

## Article

# An Aircraft Trajectory Prediction Method Based on Trajectory Clustering and a Spatiotemporal Feature Network

You Wu , Hongyi Yu \*, Jianping Du , Bo Liu and Wanting Yu

Information System Engineering College, PLA Strategic Support Force Information Engineering University, Zhengzhou 450001, China

\* Correspondence: xxgcmayu@163.com

**Abstract:** The maneuvering characteristics and range of motion of real aircraft are highly uncertain, which significantly increases the difficulty of trajectory prediction. To solve the problem that high-speed maneuvers and excessive trajectories in airspace cause a decrease in prediction accuracy and to find out the laws of motion hidden in a large number of real trajectories, this paper proposes a deep learning algorithm based on trajectory clustering and spatiotemporal feature extraction, which aims to better describe the regularity of aircraft movement for higher prediction accuracy. First, the abnormal trajectories in the public dataset of automatic dependent surveillance–broadcast (ADS-B) were analyzed, and to ensure the uniform sampling of trajectory data, the cleaning and interpolation of the trajectory data were performed. Then, the Hausdorff distance was used to measure the similarity between the trajectories, K-Medoids was used for clustering, and the corresponding prediction model was established according to the clustering results. Finally, a trajectory spatiotemporal feature extraction network was constructed based on a convolutional neural network (CNN) and a bidirectional long short-term memory (BiLSTM) network, and a joint attention mechanism was used to obtain the important features of the trajectory points. A large number of actual trajectory prediction experiments showed that the proposed method is more accurate than existing algorithms based on BP, LSTM, and CNN–LSTM models.

**Keywords:** ADS-B; trajectory clustering; trajectory prediction



**Citation:** Wu, Y.; Yu, H.; Du, J.; Liu, B.; Yu, W. An Aircraft Trajectory Prediction Method Based on Trajectory Clustering and a Spatiotemporal Feature Network. *Electronics* **2022**, *11*, 3453. <https://doi.org/10.3390/electronics11213453>

Academic Editor: Carlos Tavares Calafate

Received: 26 September 2022

Accepted: 22 October 2022

Published: 25 October 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Accurate trajectory prediction technology plays a crucial role in many civilian and military fields, including aviation monitoring, radar warning, tracking, and navigation [1]. At present, air traffic is developing rapidly, and a single radar system is no longer sufficient for flight supervision. International Civil Aviation Organization (ICAO) uses flight trajectories as the sole reference for flight management in air traffic management (ATM) systems [2]. At the same time, the Next Generation Air Transport System (NextGen) proposed by the United States [3,4], and the Single European Sky ATM Study (SESAR) [5] presented by the European Control Center have identified flight trajectory optimization and trajectory conflict detection as essential methods to improve air traffic control efficiency, reduce delay rates, and ensure safe navigation. Therefore, it is crucial to study the accurate prediction of aircraft trajectories.

The trajectory prediction studied in this paper refers to the process of using the historical trajectory of a piece of aircraft to calculate the longitude and latitude of the aircraft in the future. At present, various aircraft trajectory prediction methods are proposed. Most of the conventional methods are based on pre-defined motion models, using the Markov chain [6], the Kalman filter [7], and other algorithms for trajectory prediction. However, the aircraft trajectory is complex and diverse and contains many influencing factors; thus, it is difficult for traditional modeling methods to construct a predictive model containing many factors. As machine learning and deep learning methods develop, it becomes feasible to use

a data-driven approach to learn the complex motion laws of aircraft to improve the accuracy of trajectory prediction, so trajectory tracking [8,9] and prediction algorithms based on deep neural networks have received increasing attention from scholars. Qian et al. [10] proposed an aircraft trajectory prediction model based on a backpropagation (BP) neural network. Using the strong nonlinear mapping ability of the BP network, the trajectory prediction accuracy was improved. Shi et al. [11] presented a prediction model based on a long short-term memory network (LSTM), which applied a sliding window to maintain the continuity of the data and the dynamic dependencies of adjacent points in the long trajectory. Graves and Siami-Namini et al. proved that the bidirectional long short-term memory (BiLSTM), which is an extended model of the LSTM, provides better performance than the LSTM [12,13]. Therefore, BiLSTM was applied in this study to improve the accuracy of the prediction model. Xu et al. [14] proposed the Social-LSTM trajectory prediction model, which considered the interaction between aircraft. In this model, the LSTM network and a pooling layer were established for each aircraft, and the results were better than those of the LSTM model. MA et al. [15] presented a trajectory prediction model of a convolutional neural network (CNN) combined with LSTM, using one-dimensional convolution to extract the spatial features between the trajectories, using LSTM to extract temporal features, and combining the two features to improve the trajectory prediction accuracy further. Zhou et al. [16] proposed a method to build a hybrid trajectory prediction model, which has predictive advantages for different prediction time spans. In order to achieve an accurate time series prediction, Zhang et al. [17] proposed a model based on an attention mechanism convolution neural network combined with LSTM.

All of the above methods use all the trajectory data of the target airspace to establish a prediction model and expect to learn the motion law of the aircraft. Although a higher trajectory prediction accuracy can be achieved, these methods do not take into account the influence of factors such as flight airspace and route differences on the prediction accuracy. Considering that there are problems such as overestimated trajectory predictions for the airspace, low prediction accuracy, and the inability to correlate the important features before and after the trajectory in reality, we realized that better prediction accuracy may be achieved by building a prediction model based on route classification, and routes may contain information on aircraft movement patterns. Based on a large number of real data, a trajectory prediction algorithm is proposed in this paper based on trajectory clustering and spatiotemporal feature extraction, and a joint attention mechanism (Attention). We first collected, analyzed, and cleaned the original ADS-B data, then used the K-Medoids method to cluster the trajectories hidden in the target airspace, and established different prediction models based on the clustering results, thereby effectively improving the accuracy of trajectory prediction. Finally, we separately used all categories of data for training, while the CNN mode was used to obtain the spatial characteristics of the trajectory, and the BiLSTM network was used to obtain the time characteristics before and after the trajectory point. The joint attention mechanism assigns different weights to the different features extracted by the model and obtains the final predicted trajectory. Experimentation with real datasets shows that the CNN–BiLSTM–Attention model used in this paper can achieve high accuracy of prediction, and prediction accuracy can be further improved after trajectory clustering.

The main contributions of this paper are as follows:

- (1) Based on a large number of real trajectory data, we used the method of trajectory clustering to find out the hidden route rules in the airspace and created different trajectory prediction models according to the clustering results to improve the prediction accuracy;
- (2) According to the characteristics of the aircraft trajectory, we designed the CNN–BiLSTM–Attention trajectory prediction model. The CNN model was used to extract the spatial features of each trajectory, the BiLSTM model was used to extract the temporal features, and the attention mechanism was used to learn the weights of different trajectory points;

- (3) We compared the constructed model with the BP, LSTM, and CNN–LSTM models and found that the proposed model greatly improved the prediction accuracy, and trajectory clustering can further improve the accuracy.

The remainder of this article is organized as follows: Section 2 presents the processing method for the ADS-B data. Section 3 presents the method of trajectory clustering and presents the clustering results. Section 4 presents the CNN–BiLSTM–Attention model used in this paper. Section 5 presents the comparative experimental results using the clustered ADS-B data. Section 6 presents the conclusions, which include the limitations and further work of this study.

