function should balance noise removal and preservation of the local variation tendency of the trajectory. Many formulations are available for the regularization term [42]. For simplicity of implementation, this paper adopts the Tikhonov cost function, which is one of

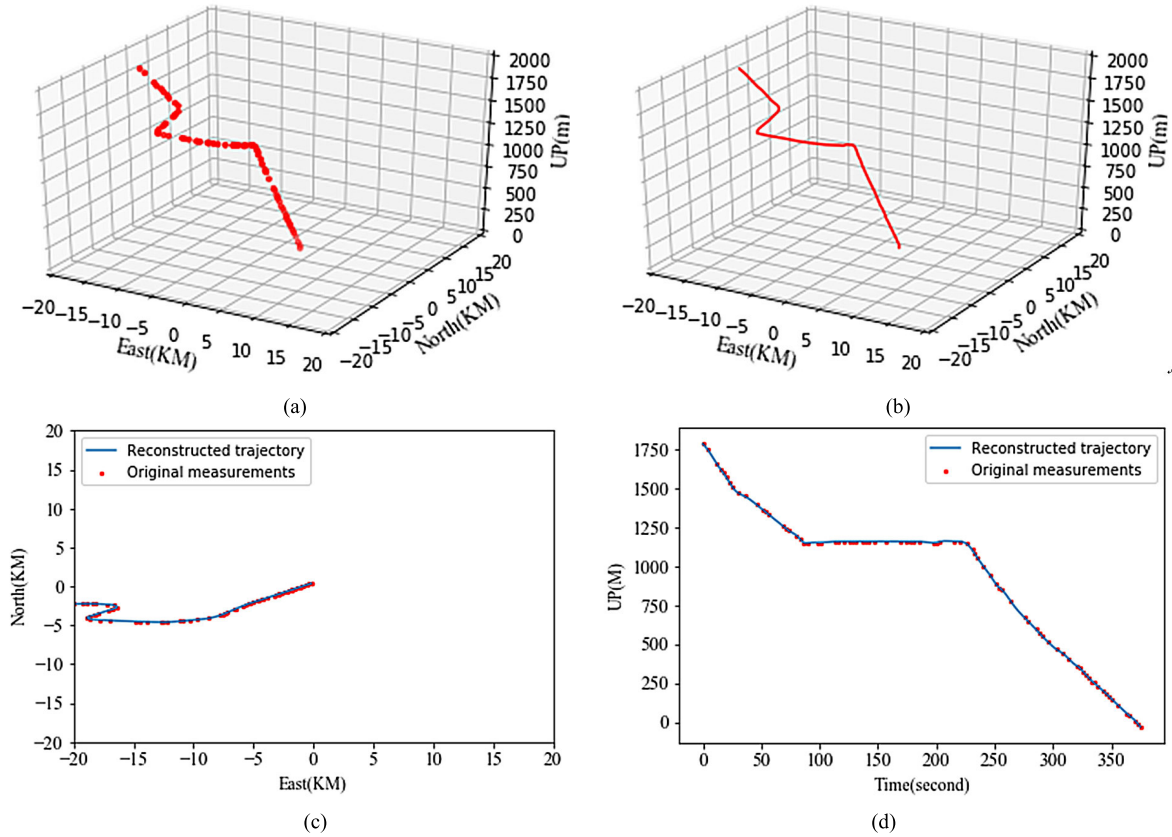


FIGURE 3. The original and the reconstructed trajectory for landing at GBIA.

the most widely referenced regularization cost functions in signal processing, as the regularization term. It is defined as

$$\rho(\mathbf{P}) = \|\Delta \mathbf{P}\|_F^2 \quad (4)$$

where $\mathbf{\Gamma}$ is the Laplacian matrix. The discrete form of the Laplacian operator is approximated by

$$(\mathbf{\Gamma P})_i = P_{i-1} - 2P_i + P_{i+1} \quad (5)$$

The Tikhonov regulation term enforces the local spatial smoothness of the trajectory sequence. As the noisy points contain high-frequency energy, they will be removed in the regularization process, and the reconstructed trajectory will not contain outliers.

Substituting the regularization term in (3) with (4), the trajectory reconstruction can be reformulated as the minimization of the following objective function:

$$f(\mathbf{P}) = \|\mathbf{A P} - \mathbf{P}_0\|_F^2 + \lambda \|\mathbf{\Gamma P}\|_F^2 \quad (6)$$

The objective function (6) is a convex quadratic function, which can be solved analytically. A solution \mathbf{P} minimizes f if and only if $\nabla f(\mathbf{P}) = 0$, which is equivalent to

$$\mathbf{P} = (\mathbf{A}^T \mathbf{A} + \lambda \mathbf{\Gamma}^T \mathbf{\Gamma})^{-1} \mathbf{A}^T \mathbf{P}_0 \quad (7)$$

Here, we present two random examples from the reconstructed trajectories. Fig. 3 and Fig. 4 show the reconstructed results for landing and takeoff, respectively, at GBIA.

Obviously, the reconstructed trajectories can fit the original measurements well.

B. SS-DLSTM PREDICTION MODEL

The sequence-to-sequence LSTM is a recently designed model for solving a neural machine translation problem, and substantially outperforms other variants of LSTM due to its ability to more accurately characterize the long and short temporal dependencies and the repetitiveness among time sequences. This paper extends this model to the prediction of aircraft trajectories in the terminal airspace, as aircraft must conform to takeoff and landing procedures, which results in repetitive patterns among trajectories. The proposed SS-DLSTM trajectory prediction model consists of two modules: an encoding module and a decoding module. The encoding module inputs the past and current states of aircraft and produces hidden feature vectors that represent long and short temporal relationships and repetitiveness among trajectory sequences. The decoding module maps the output of the encoding module to a trajectory sequence of fixed length. The encoding module and decoding module adopt deep architectures that differ in terms of their LSTM processing units.

The architectures of the encoding module and the decoding module are illustrated in Fig. 5 and Fig. 6, respectively, where J denotes the total number of layers, $C_j^{(i)}$ denotes

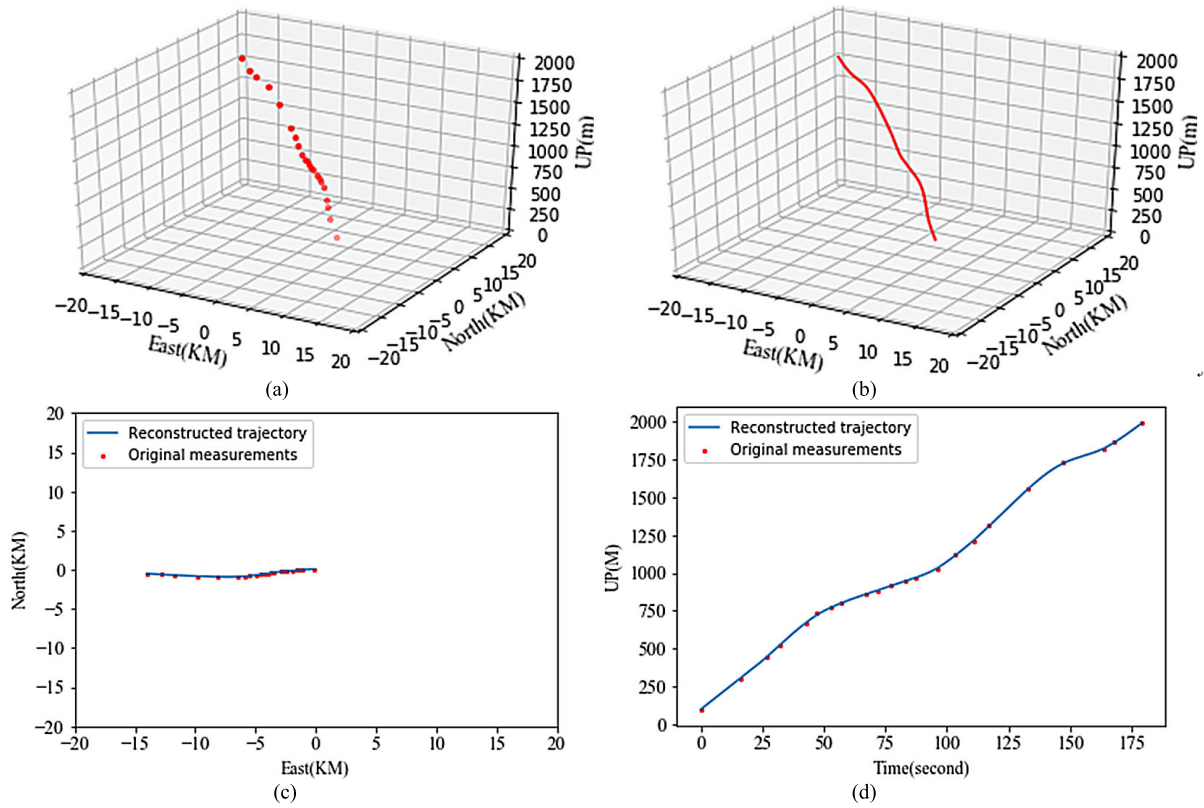


FIGURE 4. The original and the reconstructed trajectory for takeoff at GBIA.

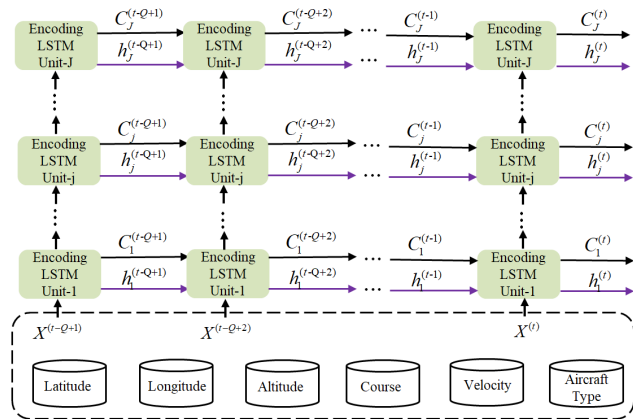


FIGURE 5. Deep architecture of the encoding module for the aircraft trajectory prediction.

the long-term hidden-state feature of the j th layer at time t , and $h_j^{(t)}$ denotes the short-term hidden-state feature of the j th layer at time t . The hidden features of the output of each layer in the encoding stage are regarded as the initial input of the corresponding layer of the decoding. Although the encoding module and the decoding module have the same layers, the LSTM processing units differ. As the activation function, we use the parametric rectified linear unit (PReLU) since it is a representative unsaturated activation function

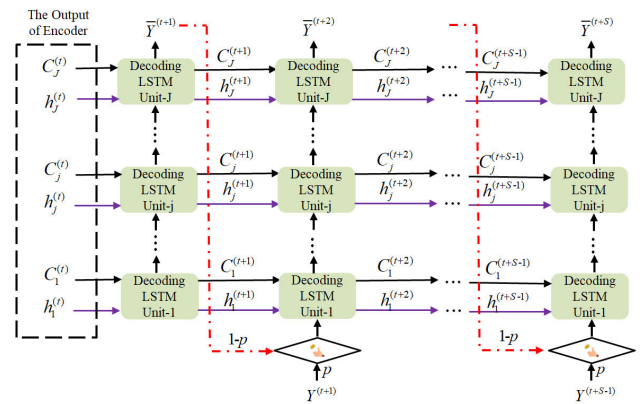


FIGURE 6. Deep architecture of the decoding module for the aircraft trajectory prediction.

with lower overfitting risk than the saturated ReLU, which is widely used in natural language processing. In the following, we will describe the updating of the hidden features for the encoding module and the decoding module at each time point and the training of the proposed model.

1) ENCODING MODULE

For layer $j = 1$, the information that is used to update the long-term hidden feature and the short-term hidden

feature comes from the current aircraft state and from previous long-term hidden features and short-term hidden features. Without loss of generality, to update the hidden features $C_j^{(\tau)}$ and $h_j^{(\tau)}$ for any layer $j \in [2, J]$ and timestamp $\tau \in [t - Q - 1, t]$, the encoding module considers the hidden feature vector $h_{j-1}^{(\tau)}$ of layer $j - 1$ at time τ , the hidden feature vector $h_j^{(\tau-1)}$ of layer j at time $\tau - 1$, and the long-memory hidden feature vector $C_j^{(\tau-1)}$ of layer j at time $\tau - 1$. The following formulas describe the updating process:

$$\begin{cases} g_{j,\tau}^e = \sigma \left(W_{g,j}^e h_j^{(\tau-1)} + U_g^e \left(h_{j-1}^{(\tau)} \right) + b_{g,j}^e \right) \\ i_{j,\tau}^e = \sigma \left(W_{i,j}^e h_j^{(\tau-1)} + U_i^e \left(h_{j-1}^{(\tau)} \right) + b_{i,j}^e \right) \\ o_{j,\tau}^e = \sigma \left(W_{o,j}^e h_j^{(\tau-1)} + U_o^e \left(h_{j-1}^{(\tau)} \right) + b_{o,j}^e \right) \\ a_{j,\tau}^e = \tanh \left(W_{a,j}^e h_j^{(\tau-1)} + U_{a,j}^e \left(h_{j-1}^{(\tau)} \right) + b_{a,j}^e \right) \\ C_j^{(\tau)} = g_{j,\tau}^e \otimes C_j^{(\tau-1)} + i_{j,\tau}^e \otimes a_{j,\tau}^e \\ h_j^{(\tau)} = o_{j,\tau}^e \otimes \tanh \left(C_j^{(\tau)} \right) \end{cases} \quad (8)$$

where $\sigma(\cdot)$ is the sigmoid activation function; $\tanh(\cdot)$ is the hyperbolic tangent function; $g_{j,\tau}^e$ denotes the proportion of the cell state information that is propagated from time $\tau - 1$ to time τ ; $i_{j,\tau}^e$ denotes the proportion of previous hidden-state information and the current input information that is stored in the current cell; $o_{j,\tau}^e$ denotes the output proportion of the current updated cell; and $W_{g,j}^e$, U_g^e , $b_{g,j}^e$, $W_{i,j}^e$, U_i^e , $b_{i,j}^e$, $W_{o,j}^e$, $b_{o,j}^e$, $U_{o,j}^e$, $W_{a,j}^e$, $U_{a,j}^e$, and $b_{a,j}^e$ are neural network parameters for the encoding process that are learned from the historical flight trajectories.

2) DECODING MODULE

The aim of the output decoding module is to produce multi-step track points that are based on the output of the encoding module. The model training and online inference processes differ substantially. During the training process, we can access the true trajectory at the previous timestamp, which can be utilized as the input of the next step. However, true track points are not available in the online prediction process. In this paper, a sampling mechanism that was proposed by Bengio *et al.* [43] is adopted to determine the input of each step for the decoding process, thereby bridging the gap between model training and online prediction. The mini-batch-based stochastic gradient descent approach is used to train the model, and the estimated value from the model with probability $1 - p$ or the previous real value with probability p is utilized as the input of the current step. Intuitively, at the beginning of the model training process, it would be helpful to select the actual previous track point since the model is not fully trained, while preferentially selecting the estimated value from the model at the end of the training, as this is consistent with the practical scenario. This process is illustrated in Fig. 6.

Under the sampling mechanism, the cell feature vector and the hidden feature vector for the first layer $j = 1$ at timestamp

$\tau \in [t + 1, t + S]$ can be updated as follows:

$$\begin{cases} g_{1,\tau}^d = \sigma \left(W_{g,1}^d h_1^{(\tau-1)} + U_g^d \left(\text{rand} \left\{ \bar{Y}^{(\tau)}, Y^{(\tau)} \right\} \right) + b_{g,1}^d \right) \\ i_{1,\tau}^d = \sigma \left(W_{i,1}^d h_1^{(\tau-1)} + U_i^d \left(\text{rand} \left\{ \bar{Y}^{(\tau)}, Y^{(\tau)} \right\} \right) + b_{i,1}^d \right) \\ o_{1,\tau}^d = \sigma \left(W_{o,1}^e h_1^{(\tau-1)} + U_o^e \left(\text{rand} \left\{ \bar{Y}^{(\tau)}, Y^{(\tau)} \right\} \right) + b_{o,1}^d \right) \\ a_{1,\tau}^d = \tanh \left(W_{a,1}^d h_1^{(\tau-1)} + U_{a,1}^d \left(\text{rand} \left\{ \bar{Y}^{(\tau)}, Y^{(\tau)} \right\} \right) + b_{a,1}^d \right) \\ C_1^{(\tau)} = g_{1,\tau}^d \otimes C_1^{(\tau-1)} + i_{1,\tau}^d \otimes a_{1,\tau}^d \\ h_1^{(\tau)} = o_{1,\tau}^d \otimes \tanh \left(C_1^{(\tau)} \right) \end{cases} \quad (9)$$

where $\text{rand}\{\bar{Y}^{(\tau)}, Y^{(\tau)}\}$ denotes the random selection function between the estimated and the true track points, and other parameters are defined similarly to the corresponding parameters in the updating process of the decoding module. For any layer $j \in [2, J]$ at a time $\tau \in [t + 1, t + S]$, the updating processes of the cell feature vector and the hidden feature vector are similar to that of the encoding module.