## 2. Data Preparation

### 2.1. ADS-B Data

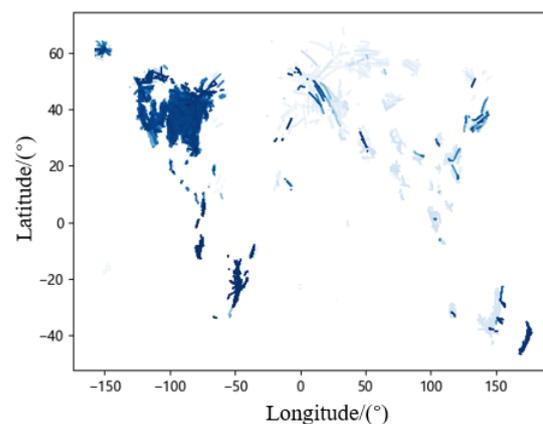
Based on the data of automatic dependent surveillance–broadcast (ADS-B), in this paper, we established a data-driven aircraft trajectory prediction model. The data used in this article were derived from the public dataset provided by the OpenSky website from January to May 2021, and the data structure is shown in Table 1. The actual ADS-B trajectory data have problems with repetition and missing trajectory points. In order to improve the model training effect, the ADS-B data should be cleaned and reconstructed first to build a high-quality dataset. The related work mainly includes trajectory selection in specific areas, outliers cleaning, trajectory segmentation, linear interpolation, data normalization, and sliding window data division.

**Table 1.** ADS-B data sample.

| Name      | Meaning  | Example      |
|-----------|--|--------------|
| Latitude  | The latitude of the aircraft   | 35.05178833  |
| Longitude | The longitude of the aircraft  | −90.35195487 |
| Velocity  | The speed over the ground of the aircraft                                  | 144.7865144  |
| Heading   | The direction of movement as the clockwise angle from the geographic north | 288.4349488  |

### 2.2. Feature Selection

The data provided by the OpenSky website are the global flight trajectory every hour. The one-hour data are provided as an example, shown in Figure 1. It can clearly be seen that the trajectory distribution is denser at latitude 20–40° and longitude −100–80°. This shows that there are frequent aircraft movements in this area, and there are a large number of diverse trajectories for model training. In deep learning, the data quality determines the effect of model training and training efficiency, so in this paper, we only used the ADS-B data of the above latitude and longitude.

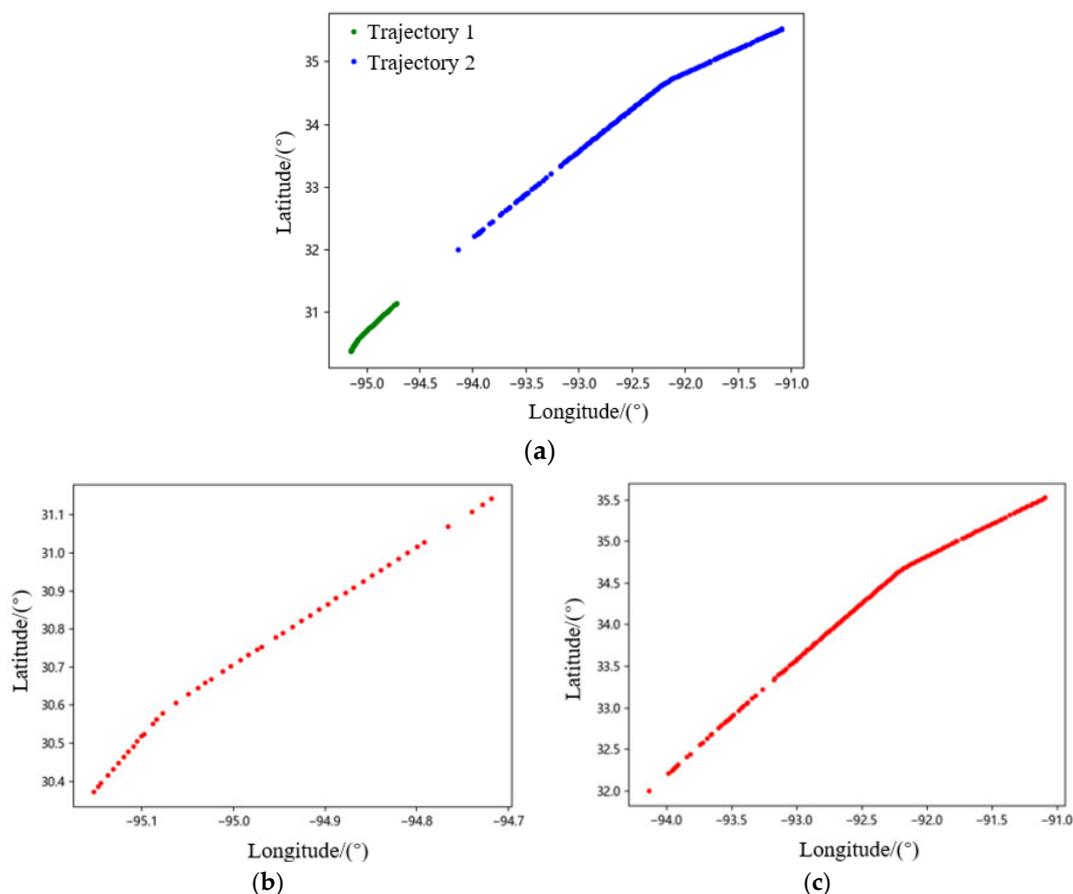


**Figure 1.** Global trajectory distribution.

The data provided by the OpenSky website are in chronological order. They contain flight data for various types of aircraft. We first compared the relationship between the aircraft type and the call sign provided by the website and filtered the data according to the latitude and longitude range to obtain the trajectory data with the aircraft type label. To highlight the trajectory characteristics of the aircraft in flight, we selected and used four parameters related to spatial information: longitude, latitude, velocity, and heading. Therefore, the aircraft trajectory data can be represented as  $T_{raj} = \{p_1, p_2, p_3, \dots, p_n\}$ , where  $n$  represents the total number of sampling points of the aircraft trajectory,  $p_t = \{lat_t, lon_t, velocity_t, heading_t\}$  refers to the sampling point of the aircraft trajectory at time  $t$ , where  $lat_t$ ,  $lon_t$ ,  $velocity_t$ , and  $heading_t$  respectively, represent the latitude, longitude, speed, and heading of the sampling point at time  $t$ .

### 2.3. Cleaning and Reconstruction

To improve model performance, it should be ensured that the available training data are correct. Firstly, the invalid values were excluded and filtered from the ADS-B data, including those points with the same longitude and latitude and the missing values between two adjacent points. Secondly, noisy trajectories were removed. The trajectory with less than 30 trajectory points was regarded as noise data and deleted. Finally, we used a distance-based trajectory reconstruction method. The threshold set in trajectory reconstruction was 30 km. If the distance between two adjacent points exceeded the threshold, it was marked as an intermittent point. If the number of data points of adjacent discontinuous points was greater than 30, the trajectory between the two intermittent points would be saved as a new one. The effect of trajectory cleaning and reconstruction is shown in Figure 2.



**Figure 2.** Trajectory reconstruction: (a) reconfigurable trajectory; (b) trajectory 1 after reconstruction; (c) trajectory 2 after reconstruction.