3) MODEL TRAINING

The training objective is to minimize the error between the estimated output and the actual output. Natural language translation [43] is a discrete state problem, whereas trajectory prediction is a continuous state problem. Therefore, we cannot use the posterior probability as a cost function, which is widely used in discrete state problems. Here, the following square error cost function is used to learn the parameters of the encoding module and the decoding module simultaneously:

$$\text{loss}(W) = \sum_{i=1}^C \sum_{\tau=1}^S \|\bar{Y}_i^{(\tau)} - Y_i^{(\tau)}\|_2^2 \quad (10)$$

where $\|\cdot\|_2$ denotes the 2-norm; C is the total number of samples; $\bar{Y}_i^{(\tau)}$ and $Y_i^{(\tau)}$ are the estimated and true trajectory positions, respectively, at timestamp τ .

The gradient descent method is typically used to solve neural network models. The gradient descent method has three commonly used forms: batch gradient descent, stochastic gradient descent, and mini-batch stochastic gradient descent. Due to the large number of trajectory samples that are considered in this paper, to reduce the number of iterations needed for convergence to the globally optimal solution, this paper uses the minimum batch random gradient descent method to identify the optimal parameters, thereby ensuring high performance while reducing the number of epochs. The training process is described in detail as follows:

Step 1: Data preparation: First, the original trajectory dataset is filtered according to the range of the considered airspace and the basic requirements for trajectories, and the regularization method is used to reconstruct the filtered trajectories. Then, the derived attributes are constructed to obtain a trajectory dataset that contains the basic attributes and the derived attributes. To eliminate dimensional differences among the attributes, the maximum-minimum

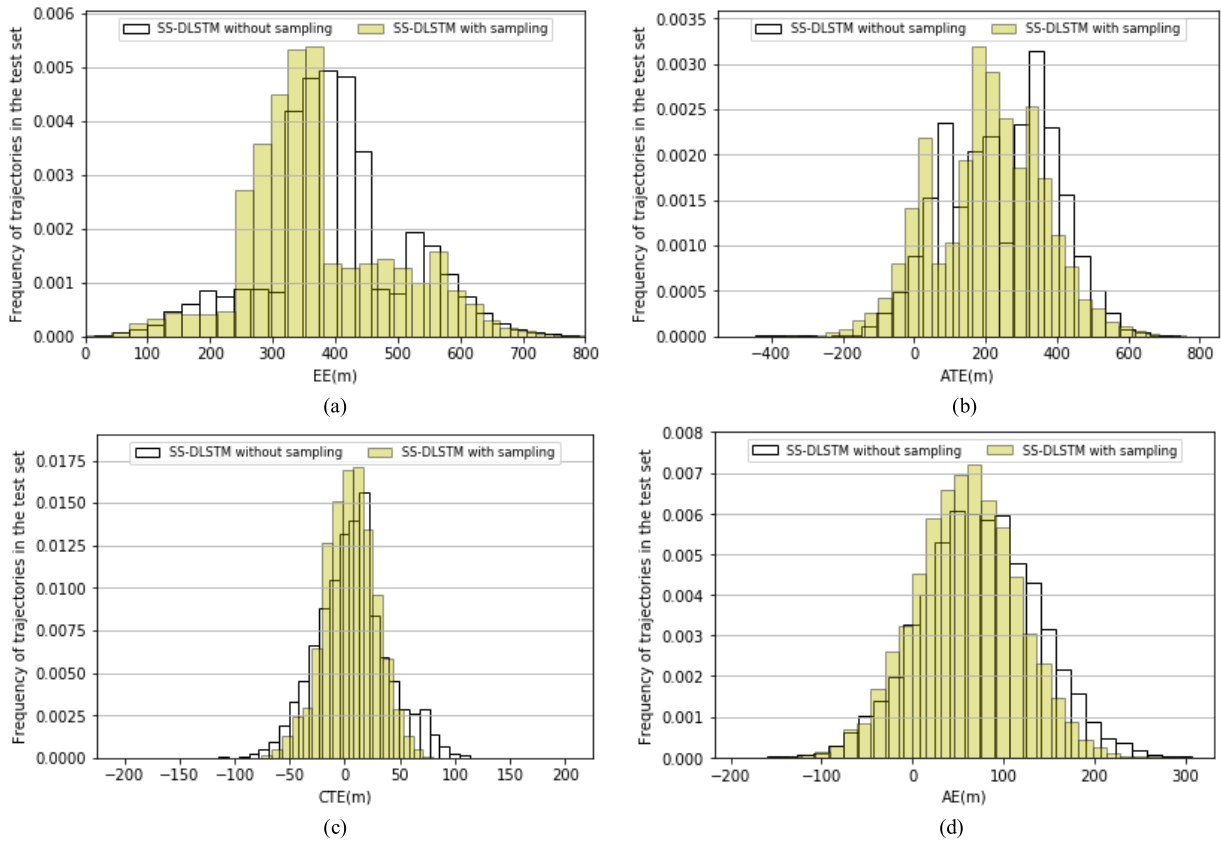


FIGURE 7. Histograms of EE, ATE, CTE and AE of the proposed SS-DLSTM with and without the sampling mechanism for landing aircraft.

standardization method is used to transform each attribute to a value in $[0, 1]$. Finally, the transformed trajectory dataset is randomly divided into a training set and a validation set.

Step 2: Mode parameter initialization: The model parameters consist of network structure parameters and solution method parameters. The network structure parameters include the length of the encoding sequence, the length of the decoding sequence, the number of network layers, and the number of neurons in each layer, where the length of the encoding sequence can be determined based on the applications, and other parameters are selected from a specified range. The solution method parameters include the sample size of each batch, the maximum number of epochs, the thresholds of the stopping criteria, and the initial learning rate, where the maximum number of epochs and the thresholds of the stopping criteria depend on the accuracy requirements, and the initial learning rate is a positive constant that is less than 1. In addition to these parameters, the model weights are initialized via random selection from the interval $[0, 1]$.

Step 3: Updating the weights on the training set: During each epoch, a specified number of trajectories are randomly selected from the training dataset, and these trajectories are divided into encoding and decoding trajectory sequences according to the lengths of the input sequence and the predicted sequence. The encoding sequence is used as the input

of the model, and the estimated output of the decoding process is obtained via a feed-forward propagation process. After that, the loss function is calculated by considering the output of the decoding sequence. If the loss function is less than the predefined threshold or the maximum number of epochs has been reached, then proceed to step 4; otherwise, the weights are adjusted via the gradient descent method.

Step 4: Verification on the validation set: Input the validation set into the learned model on the training set. If the error between the estimated output and the actual output is within the expected range, then the optimal prediction model is obtained; otherwise, repeat steps 2-4 until the stopping criterion is satisfied.

V. EXPERIMENTAL RESULTS

This section will analyze the performance of SS-DLSTM for aircraft trajectory prediction. The basic information of the dataset that is used in the following experiments has been described in Section III A. The dataset has a total of 98,620 trajectories, which include 47,562 takeoffs and 51,058 landings, and the dataset is randomly divided into a training dataset, a validation dataset, and a test dataset with the ratio 7:2:1. We run SS-DLSTM on a high-performance computer with 16 Core i7 CPUs, 128 GB RAM, and