#### 2.4. Linear Interpolation

After the data cleaning and reconstruction in the above steps, the timestamps of two adjacent trajectory points may be discontinuous. Considering that the time interval between the adjacent data is short, the aircraft can be regarded as moving in a straight line at a uniform speed in a short period of time. Therefore, we used linear interpolation to complete the missing data. The reception interval of ADS-B data used in this paper is 10 s, which is set as  $\Delta t$ . The timestamp difference between two adjacent data is recorded as  $T$ , and the value of  $T/\Delta t$  represents the number of missing data between the adjacent points. The sampled data at time  $T_t$  are recorded as  $X_t$ ,  $T_{t+\Delta t}$  is the adjacent timestamp, and the sampled data here are  $X_{t+\Delta t}$ . As shown in Formula (1), the interpolated data at time  $T_{t+1}$  between the two timestamps can be calculated.

$$X_{t+1} = \frac{X_{t+\Delta t} + X_t}{2} \quad (1)$$

The trajectory after interpolation is shown in Figure 3. It can be seen that after completing the missing timestamp, the trajectory is more continuous, and the movement trend of the original trajectory is maintained and fully prepared for subsequent trajectory sample segmentation. At this time, the trajectory can be re-expressed as  $T'_{raj} = \{p'_1, p'_2, p'_3, \dots, p'_n\}$ , and the sampling points of the aircraft trajectory at time  $t$  can be expressed as  $p'_t = \{lat'_t, lon'_t, velocity'_t, heading'_t\}$ . After trajectory cleaning, reconstruction, and interpolation, a trajectory with continuous points and a suitable length for model training can be obtained. The above trajectory processing flow is shown in Algorithm 1.

---

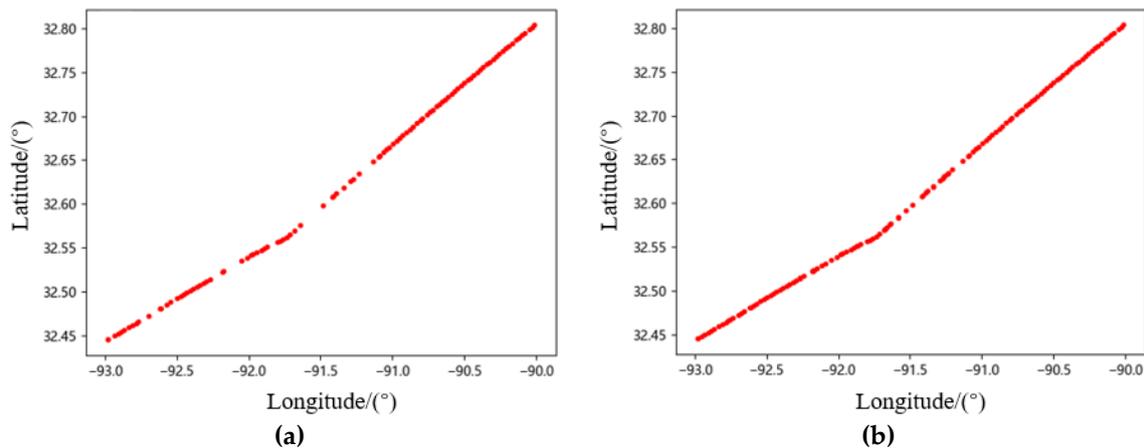
#### Algorithm 1: Trajectory Cleaning and Reconstruction

---

**Input:** ADS-B dataset A

**Output:** Cleaning and Reconstructing Trajectory Dataset F

- 1: Abnormal data are cleaned and filtered based on region from A;
  - 2: Trajectory C is filtered from A based on the aircraft call sign and stored in dataset D;
  - 3: For C in D:
  - 4:   if the number of trajectory points is greater than 30, then
  - 5:     if the distance between two adjacent points is more than 30 km, mark it as a breakpoint, then
  - 6:     if the number of adjacent breakpoint trajectory points is greater than 30, then
  - 7:       save the trajectory in dataset E;
  - 8:     else
  - 9:       back to step 5
  - 10:    else
  - 11:     save the trajectory in dataset E;
  - 12:    end
  - 13: else
  - 14:   delete the trajectory;
  - 15: end
  - 16: For C in E:
  - 17:   if the time between two adjacent points is not continuous, then
  - 18:     use linear interpolation to complete the trajectory and save it to dataset F;
  - 19: end.
-



**Figure 3.** Linear interpolation: (a) trajectory before interpolation; (b) trajectory after interpolation.

### 3. Trajectory Clustering

#### 3.1. K-Medoids Clustering

Clustering is an unsupervised machine learning method that is widely used in many fields [18]. We believe that in the same airspace, the trajectories on the same route are similar. In trajectory clustering, the aim is to classify flight trajectories with similar paths into the same category [19]. Comparing the similarity between different trajectories is a common clustering method. The similarity is measured by the distance between trajectories. Finally, clustering results are obtained by measuring the similarity between trajectories and merging the similar trajectory clusters. The trajectory length obtained after processing is different. This characteristic of trajectories makes them challenging to analyze through classical clustering methods. These trajectories are not just one point in space, but a collection of points, so Euclidean cannot be directly applied. The distance between trajectories is calculated using classical distance functions such as several miles or the half-sine distance. For this kind of problem, special functions such as the dynamic time warp [20], the symmetric Euclidean distance [21], the longest common subsequence [22], the Hausdorff distance [23], etc. are proposed to solve these problems.

In this paper, we used the K-Medoids clustering method for trajectory clustering. Compared with manual observation, it is more accurate to distinguish the route pattern from a large number of trajectories. Compared with the K-Means clustering method, K-Medoids clustering is less sensitive to outliers and has strong robustness [24]. Compared with the DBSCAN clustering method, K-Medoids clustering requires only one parameter adjustment to obtain a better clustering effect.

#### 3.2. Number of Clusters

In this paper, the Euclidean distance was used to calculate the distance between trajectory points, and the bidirectional Hausdorff distance was used to measure the similarity between trajectories. The bidirectional Hausdorff distance is defined as the maximum value of the minimum point-to-point distance between two trajectories A and B, followed by the maximum value, as shown in Formulas (2)–(4).

$$h(A, B) = \max_{a \in A} \min_{b \in B} \| a - b \| \quad (2)$$

$$h(B, A) = \max_{b \in B} \min_{a \in A} \| b - a \| \quad (3)$$

$$H(A, B) = \max[h(A, B), h(B, A)] \quad (4)$$

Assuming a dataset with  $n$  trajectories, the similarity was calculated using the bidirectional Hausdorff distance. A two-dimensional distance matrix of  $n * n$  can be obtained. For

the trajectory set  $E_t$ , the resulting distance matrix is shown in Formula (5), where  $d_{(E_{t1}, E_{t2})}$  is the similarity between the 1th trajectory and the nth trajectory.

$$\begin{bmatrix}
 d_{(E_{t1}, E_{t1})} & \cdots & d_{(E_{t1}, E_{tj})} & \cdots & d_{(E_{t1}, E_{tm})} \\
 \vdots & \ddots & \vdots & \ddots & \vdots \\
 d_{(E_{tj}, E_{t1})} & & d_{(E_{tj}, E_{tj})} & & d_{(E_{tj}, E_{tm})} \\
 \vdots & \ddots & \vdots & \ddots & \vdots \\
 d_{(E_{tm}, E_{t1})} & \cdots & d_{(E_{tm}, E_{tj})} & \cdots & d_{(E_{tm}, E_{tm})}
 \end{bmatrix} \tag{5}$$

When using K-Medoids clustering, the main problem is selecting the number of expected clusters K. In this paper, the number of clusters was initially determined by using the elbow method; in this method, the SSE metric is used to calculate the loss value, as shown in Equation (6), where  $C_i$  represents the  $i_{th}$  cluster,  $p$  is the sample point in  $C_i$ , and  $m_i$  is the centroid of  $C_i$ . The clustering error variation tends to be stable when the number of clusters reaches the actual number of clusters, and the K value at this time is the expected value, as shown in Figure 4.