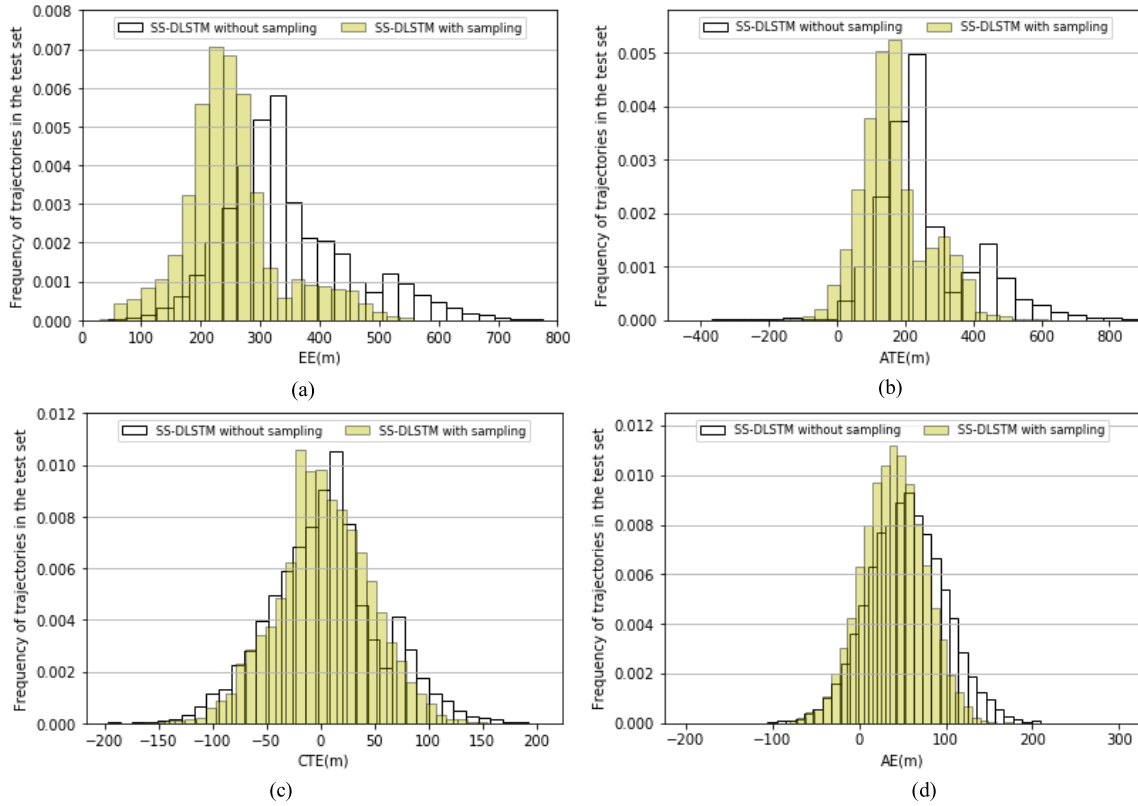


FIGURE 8. Histograms of EE, ATE, CTE and AE of the proposed SS-DLSTM with and without the sampling mechanism for takeoff aircraft.

NVIDIA P100 GPU under a Windows system using the TensorFlow package.

To evaluate the performance of the trajectory prediction model, we use the Euclidean error (EE), the along-track error (ATE), the cross-track error (CTE) and the altitude error (AE) as metrics, which have been widely adopted in the literature [6], [9], [44]. The EE measures the difference between the actual and predicted aircraft positions in 3D space. The ATE and CTE measure the parallel and perpendicular errors, respectively, of the course from the north. AE is the difference in the vertical positions between the actual and predicted trajectories. These metrics are calculated via the following formulas:

$$EE = \frac{1}{S} \sum_{t=1}^S \sqrt{(\bar{y}_1^t - y_1^t)^2 + (\bar{y}_2^t - y_2^t)^2 + (\bar{y}_3^t - y_3^t)^2} \quad (11)$$

$$ATE = \frac{1}{S} \sum_{t=1}^S \left[(\bar{y}_1^t - y_1^t) \sin(\theta_t^i) + (\bar{y}_2^t - y_2^t) \cos(\theta_t^i) \right] \quad (12)$$

$$CTE = \frac{1}{S} \sum_{t=1}^S \left[(\bar{y}_1^t - y_1^t) \cos(\theta_t) - (\bar{y}_2^t - y_2^t) \sin(\theta_t) \right] \quad (13)$$

$$AE = \frac{1}{S} \sqrt{\sum_{t=1}^S (\bar{y}_3^t - y_3^t)^2} \quad (14)$$

where S is the length of the predicted time; θ_t denotes the course from the north at timestamp t ; (y_1^t, y_2^t, y_3^t) and $(\bar{y}_1^t, \bar{y}_2^t, \bar{y}_3^t)$ denote the actual position and the estimated position at timestamp t , respectively.

A. STRUCTURE OF SS-DLSTM

This section will discuss the parameters of our proposed SS-DLSTM trajectory prediction model. SS-DLSTM contains two modules: an encoding module and a decoding module. These two modules have the same layers, and the output of each layer of the encoding module is the initial input of the corresponding layer of the decoding. The hyperparameters of our model can be divided into two categories: structure parameters and training parameters.

Regarding the structure parameters, it is necessary to determine the length of the encoding sequence, the length of the decoding sequence, the number of hidden layers, and the number of hidden units in each layer. The length of the decoding sequence depends on the application scenario. Compared with other airspaces, the flying time in the terminal airspace is relatively short, and the flying times of arrival flights are longer than those of departure flights within the terminal airspace; hence, we set the lengths of the decoding sequences for takeoff and landing aircraft to 90 seconds and 150 seconds, respectively. However, for the remaining three structure parameters, no universal selection principle is available; their values will be determined via many experiments.

This paper utilizes a mini-batch stochastic gradient descent method to solve the model. The parameters to be set include the batch size, the learning rate, the maximum number of epochs, and the threshold of stopping criterion. The learning

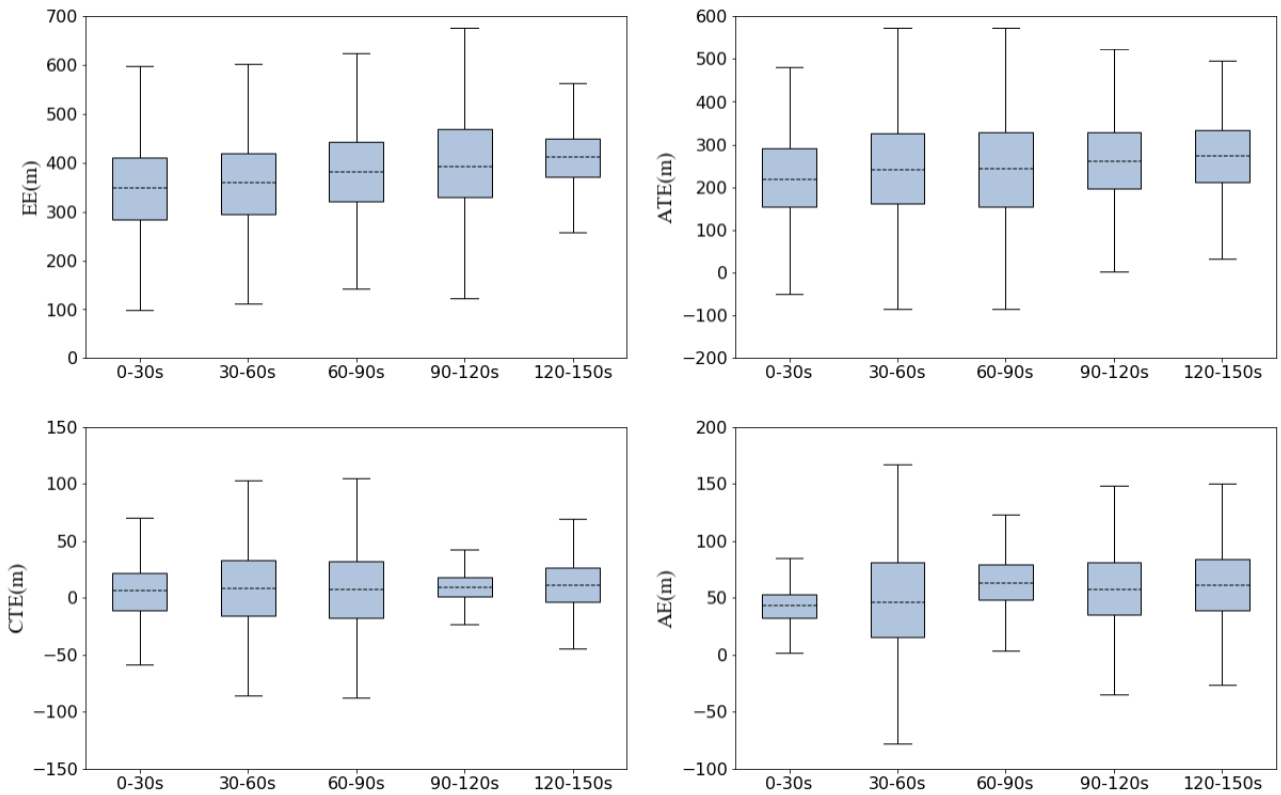


FIGURE 9. Boxplots of four metrics for landing aircraft in different time intervals.

rate affects the convergence of the model. If the learning rate is too low, the convergence will be very slow. However, the model may diverge as the number of iterations increases. The training error is large in the initial stage of iteration; hence, the learning rate typically must be set to a relatively large value. As the model gradually converges to the optimal solution, the learning rate should be progressively reduced. The initial learning rate of this paper is set to 0.006. In the training process, the learning rate will be reduced by half if the training error reduces too slowly. The maximum number of epochs and the threshold of the stopping criterion should be set according to the application. In this paper, the maximum number of iterations is set to 1000 for both takeoff and landing flights, and the threshold of the stopping criterion is set to 0.0001. The batch size of each epoch affects the convergence speed and the convergence accuracy, and its value depends on the predictive performance.