$$SSE = \sum_{i=1}^k \sum_{p \in C_i} |p - m_i|^2 \tag{6}$$

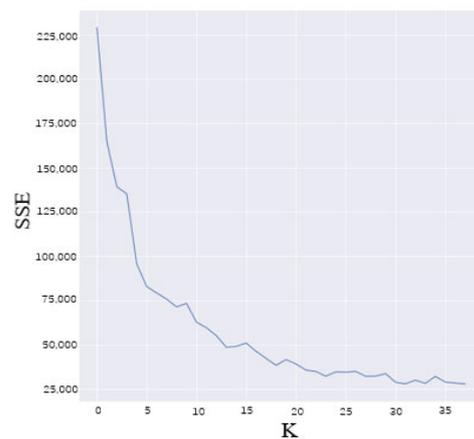
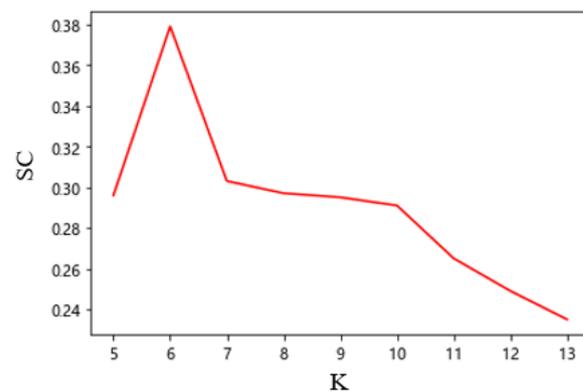


Figure 4. The determined values of K.

The clustering results are usually evaluated with the silhouette coefficient [25]. This method measures the quality of clustering from the perspective of intra-cluster compactness and inter-cluster separation and obtains the evaluation criteria for the pros and cons of clustering, as shown in Formula (7), where  $a(x)$  is the average distance between  $x$  and other trajectories in the cluster. The value of  $a(x)$  reflects the tightness between clusters to which  $x$  belongs; the smaller the value is, the more compact it is, and  $b(x)$  is the minimum of the average distance from  $x$  to all other clusters. The larger the value is, the better the separation is. The closer the silhouette coefficient is to 1, the more ideal the clustering effect is. In order to determine the value of K more precisely, Figure 5 shows the silhouette coefficients under different K values. As can be seen, when  $K = 6$ , the silhouette coefficient  $SC = 0.379$ , so  $K = 6$  was selected as the optimal number for clustering.

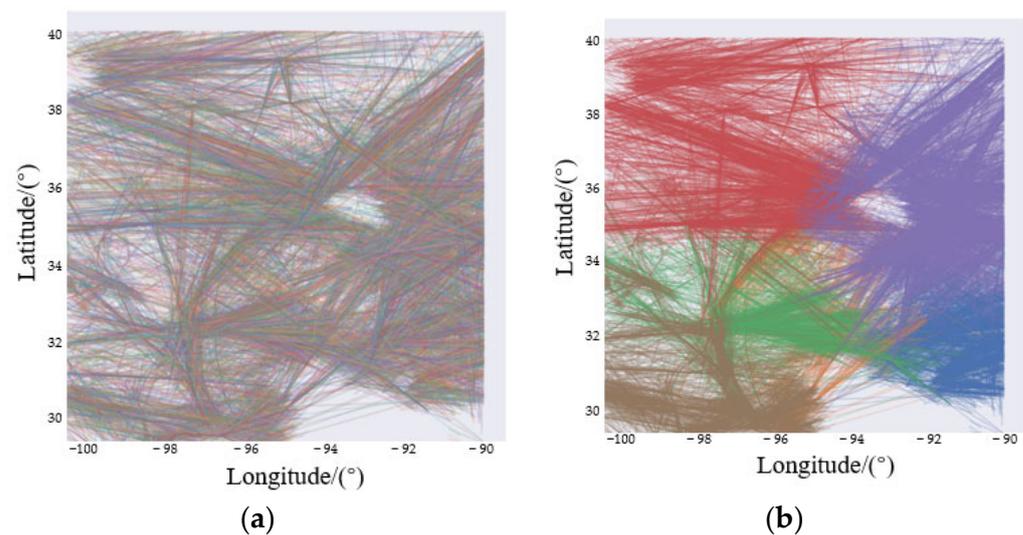
$$SC(x) = \frac{b(x) - a(x)}{\max\{a(x), b(x)\}} \tag{7}$$



**Figure 5.** Silhouette coefficients for different K values.

### 3.3. Clustering Result

Figure 6 shows the comparison of trajectories before and after clustering. It can be seen that, before clustering, various trajectories are intertwined, and the route distinction is not obvious. For example, in the entire airspace, the flight path of the aircraft is affected by the terrain or the purpose of the flight. It is challenging to build a prediction model in the entire area to obtain a good prediction effect. In this paper, through trajectory clustering, the trajectory of the entire airspace was divided into six routes, each of which represents a route. Trajectory clustering can discover the hidden rule of a trajectory in an airspace model, which cannot be found by the human eye. Making full use of this hidden rule will greatly help to improve the accuracy of trajectory prediction [26]. Figure 7 shows the distribution of various trajectories. It can be seen that through trajectory clustering, the routes entering the airspace from different directions can be distinguished, which reflects the specific flight purpose and achieves the expected effect.



**Figure 6.** Trajectory clustering: (a) trajectory before clustering; (b) trajectory after clustering.

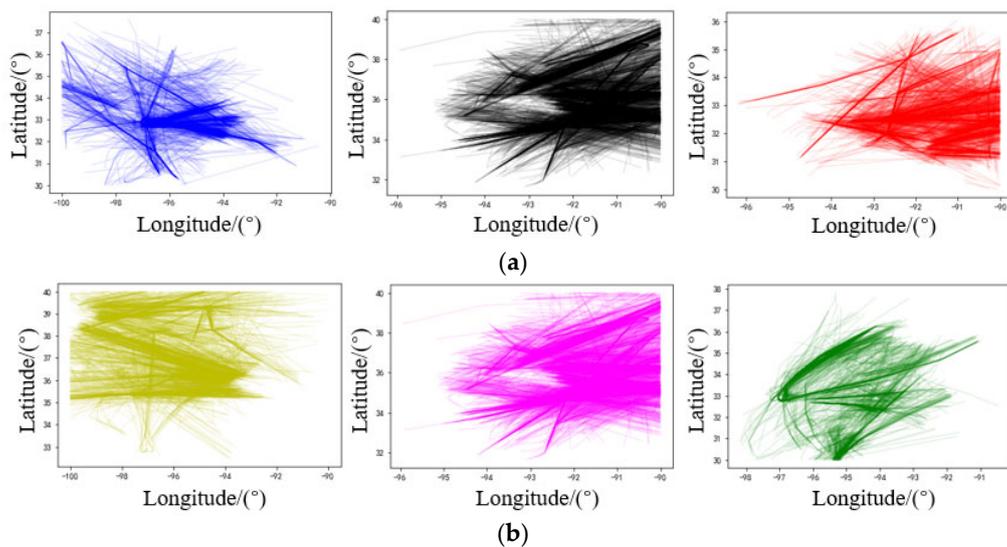


Figure 7. Trajectory distribution: (a) category 1–3; (b) category 4–6.

### 4. Trajectory Prediction Model

#### 4.1. CNN

The CNN model is shown in Figure 8. The model is composed of four parts: an input data matrix, a convolution layer, a pooling layer, and a fully connected layer. First, the processed data matrix is convoluted to extract its features, and then the downsampling and dimension reduction are performed through the pooling layer. The convolution and pooling operations can be repeated, and the extraction features become more abstract as the number of processing times increases. Finally, the extracted high-dimensional features are fed into the fully connected layer, and the final output is obtained through the activation function. Notably, 1DCNN can be well-applied to time series analysis, such as the ADS-B data. By setting the size of the convolution kernel and sliding the convolution on the time series, it can effectively extract the associated features in the time series, with strong local feature extraction advantages, and provide more effective data for subsequent models.

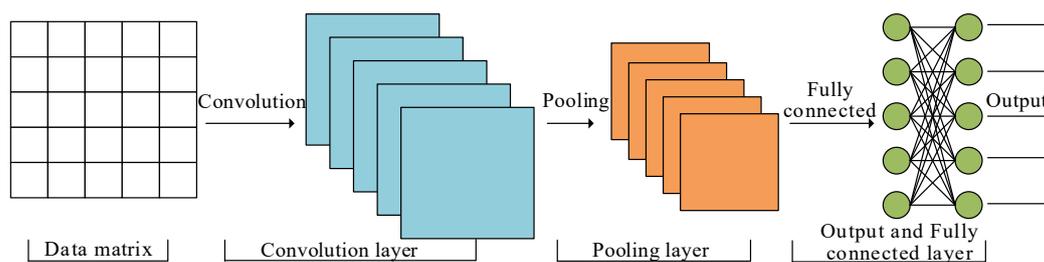


Figure 8. CNN model structure.

#### 4.2. BiLSTM

LSTM is equivalent to an improved recursive neural network (RNN) model that deals with time-series-related problems. It has three gate structures inside. The forget gate  $f_t$  is used to determine what information is discarded in cell states and how much information is discarded, the update gate  $i_t$  is used to determine what information is stored in the cell state  $c_t$ , and the output gate  $o_t$  is used to decide which information to output. The LSTM model solves the gradient loss and gradient explosion problems in the RNN mode. The LSTM unit structure is shown in Figure 9.

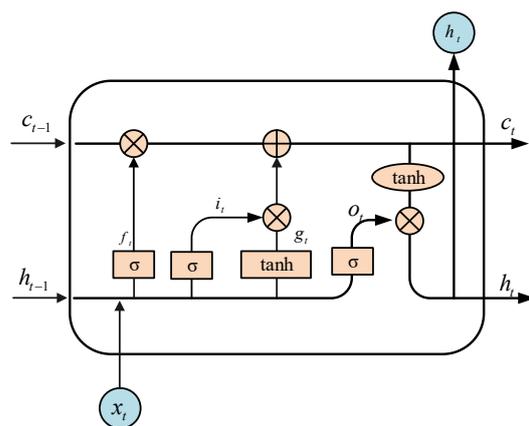


Figure 9. LSTM unit structure.

Figure 9 shows that the LSTM model adds the cell state  $c$  to maintain long-term memory between sequences. At each time step, the LSTM model simultaneously receives three inputs, namely,  $x$  for the current moment,  $c$  for the cell state, and  $h$  as the hidden layer state of the previous moment. The network update process is performed using Formulas (8)–(13), where  $W_{xi}$ ,  $W_{xf}$ ,  $W_{xg}$ , and  $W_{xo}$  are the coefficient matrices;  $b_i$ ,  $b_f$ ,  $b_g$ , and  $b_o$  are the bias matrices, and  $\sigma$  denotes a sigmoid activation function.

$$i_t = \sigma(\tilde{i}_t) = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \tag{8}$$

$$f_t = \sigma(\tilde{f}_t) = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \tag{9}$$

$$g_t = \tanh(\tilde{g}_t) = \tanh(W_{xg}x_t + W_{hg}h_{t-1} + b_g) \tag{10}$$

$$o_t = \sigma(\tilde{o}_t) = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \tag{11}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t \tag{12}$$

$$h_t = o_t \odot \tanh(c_t) \tag{13}$$

The BiLSTM model is an improvement over the traditional LSTM model. It includes forward iterative and backward iterative LSTM layers. In trajectory prediction, BiLSTM can solve the problem of forgetting the initial trajectory information caused by the long trajectory of the aircraft and can effectively extract the temporal features and retain long-term memory. Compared with LSTM with a one-way structure, BiLSTM uses both forward trajectory information and reverse trajectory information, which is helpful to improve the accuracy of the model prediction. The BiLSTM network structure is shown in Figure 10.

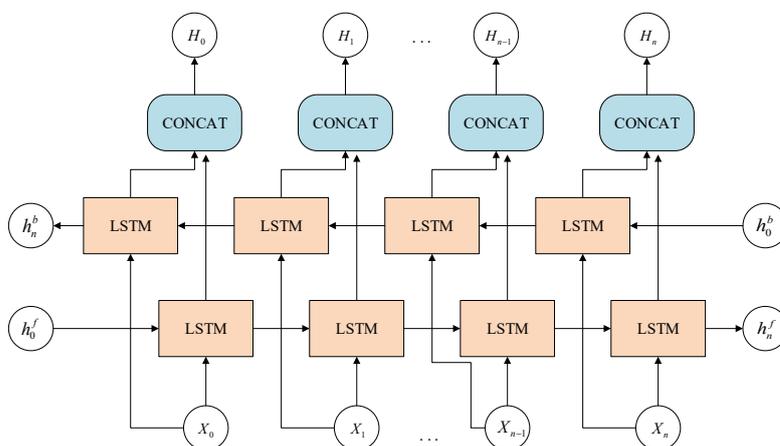


Figure 10. BiLSTM Model structure.

### 4.3. Attention

Recently, the attention mechanism [27] has been widely used in neural networks. Its working principle is shown in Figure 11. It was originally used for image recognition. The mechanism mimics the way humans observe objects and can extract critical features from a large amount of information and assign greater weight to the important features so that they can play more critical roles in the final output. The attention mechanism in the trajectory prediction model can enhance the influence of the critical features of input data, thereby improving the prediction effect of the model [28]. The calculation formulas are as provided below, where  $q$  is the state of the BiLSTM hidden layer,  $x_i$  represents the input data at the  $i$ th time, and  $a_i$  represents the calculated weight value.

$$a_i = \text{softmax}(s_i(x_i, q)) \tag{14}$$

$$\text{attention}(x, q) = \sum_{i=1}^n a_i x_i \tag{15}$$

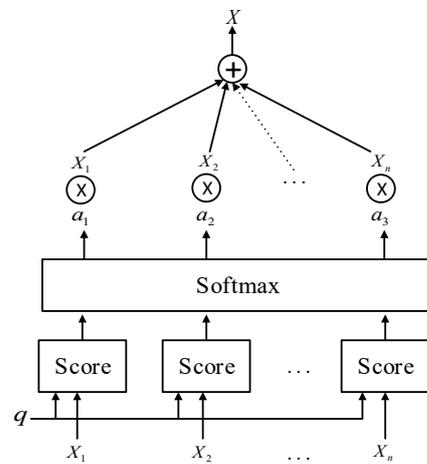


Figure 11. Attention model structure.

### 4.4. Model Framework

In this paper, we used the historical trajectory points of the aircraft as input to the model. The input can be expressed as (none, 10, 4); none represents the batch size of the input, 10 represents the input of 10 historical points of input, and 4 represents the feature dimension of each point, so the input is a data matrix of  $10 \times 4$ . The output can be expressed as (none, 1, 2), representing the point predicted by the model.