According to the above discussion, there is no universal guidance for the selection of the four parameters, namely, the length of the decoding sequence, the number of hidden layers, the number of neurons of each layer, and the batch size of each epoch, the values of which will be determined via many randomized trials. The possible selection ranges of these four parameters are as follows: The length of encoding sequence is in the range of [10, 60], the number of hidden layers is in the range of [2, 10], the number of neurons of each layer in the encoding and decoding modules is selected

from the set {50, 100, 150, 200, 250, 300, 350, 400}, and the batch size is selected from the set {300, 400, 500, 600, 700, 800, 900, 1000}. First, we randomly choose 1000 parameter settings from the possible ranges of each parameter. Then, we train the model based on those parameter settings. The optimal combinations for our model is as follows: (a) For takeoff aircraft, the length of encoding sequence is 35, the batch size of each epoch is 400, the number of hidden layers is 4, the numbers of neurons in the layers of the encoding module are [300, 250, 150, 100], and the numbers of neurons of the layers in the decoding module are [300, 250, 150, 100]. (b) For landing aircraft, the length of encoding sequence is 48, the size of each batch of samples is 600, the number of hidden layers is 6, the numbers of neurons of the layers in the encoding module are [350, 350, 200, 150, 100, 100], and the numbers of neurons of the layers in decoding module are [350, 350, 200, 150, 100, 100].

B. PERFORMANCE ANALYSIS

The input of our SS-DLSTM model for trajectory prediction includes basic attributes and derived attributes, and the validity of these two types of attributes will be evaluated below. Table 1 presents the performance of our SS-DLSTM with and without derived attributes. For takeoff aircraft, the accuracy of the model improved substantially when the derived attributes of the course were considered, with EE and ATE decreasing 126.4 m and 147.4 m, respectively. The effects of the derived

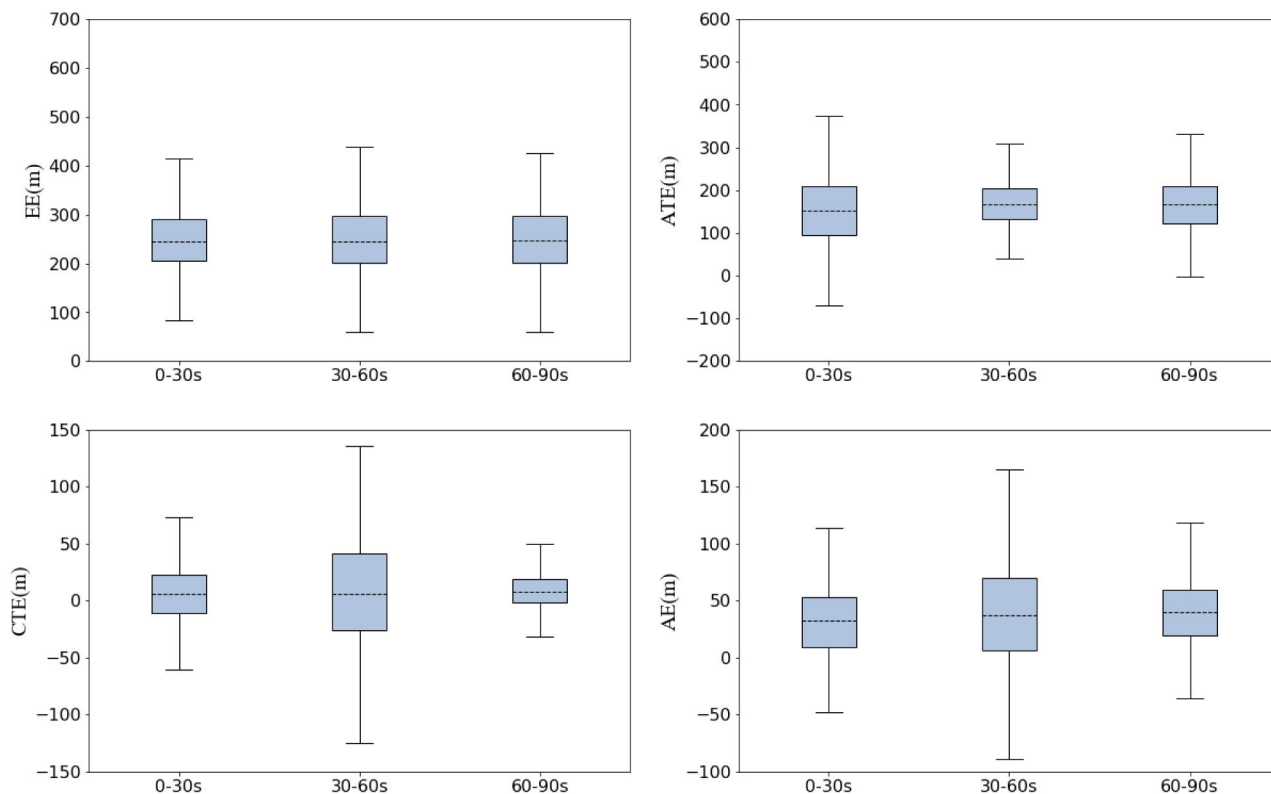


FIGURE 10. Boxplots of four metrics for takeoff aircraft in different time intervals.

TABLE 1. Performance of SS-DLSTM with different input attributes.

	Input attributes	Mean				Standard deviation			
		EE(m)	ATE(m)	CTE(m)	AE(m)	EE(m)	ATE(m)	CTE(m)	AE(m)
Landing aircraft	Basic attributes only	487.1	427.5	19.3	109.8	94.1	217.6	77.6	41.3
	Basic attributes and derived attributes of course	462.8	430.1	10.5	108.1	42.8	123.9	23.5	15.8
	Basic attributes and derived attributes of velocity	440.9	411.7	13.5	75.0	47.4	112.8	27.1	19.4
	Basic attributes and derived attributes of X	464.7	427.0	11.0	99.8	57.5	149.4	41.2	29.1
	Basic attributes and derived attributes of Y	383.0	352.5	9.8	95.2	69.8	156.2	33.9	26.4
	Basic attributes and derived attributes of Z	360.7	348.6	11.4	72.5	52.6	147.5	37.8	27.5
	Basic attributes and all derived attributes	330.1	251.8	6.4	45.3	36.9	87.4	14.9	8.2
Takeoff aircraft	Basic attributes only	363.1	359.3	21.0	98.1	81.2	187.3	55.2	37.1
	Basic attributes and derived attributes of course	236.7	211.9	12.2	96.4	39.8	95.6	24.8	11.6
	Basic attributes and derived attributes of velocity	259.0	213.5	15.2	63.3	41.0	99.5	22.4	15.8
	Basic attributes and derived attributes of X	338.8	288.8	12.7	88.1	59.1	128.4	31.6	29.4
	Basic attributes and derived attributes of Y	340.7	274.3	11.5	83.5	65.4	132.9	35.1	24.8
	Basic attributes and derived attributes of Z	316.9	300.4	13.1	60.8	53.3	121.3	32.6	22.1
	Basic attributes and all derived attributes	206.1	176.2	7.1	33.6	31.5	69.7	11.4	6.9

attributes of course on CET and AE are comparable to those of other derived attributes; however, the standard deviation of the prediction error is the smallest among all derived attributes. The derived attribute of velocity is only slightly outperformed by the derived attributes and shows a more substantial improvement than the three derived attributes of the position. For landing aircraft, the improvements of all

derived attributes in terms of all four evaluation metrics are comparable; however, the derived attributes of course and velocity still outperform the derived attributes of position in terms of the standard deviation.

From the above analysis, we conclude that our model with the derived attributes of velocity and course outperforms that with the derived attributes of position, which is due to the

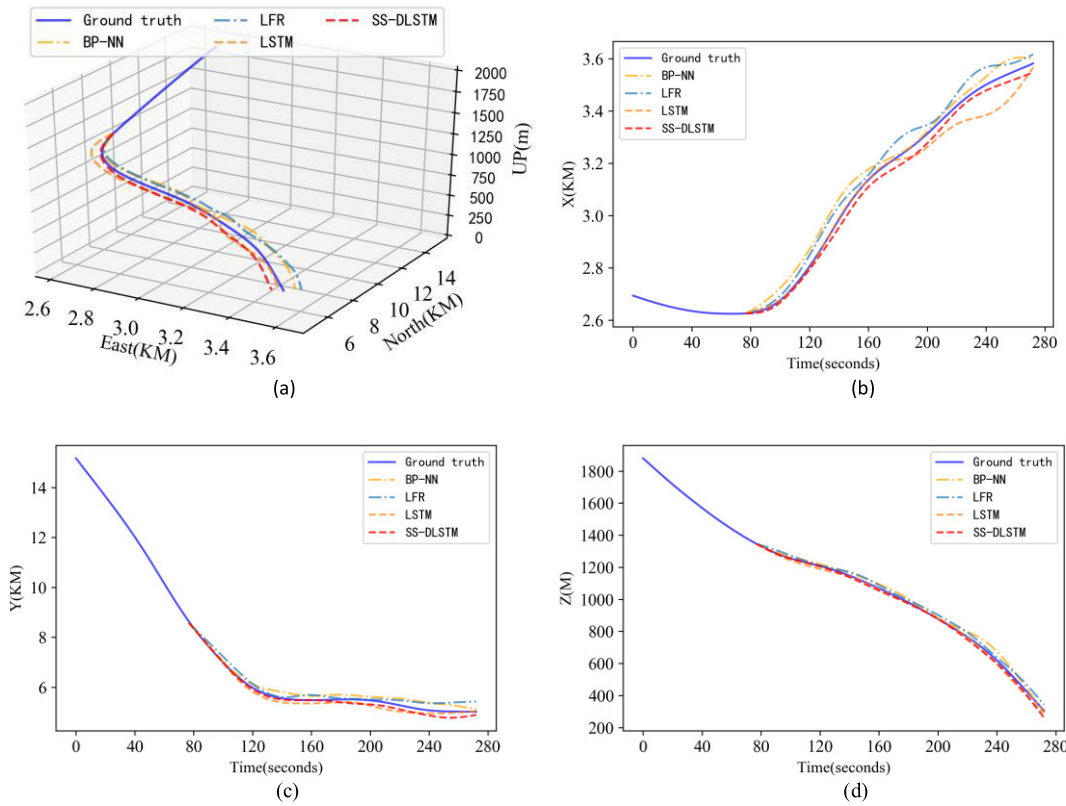


FIGURE 11. Visualization results of various methods for landing aircraft.