The core of the model is a one-dimensional CNN, BiLSTM, and Attention. First, the data are input into the CNN model, and the trajectory data first pass through a 1D convolution layer (Cov1d), where the activation function is ReLU, and then pass through a max pooling layer. The above convolution, activation, and pooling operations are repeated, and then the data go through a 1D average pooling layer; after the above process, the spatial features between the trajectories can be better extracted, and then the output of the CNN model is input to the BiLSTM model to extract the features with time correlation between trajectories. Next, the output of the trajectory that reflects the spatiotemporal information obtained in BiLSTM and the final hidden state  $h_n$  is input to the attention layer; with the dot product, the attention score is calculated and processed using SoftMax to obtain an attention weight. The final output is obtained after the weighted summation of the attention weight and the input of the model. The attention mechanism is to learn the importance of different trajectory points and to adjust the weight value of each trajectory point, thereby increasing the accuracy of trajectory prediction. The model framework in this article is shown in Figure 12.

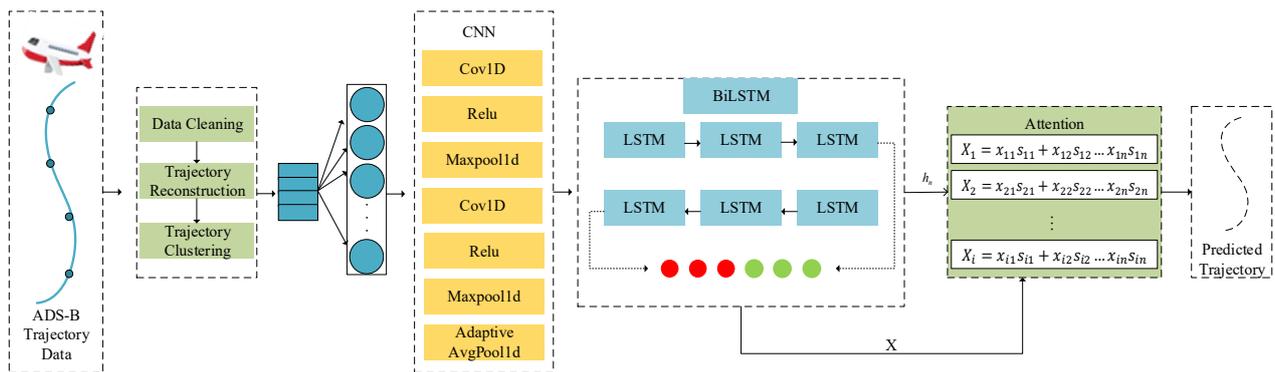


Figure 12. Model framework.

### 5. Experiment and Analysis

#### 5.1. Experiment Data

After performing the trajectory clustering procedure, the number of various trajectories was determined, which is shown in Table 2. In this paper, in order to ensure the effect of model training, we chose the fifth type of data in the clustering results to build dataset  $D_1$ , because the fifth type of data is the largest. In order to compare the influence of clustering on the prediction results, the data with the same data size as that of other clustering results were selected and defined as dataset  $D_2$ . The test set was 100 trajectories that were not involved in training in the fifth category of trajectories. At the same time, in order to compare the influence of classification model building on the prediction accuracy, we also used 1–6 types of data in the clustering results as the training data for model training and compared the prediction accuracy of the different models. The training set and the validation set were determined according to the ratio of 8:2 from the dataset. In order to prevent the model from over-fitting during training, the training set was randomly scrambled to increase the randomness of the samples. The deep learning framework for this experiment was PyTorch, with 150 iterations, a batch size of 256, and a learning rate set to 0.0001.

Table 2. Number of various trajectories.

| Category 1 | Category 2 | Category 3 | Category 4 | Category 5 | Category 6 |
|------------|------------|------------|------------|------------|------------|
| 3641       | 2903       | 3400       | 3639       | 5868       | 3323       |

#### 5.2. Data Normalization

After clustering, the trajectory features of each sampling point still contain four-dimensional data. In order to eliminate the difficulty of model training caused by different dimensions between different features, the method of Formula (16) is adopted to deal with the trajectory features and obtain the standardized trajectory.

$$x' = \frac{x - \mu(x)}{\sigma(x)} \tag{16}$$

In the above formula,  $x$  is the original trajectory information,  $\mu(x)$  and  $\sigma(x)$  represent the expectation and variance in the data, and  $x'$  is the trajectory information obtained after normalization.

#### 5.3. Sliding Time Window

The trajectory prediction belongs to supervised training, which needs to split the data into training samples and labels. In this paper, we used the sliding window method to segment the trajectory sequences, as shown in Figure 13. Here,  $p_1, p_2, p_3, \dots, p_n$  is the trajectory point, and the time window size is 10; the trajectory at time  $t$  can be expressed

as  $E_t = \{p_{t-9}, p_{t-8}, p_{t-7}, \dots, p_t\}$ , which indicates that each trajectory contains ten consecutive trajectory points, and the time sliding window is continuously moved forward for the segmentation operation until the last trajectory point is reached. The segmented trajectory can be denoted as  $E_{nTraj} = \{E_t, E_{t+1}, E_{t+2}, \dots, E_n\}$ , where  $t = 10, 11, 12, \dots, n$ . The experimental process is shown in Figure 14.

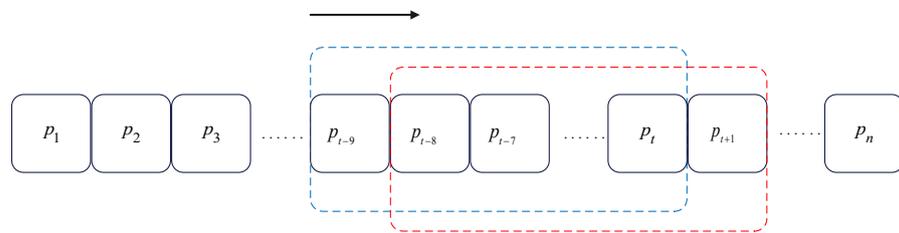


Figure 13. Sliding time window.

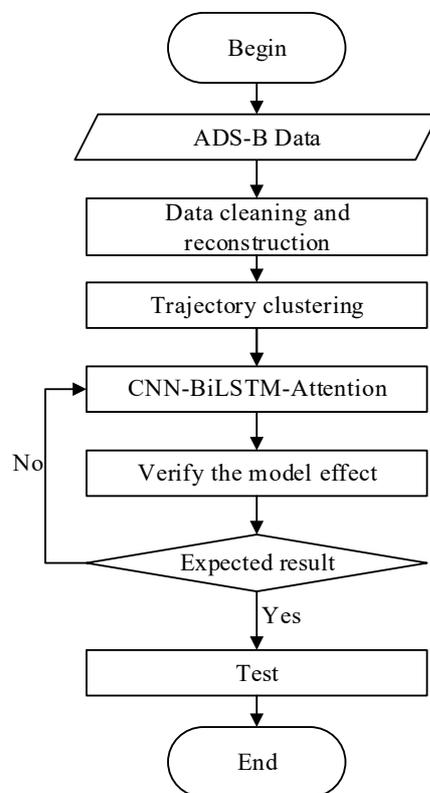


Figure 14. Experimental flowchart.

#### 5.4. Evaluation Indicators

We used the root mean square error (RMSE), the mean absolute error (MAE), and the mean absolute percentage error (MAPE) as evaluation indicators to measure the quality of trajectory prediction. At the same time, we used the mean deviation (MEDV) to demonstrate the model’s prediction accuracy more intuitively. The above four indicators were used to evaluate the effectiveness of the predictive model, and the calculation formulas are presented in Formulas (17)–(20).

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (A_i - E_i)^2} \tag{17}$$

$$MAE = \frac{1}{n} \sum_{i=1}^n |A_i - E_i| \tag{18}$$

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{A_i - E_i}{A_i} \right| \times 100\% \quad (19)$$

$$MEDV = \frac{1}{n} \sum_{i=1}^n distance(A_i, E_i) \quad (20)$$

where  $n$  is the number of samples,  $A_i$  is the predicted trajectory, and  $E_i$  is the actual trajectory; *distance* is used to calculate the geographical distance between two points. The smaller the value of these four evaluation indicators, the closer the prediction trajectory is to the actual trajectory, and the better the prediction effect.

### 5.5. Comparative Experiment Analysis

The CNN–BiLSTM–Attention model was compared with the BP, LSTM, and CNN–LSTM models to fully validate the performance of the CNN–BiLSTM–Attention model. The indicators for the evaluation of experimental results are shown in Table 3.

**Table 3.** Comparison of four models in evaluation index.

| Model                | Evaluation Indicators |               |               |               |               |               | MEDV/km      |
|----------------------|-----------------------|---------------|---------------|---------------|---------------|---------------|--------------|
|                      | Latitude              |               |               | Longitude     |               |               |              |
|                      | MAE                   | RMSE          | MAPE/%        | MAE           | RMSE          | MAPE/%        |              |
| BP                   | 0.0215                | 0.0245        | 0.0614        | 0.0139        | 0.0176        | 0.0151        | 1.562        |
| LSTM                 | 0.0096                | 0.0115        | 0.0271        | 0.0095        | 0.0121        | 0.0105        | 1.069        |
| BiLSTM–Attention     | 0.0086                | 0.0109        | 0.0243        | 0.0084        | 0.0114        | 0.0915        | 0.939        |
| CNN–LSTM             | 0.0077                | 0.0099        | 0.0218        | 0.0081        | 0.0114        | 0.0088        | 0.906        |
| CNN–BiLSTM–Attention | <b>0.0068</b>         | <b>0.0090</b> | <b>0.0192</b> | <b>0.0075</b> | <b>0.0106</b> | <b>0.0081</b> | <b>0.838</b> |

In this paper, the evaluation index RMSE was taken as an example. According to the experimental results, the performance of the other four prediction models was significantly improved compared with the BP model. Compared with the LSTM model, BiLSTM–Attention had 5.5% and 6.1% improvement in the latitude and longitude, respectively. This shows that the BiLSTM model and the attention mechanism are helpful for performance improvement. Compared with the BiLSTM–Attention model, CNN–LSTM had a 10.1% improvement in latitude, which shows that the CNN model can better extract the spatial information in the trajectory data. Compared with LSTM, BiLSTM–Attention, and CNN–LSTM models, the CNN–BiLSTM–Attention model used in this paper had a 27.7%, 17.4%, and 10% improvement in the latitude prediction accuracy and 14.1% and 7.5% improvement in the longitude prediction accuracy. This shows that making full use of the relationship between adjacent trajectory points can better improve prediction accuracy. Taking one test trajectory as an example, Figure 15 shows the prediction results of each model on the trajectory. Figure 16 shows the error distribution of the trajectory prediction of the four models. It can be seen from the figure that the above models can predict the overall trend of the trajectory, but the model in this paper can better fit the actual trajectory when there is a sudden change in the trajectory, and the average distance deviation is small, which satisfies the accuracy requirement of trajectory prediction. Additionally, the presented results showed that the use of the attention mechanism can focus on the important information in the trajectory and can obtain more accurate prediction results.

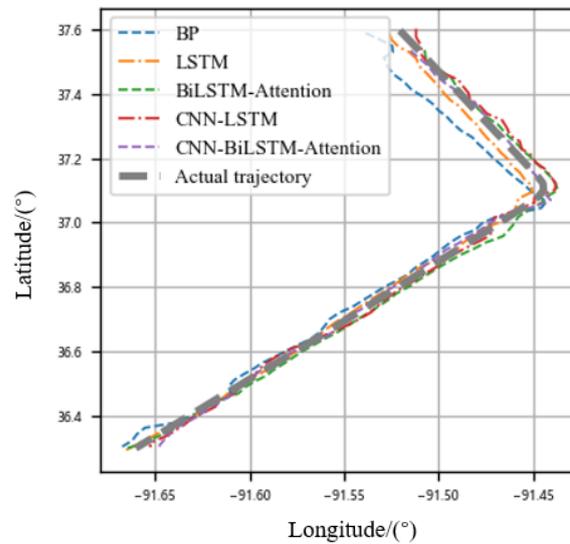


Figure 15. Comparison of prediction results of five models.

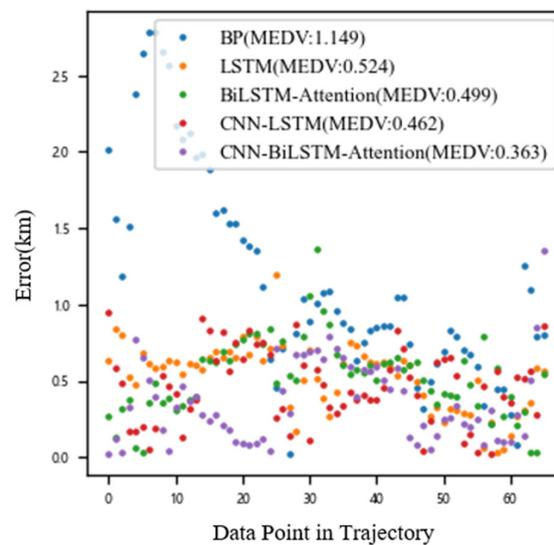
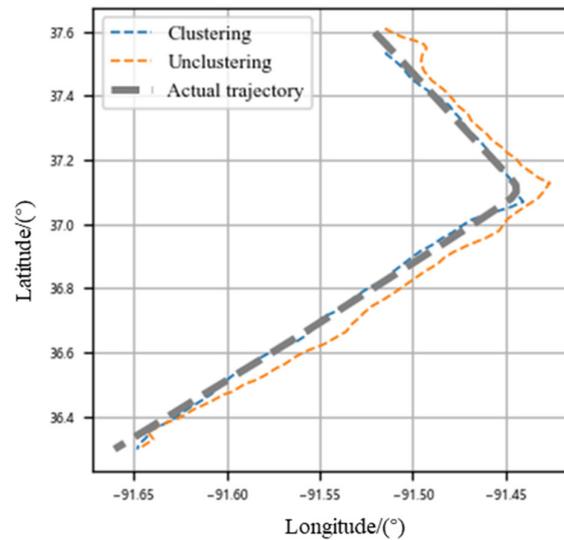


Figure 16. Error scatter plot of five models.