frequent changes of course and velocity of an aircraft in the terminal airspace.

By comparing the experimental results for takeoff and landing aircraft, the overall accuracy of prediction for takeoff aircraft is higher than that for landing aircraft. The main reasons are that takeoff aircraft stay in the terminal airspace for slightly less time than landing aircraft and exhibit relatively few trajectory patterns, while the trajectory patterns of landing aircraft are more diverse and random (e.g., an aircraft must hover in the air if no runway is available at the time). Overall, all derived attributes can improve the performance of the model, which is mainly because the trajectory of an aircraft is a sequence that is strongly correlated with time and space, and the derived attributes can directly reflect the change in the speed of the states of adjacent points in the trajectory sequence, which can further represent the time-varying characteristics of the trajectory. Our model yielded optimal results when all the derived attributes were considered; hence, the derived attributes had a positive effect on the performance of the model and can more accurately capture the motion state and time-varying characteristics of an aircraft.

The sampling mechanism that is used in the decoding module facilitates the improvement of the generalization performance of our model. The results of SS-DLSTM with and without the sampling mechanism for landing aircraft are

presented in Fig. 7, which plots histograms of four evaluation metrics. The histograms of EE for SS-DLSTM with the sampling mechanism are skewed to the left, and the histograms of ATE, CTE and AE are aggregated to the center, in contrast to our model without the sampling mechanism. It is concluded that the model with the sampling mechanism outperforms that without sampling mechanism in terms of generalization. Fig. 8 presents histograms of the four evaluation metrics of takeoff aircraft, and a conclusion that is similar to that for Fig. 7 is obtained. In addition, according to Fig. 7 and Fig. 8, the standard deviations of EE, ATE, and AE for landing aircraft are more significant than those for takeoff aircraft. In contrast, the standard deviation of AE is comparable to that for departing flights. Hence, our model performs better in predicting takeoff aircraft than in predicting landing aircraft, which agrees with the conclusion from the above experiment.

To evaluate the accuracy and robustness of the proposed method, we calculate the prediction errors of all trajectories in the validation dataset in each predicted time interval. Then, a boxplot is used to visualize the overall distribution. Fig. 9 presents boxplots of the four evaluation metrics of landing aircraft. According to the boxplot of EE, the error of most trajectories is within the range of 250 to 350 meters in the time interval of 0 to 30 seconds. The error tends to increase as the prediction time increases; however, the difference in EE among time intervals is not very large.

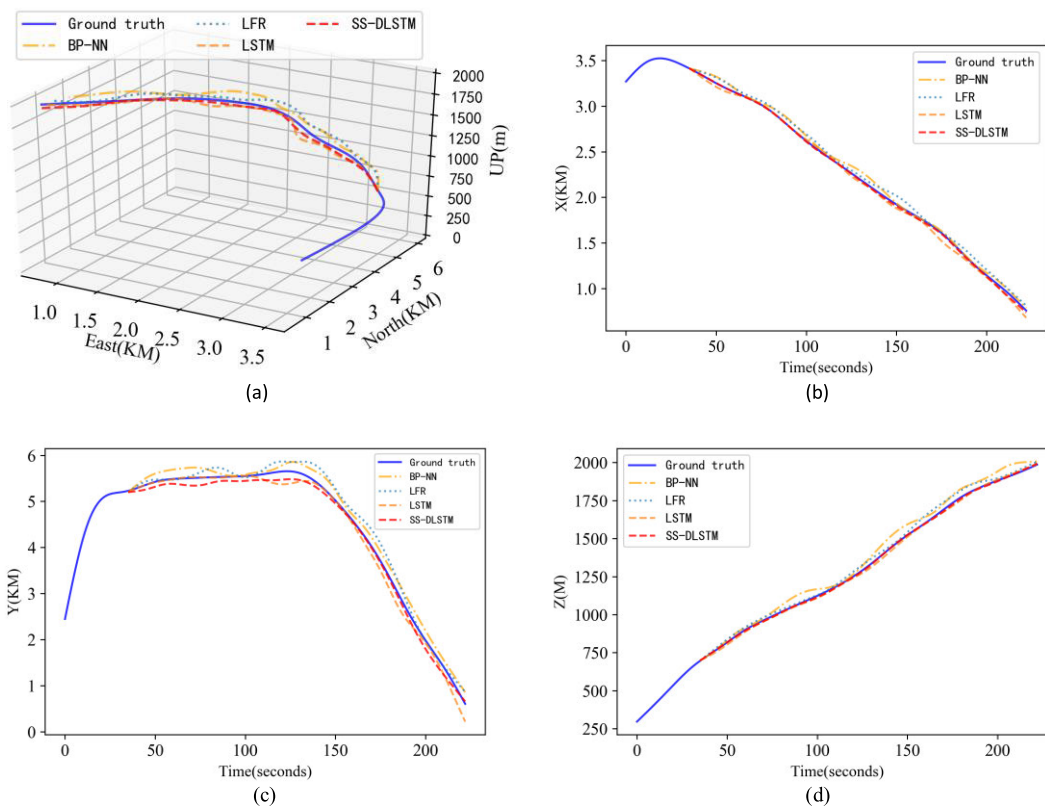


FIGURE 12. Visualization results of various methods for takeoff aircraft.

In addition, the EEs of a few trajectories are within the range of 500 to 700 meters, and it is found through analysis that these trajectories correspond to abnormal scenarios (such as aircraft hovering in the air). Furthermore, according to the boxplot of ATE, the value of ATE is positive for most trajectories and ranges from 200 to 400 meters; hence, the predicted trajectory of SS-DLSTM is ahead of the real trajectory in the direction of flight. Similarly, from the boxplot of AE, it is concluded that the predicted value for the altitude is higher than the real value. However, according to the boxplot of CTE, the values are mainly concentrated in the range of -50 to 50 meters symmetrically; hence, the prediction result has a similar probability of left-right deviation in the horizontal direction that is perpendicular to the flight direction. Furthermore, from the boxplots of ATE and AE, the errors in the horizontal position and altitude are relatively large in the time interval of 30 to 60 seconds. Via the analysis of real trajectories, it is found that the course and altitude of the aircraft vary substantially within this time interval.

Boxplots of the four evaluation indicators for takeoff aircraft are presented in Fig. 10. In terms of EE, the prediction error increases with the length of the predicted time. In addition, the values of EE, CTE, and AE in the time interval of 30 to 60 seconds are higher than those in the other two time intervals, as the altitude and course change more frequently in this time interval. Meanwhile, comparing Fig. 9 and Fig. 10,

the average error of the prediction model for landing aircraft is higher than that of the prediction model for takeoff aircraft overall, and the standard deviation of the error distribution for landing aircraft is larger than that for takeoff aircraft; hence, the model for predicting takeoff aircraft is more robust.

To more effectively evaluate the relative predictive performance of the proposed SS-DLSTM, we compare it with three baseline approaches for trajectory prediction: the back-propagation neural network (BP-NN) [12], local function regression (LFR) [13] and the long short-term memory neural network (LSTM-NN) [45]. Similar to our method, all three benchmarks are essentially data-driven methods that take into account only past information related to the trajectory and do not utilize any weather or aeronautical parameter information. All four approaches were trained by using the same training dataset and test dataset. The lengths of the predictive time intervals for landing and takeoff aircraft are 150 seconds and 90 seconds, respectively. To deeply analyze the performances of the methods, the experimental results will be divided into three time intervals for analysis in terms of predictive length. The quantitative results in Table 2 and Table 3 represent the performances of the methods on the validation dataset. Moreover, we choose two representative trajectories from the landing and takeoff validation datasets, and visualizations of the results of the methods are presented in Fig. 11 and Fig. 12, respectively. We plot only the

TABLE 2. Results of a quantitative comparison of various methods for landing aircraft in three time intervals.

Time interval	Methods	EE (m)	ATE (m)	CTE (m)	AE (m)
0-50 s	BP-NN	486.2	394.9	10.4	101.9
	LFR	392.3	345.4	8.4	96.4
	LSTM	349.5	318.2	9.6	69.9
	SS-DLSTM	289.7	253.1	5.4	42.7
50-100 s	BP-NN	522.9	430.1	13.2	123.2
	LFR	411.9	378.7	11.0	112.8
	LSTM	378.1	337.9	12.8	72.3
	SS-DLSTM	358.2	311.1	7.2	58.6
100-150 s	BP-NN	573.1	464.8	15.9	116.2
	LFR	410.4	429.7	14.7	102.1
	LSTM	402.6	357.8	14.1	88.1
	SS-DLSTM	351.5	319.2	11.9	57.2

TABLE 3. Results of a quantitative comparison of various methods for takeoff aircraft in three time intervals.

Time interval	Methods	EE (m)	ATE (m)	CTE (m)	AE (m)
0-30 s	BP-NN	201.5	141.2	12.7	52.4
	LFR	209.5	147.8	13.5	46.8
	LSTM	188.3	137.5	5.4	35.2
	SS-DLSTM	172.4	123.9	4.1	29.7
30-60 s	BP-NN	211.4	162.5	15.3	79.2
	LFR	217.9	175.8	14.8	67.9
	LSTM	203.1	142.3	7.9	41.9
	SS-DLSTM	191.2	128.4	6.5	38.7
60-90 s	BP-NN	219.7	173.2	11.6	62.8
	LFR	213.5	156.9	14.9	75.3
	LSTM	209.5	160.2	6.3	44.1
	SS-DLSTM	199.7	158.4	6.4	40.6

coordinate range that contains the whole trajectory to clearly show the difference between the predicted trajectory and the real trajectory.

Comparing the experimental results of these two tables, the predictive accuracy for takeoff aircraft is higher than that for landing aircraft for all methods. The main reason is that trajectory modes of arriving flights are more complex than those of departing flights in the terminal airspace that surrounds an airport. According to the quantitative results, BP-NN performs the poorest both for landing and takeoff aircraft among four methods. This is due to its network structure, as it belongs to a class of shallow neural networks that are unable to capture all flights modes, and it does not take into account the time-dependence of the track points. LFR outperforms BP-NN by using wavelet decomposition and local linear functional regression to capture the repetitiveness of the historical trajectories. The LFR model can more accurately characterize the landing and takeoff procedures of aircraft than BP-NN, but it is essentially a regression

method with only a few parameters; hence, its performance in capturing the intricate motion patterns of aircraft in the terminal airspace is limited. LSTM significantly outperforms BP-NN and LFR, as the network structure of LSTM uses a sequential memory cell to preserve the past information, which can more accurately characterize the time-dependence among tracks. SS-DLSTM outperforms LSTM in terms of accuracy; hence, our proposed method can more effectively learn the long-term dependencies of the aircraft trajectories and the repetitive patterns among trajectories of landing and takeoff aircraft.

VI. CONCLUSION

A data-driven aircraft trajectory prediction model is proposed for the terminal airspace that surrounds an airport, which is learned from many historical trajectories. The proposed method is composed of two main parts: trajectory reconstruction and a prediction model. For problems with unequal time intervals and noise in the original trajectories, a regularization method is proposed for reconstructing the trajectories with equal time intervals and smoothed noise. In this paper, trajectory prediction is regarded as mapping the historical state sequence to the future position sequence. To address this problem, an SS-DLSTM model is proposed, which consists of a deep encoding LSTM network and a deep decoding LSTM network. The deep encoding LSTM network uses historical aircraft states (longitude, latitude, altitude, course, velocity, and aircraft type) as input to generate long-term memory features and short-term memory features. The deep decoding LSTM network uses the output of the deep encoding LSTM network as input and generates a predicted trajectory sequence recursively. In the model training phase, the input that corresponds to each timestamp of the deep decoding LSTM network is randomly selected between the estimated output in the previous timestamp and the real output to improve the generalization performance of the model.

The proposed model is applied to datasets that were collected in the terminal airspace in Guangzhou, China. A total of 98,620 trajectories are considered in the experiments. The EE, ATE, CTE, and AE are used to evaluate the trajectory prediction errors. The experimental results demonstrated that our proposed model outperforms mainstream data-driven methods. In future research, we will explore the extension of the proposed method to a medium-term or long-term prediction by considering additional features such as the weather and the flight plan.

ACKNOWLEDGMENT

The authors thank Prof. Washington Yotto Ochieng from Imperial College London for helpful suggestions on the formulation of the model.

REFERENCES

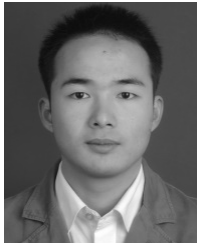
- [1] Airplanes, Boeing Commercial, "Statistical summary of commercial jet airplane accidents—worldwide operations| 1959–2017," Aviation Saf., Seattle, WA, USA, 2018.

- [2] G. Chatterji, "Short-term trajectory prediction methods," in *Proc. Guid., Navigat., Control Conf. Exhibit*, Aug. 1999, p. 4233.
- [3] G. Avanzini, "Frenet-based algorithm for trajectory prediction," *J. Guid., Control, Dyn.*, vol. 27, no. 1, pp. 127–135, Jan. 2004.
- [4] C. E. Seah and I. Hwang, "Terminal-area aircraft tracking using hybrid estimation," *J. Guid., Control, Dyn.*, vol. 32, no. 3, pp. 836–849, May 2009.
- [5] W. Liu and I. Hwang, "Probabilistic trajectory prediction and conflict detection for air traffic control," *J. Guid., Control, Dyn.*, vol. 34, no. 6, pp. 1779–1789, Nov. 2011.
- [6] S. Ayhan and H. Samet, "Aircraft trajectory prediction made easy with predictive analytics," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 21–30.
- [7] I. Lympopoulos, J. Lygeros, and A. Lecchini, "Model based aircraft trajectory prediction during takeoff," in *Proc. AIAA Guid., Navigat., Control Conf. Exhibit*, Keystone, O, USA, Aug. 2006, p. 6098.
- [8] W. Schuster, "Trajectory prediction for future air traffic management—complex manoeuvres and taxiing," *Aeronaut. J.*, vol. 119, no. 1212, pp. 121–143, 2015.
- [9] W. Schuster, M. Porretta, and W. Ochieng, "High-accuracy four-dimensional trajectory prediction for civil aircraft," *Aeronaut. J.*, vol. 116, no. 1175, pp. 45–66, Jan. 2012.
- [10] T. Baklacioglu and M. Cavcar, "Aero-propulsive modelling for climb and descent trajectory prediction of transport aircraft using genetic algorithms," *Aeronaut. J.*, vol. 118, no. 1199, pp. 65–79, Jan. 2014.
- [11] J. Zhang, J. Liu, R. Hu, and H. Zhu, "Online four dimensional trajectory prediction method based on aircraft intent updating," *Aerosp. Sci. Technol.*, vol. 77, pp. 774–787, Jun. 2018.
- [12] Y. Le Fablec and J. Alliot, "Using neural networks to predict aircraft trajectories," in *Proc. IC-AI*, 1999, pp. 524–529.
- [13] K. Tastambekov, S. Puechmorel, D. Delahaye, and C. Rabut, "Aircraft trajectory forecasting using local functional regression in sobolev space," *Transp. Res. C, Emerg. Technol.*, vol. 39, pp. 1–22, Feb. 2014.
- [14] Z. Wang, M. Liang, and D. Delahaye, "Short-term 4D trajectory prediction using machine learning methods," in *Proc. 7th SESAR Innov. Days*, Belgrade, Serbia, 2017, pp. 1–9.
- [15] A. M. Hernandez, E. J. Casado Magana, and A. G. Berna, "Data-driven aircraft trajectory predictions using ensemble meta-estimators," in *Proc. IEEE/AIAA 37th Digit. Avionics Syst. Conf. (DASC)*, Sep. 2018, pp. 1–10.
- [16] S. T. Barratt, M. J. Kochenderfer, and S. P. Boyd, "Learning probabilistic trajectory models of aircraft in terminal airspace from position data," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 9, pp. 3536–3545, Sep. 2019.
- [17] R. Alligier, D. Gianazza, and N. Durand, "Learning the aircraft mass and thrust to improve the ground-based trajectory prediction of climbing flights," *Transp. Res. C, Emerg. Technol.*, vol. 36, pp. 45–60, Nov. 2013.
- [18] R. Alligier and D. Gianazza, "Learning aircraft operational factors to improve aircraft climb prediction: A large scale multi-airport study," *Transp. Res. C, Emerg. Technol.*, vol. 96, pp. 72–95, Nov. 2018.
- [19] X. Guan, R. Lv, L. Sun, and Y. Liu, "A study of 4D trajectory prediction based on machine deep learning," in *Proc. 12th World Congr. Intell. Control Autom. (WCICA)*, Jun. 2016, pp. 24–27.
- [20] Z. Quan, Z. Gao, W. Zeng, X. Li, and M. Zhu, "On the duration, addressability, and capacity of memory-augmented recurrent neural networks," *IEEE Access*, vol. 6, pp. 12462–12472, 2018.
- [21] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [22] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Gated feedback recurrent neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 2067–2075.
- [23] X. Shi, Z. Chen, H. Wang, D. Yeung, W. Wong, and W. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 802–810.
- [24] F. Weninger, J. Geiger, M. Wöllmer, B. Schuller, and G. Rigoll, "Feature enhancement by deep LSTM networks for ASR in reverberant multi-source environments," *Comput. Speech Lang.*, vol. 28, no. 4, pp. 888–902, Jul. 2014.
- [25] I. V. Sutskever Oriol and V. Quoc, "Sequence to sequence learning with neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3104–3112.
- [26] X. Li, L. Peng, X. Yao, S. Cui, Y. Hu, C. You, and T. Chi, "Long short-term memory neural network for air pollutant concentration predictions: Method development and evaluation," *Environ. Pollut.*, vol. 231, pp. 997–1004, Dec. 2017.
- [27] I. Verma, R. Ahuja, H. Meisheri, and L. Dey, "Air pollutant severity prediction using bi-directional LSTM network," in *Proc. IEEE/WIC/ACM Int. Conf. Web Intell. (WI)*, Dec. 2018, pp. 651–654.
- [28] T. Fischer and C. Krauss, "Deep learning with long short-term memory networks for financial market predictions," *Eur. J. Oper. Res.*, vol. 270, no. 2, pp. 654–669, Oct. 2018.
- [29] J. Cao, Z. Li, and J. Li, "Financial time series forecasting model based on CEEMDAN and LSTM," *Phys. A, Stat. Mech. Appl.*, vol. 519, pp. 127–139, Apr. 2019.
- [30] J. Hu, C. Guo, B. Yang, and C. S. Jensen, "Stochastic weight completion for road networks using graph convolutional networks," in *Proc. IEEE 35th Int. Conf. Data Eng. (ICDE)*, Apr. 2019, pp. 1274–1285.
- [31] Y. Gao, J. M. Phillips, Y. Zheng, R. Min, P. T. Fletcher, and G. Gerig, "Fully convolutional structured LSTM networks for joint 4D medical image segmentation," in *Proc. IEEE 15th Int. Symp. Biomed. Imag. (ISBI)*, Apr. 2018, pp. 1104–1108.
- [32] C. Xu, J. Ji, and P. Liu, "The station-free sharing bike demand forecasting with a deep learning approach and large-scale datasets," *Transp. Res. C, Emerg. Technol.*, vol. 95, pp. 47–60, Oct. 2018.
- [33] Z. Zhao, W. Chen, X. Wu, P. C. Y. Chen, and J. Liu, "LSTM network: A deep learning approach for short-term traffic forecast," *IET Intell. Transp. Syst.*, vol. 11, no. 2, pp. 68–75, Mar. 2017.
- [34] B. Yang, S. Sun, J. Li, X. Lin, and Y. Tian, "Traffic flow prediction using LSTM with feature enhancement," *Neurocomputing*, vol. 332, pp. 320–327, Mar. 2019.
- [35] I. M. Kamal, H. Bae, N. I. Utama, and C. Yulim, "Data pixelization for predicting completion time of events," *Neurocomputing*, vol. 374, pp. 64–76, Jan. 2020.
- [36] W. Yu, R. Mittra, T. Su, Y. Liu, and X. Yang, *Parallel Finite-Difference Time-Domain Method*. Norwood, MA, USA: Artech House, 2006.
- [37] B. S. Ali, W. Y. Ochieng, and A. Majumdar, "ADS-B: Probabilistic safety assessment," *J. Navigat.*, vol. 70, no. 4, pp. 887–906, Jul. 2017.
- [38] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social LSTM: Human trajectory prediction in crowded spaces," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 961–971.
- [39] A. Tabassum, N. Allen, and W. Semke, "ADS-B message contents evaluation and breakdown of anomalies," in *Proc. IEEE/AIAA 36th Digit. Avionics Syst. Conf. (DASC)*, Sep. 2017, pp. 1–8.
- [40] A. Tabassum and W. Semke, "UAT ADS-B data anomalies and the effect of flight parameters on dropout occurrences," *Data*, vol. 3, no. 2, p. 19, 2018.
- [41] J. Zhang, W. Liu, and Y. Zhu, "Study of ADS-B data evaluation," *Chin. J. Aeronaut.*, vol. 24, no. 4, pp. 461–466, Aug. 2011.
- [42] S. Boyd, and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
- [43] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, "Scheduled sampling for sequence prediction with recurrent neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 1171–1179.
- [44] C. Gong and D. McNally, "A methodology for automated trajectory prediction analysis," in *Proc. AIAA Guid., Navigat., Control Conf. Exhibit*, Providence, RI, USA, Aug. 2004, p. 4788.
- [45] Z. Zhao, W. Zeng, Z. Quan, M. Chen, and Z. Yang, "Aircraft trajectory prediction using deep long short-term memory networks," in *Proc. CICTP*, Nanjing, China, Jul. 2019, pp. 124–135.



WEILI ZENG received the B.S. degree in applied mathematics from Jishou University, Jishou, China, in 2007, and the M.S. degree in applied mathematics and the Ph.D. degree in intelligent transportation systems from Southeast University, Nanjing, China, in 2009 and 2013, respectively.

From 2013 to 2015, he was a Postdoctoral Researcher with the School of Automation, Southeast University. He is currently an Associate Professor and a Master's Supervisor with the College of Civil Aviation, Nanjing University of Aeronautics and Astronautics, Nanjing. He has published more than 50 articles. His current research interests include artificial intelligence, air traffic management, air intelligent transportation systems, and traffic data mining.



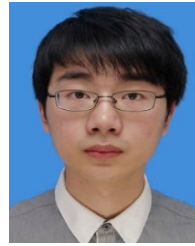
ZHIBIN QUAN (Member, IEEE) received the B.S. degree in computer science and technology from Jishou University, Jishou, China, in 2009, and the Ph.D. degree in computer software and theory from Southeast University, Nanjing, China, in 2018.

He is currently a Postdoctoral Fellow with the Department of Computer and Information Science, Faculty of Science and Technology, University of Macau, Taipa, Macau, China. He is also with the School of Automation, Southeast University. His current research interests include machine learning and natural language processing. He is a Reviewer of IEEE ACCESS, the IEEE/ACM TRANSACTIONS ON AUDIO SPEECH AND LANGUAGE PROCESSING, and the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS.



ZIYU ZHAO received the B.S. degree from the School of Transportation Engineering, Tongji University, Nanjing, China, in 2017. He is currently pursuing the M.S. degree with the College of Civil Aviation, Nanjing University of Aeronautics and Astronautics, Nanjing.

His research interests include machine learning and air intelligent transportation systems.



CHAO XIE received the B.S. degree in automation and the M.S. degree in control theory and control engineering from Nanjing Forestry University, Nanjing, China, in 2010 and 2013, respectively, and the Ph.D. degree in pattern recognition and intelligent systems from Southeast University, Nanjing, in 2018.

He is currently a Lecturer with the College of Mechanical and Electronic Engineering, Nanjing Forestry University. His current research interests include machine learning and data mining.



XIAOBO LU received the B.S. degree from the Department of Precision Instruments, Shanghai Jiao Tong University, Shanghai, China, the M.S. degree from the School of Automation, Southeast University, Nanjing, China, and the Ph.D. degree from the Department of Testing Engineering, Nanjing University of Aeronautics and Astronautics.

From 1998 to 2000, he was a Postdoctoral Researcher with the Chien-Shiung Wu Laboratory, Southeast University. He is currently a Professor with the School of Automation and the Deputy Director of the Detection Technology and Automation Research Institute, Southeast University. He is a coauthor of the book *An Introduction to the Intelligent Transportation Systems* (Beijing: China Communications Press, 2008). His research interests include image processing, signal processing, pattern recognition, and computer vision. He received several research awards, such as the first prize in Natural Science Award of the Ministry of Education of China and the prize in Science and Technology Award of Jiangsu province.

...