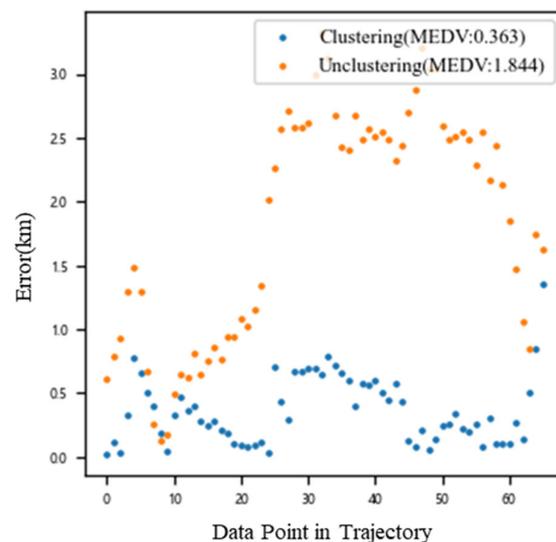
In order to verify the impact of trajectory clustering on the prediction results, we used  $D_1$  and  $D_2$  datasets to train the CNN–BiLSTM–Attention model and compare the performance differences. The experimental evaluation indicators are shown in Table 4, and Figure 17 shows the comparison of the prediction effects of the trajectory. Figure 18 shows the prediction error distributions for the two models, and it can be seen that the prediction error distribution of the model trained with the clustered data was more stable and more practical. The experimental evaluation indicators are shown in Table 4. The model trained with the clustered data was better than the model trained with the unclustered data in terms of evaluation index and average distance error. Still taking the RMSE evaluation index as an example, the prediction accuracy of the latitude and longitude increased by 103% and 110%, respectively, and the improvement effect was obvious. Figure 16 shows the comparison of the effects of the two types of models in trajectory prediction. It can be seen that the clustering of the trajectories is of great help in obtaining more accurate prediction results.

**Table 4.** Comparison of clustering and unclustering data in evaluation indicators.

| Model                | Dataset | Evaluation Indicators |               |               |               |               |               | MEDV/km      |
|----------------------|---------|-----------------------|---------------|---------------|---------------|---------------|---------------|--------------|
|                      |         | Latitude              |               |               | Longitude     |               |               |              |
|                      |         | MAE                   | RMSE          | MAPE/%        | MAE           | RMSE          | MAPE/%        |              |
| CNN-BiLSTM-Attention | $D_2$   | 0.0158                | 0.0183        | 0.0443        | 0.0187        | 0.0223        | 0.0203        | 2.088        |
|                      | $D_1$   | <b>0.0068</b>         | <b>0.0090</b> | <b>0.0192</b> | <b>0.0075</b> | <b>0.0106</b> | <b>0.0081</b> | <b>0.838</b> |



**Figure 17.** Comparison of the influence of clustering on prediction results.



**Figure 18.** Error scatter plot of two models.

In this paper, we classified six types of trajectories in airspace via clustering, the above results verified that under the same number of training data, a more targeted training model through clustering could achieve higher prediction accuracy for trajectory prediction. In order to further verify the help of trajectory clustering to improve the prediction accuracy of the model, we used six types of trajectory data to train different prediction models for testing. When performing the trajectory prediction process, first, the distance was determined between the trajectory to be predicted in the airspace and each cluster center; then, the category to which the trajectory belonged was determined based on the distance,

and then the corresponding prediction model was selected for trajectory prediction. In this paper, a test trajectory was selected from the sixth type of data in the clustering results to display the prediction effect, as shown in Figure 19. The evaluation index MEDV was taken as an example to compare the effects. As shown in Figure 20, compared with the unclustered model, the prediction accuracy of the model trained with the sixth type of data improved by 32%. Compared with the model trained with the other five types of data, the prediction error of the trajectory was the lowest, which proves the effectiveness of the training model established based on the determined categories through trajectory clustering.

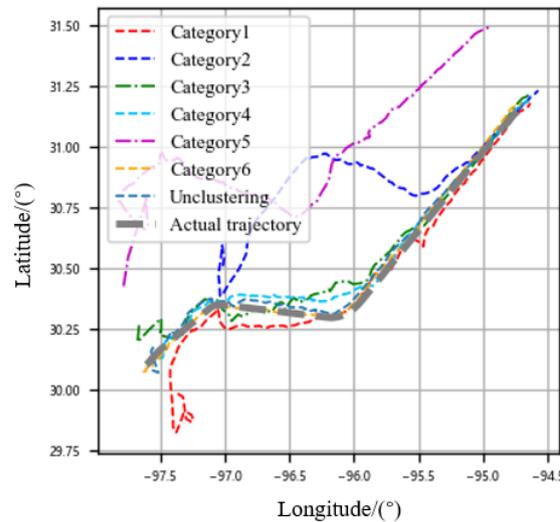


Figure 19. Comparison of the influence of classification training on prediction results.

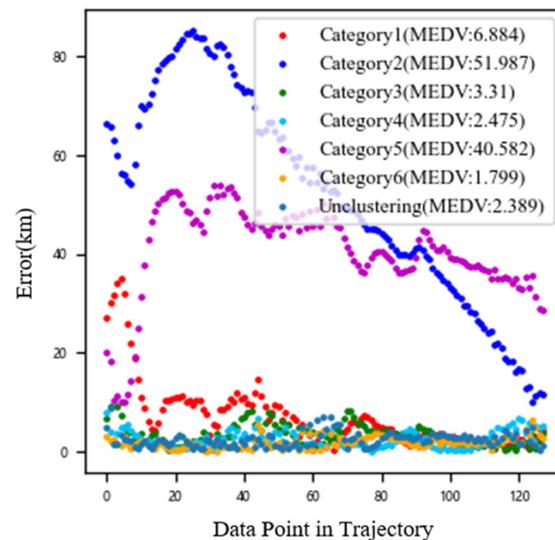


Figure 20. Error scatter plot of seven models.

### 6. Conclusions

In this paper, we proposed a trajectory prediction model based on the CNN–BiLSTM–Attention model to solve the problems of complex and changeable aerial trajectories, low trajectory prediction accuracy, and largely predicted airspace. Through data cleaning and reconstruction, and the data processing methods of trajectory clustering, the aircraft trajectory was predicted, and MAE, RMSE, MAPE, and MEDV were used as evaluation indexes to compare the BP, LSTM, and CNN–LSTM models through experiments, and the prediction effects of models trained with the clustering data and non-clustering data were compared. The experimental results showed that the accuracy of this model was high,

and the trajectory clustering could significantly improve the trajectory prediction accuracy. However, there are still some shortcomings: (1) The model can only predict the trajectory of the next step and cannot track the trajectory for a long time. (2) The trajectory predictions at cluster junctions are not resolved. (3) The trajectory of aircraft is also affected by many other factors, such as weather and air traffic control orders. The ADS-B data used in this paper contain limited information, and the influence of these factors was not considered in the model training. In the next step, further research can be carried out on the above issues.

**Author Contributions:** Conceptualization, Y.W.; data curation, B.L.; formal analysis, Y.W. and H.Y.; methodology, W.Y.; project administration, J.D. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** The data used in this paper can be obtained by contacting the corresponding author.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Cheng, C.; Guo, L.; Wu, T.; Sun, J.; Gui, G.; Adebisi, B.; Gacanin, H.; Sari, H. Machine-Learning-Aided Trajectory Prediction and Conflict Detection for Internet of Aerial Vehicles. *IEEE Internet Things J.* **2022**, *9*, 5882–5894. [[CrossRef](#)]
2. Slattery, R.; Zhao, Y. Trajectory Synthesis for Air Traffic Automation. *J. Guid. Control. Dyn.* **1997**, *20*, 232–238. [[CrossRef](#)]
3. Harrison, M. ADS-X the Next Gen Approach for the Next Generation Air Transportation System. In Proceedings of the 25th Digital Avionics Systems Conference, Portland, OR, USA, 15–19 October 2006; pp. 1–8.
4. Luckenbaugh, G.; Landriau, S.; Dehn, J.; Rudolph, S. Service Oriented Architecture for the Next Generation Air Transportation System. In Proceedings of the 2007 Integrated Communications, Navigation and Surveillance Conference, Herndon, VA, USA, 30 April–3 May 2007; pp. 1–9.
5. Sipe, A.; Moore, J. Air traffic functions in the NextGen and SESAR airspace. In Proceedings of the IEEE/AIAA Digital Avionics Systems Conference, Orlando, FL, USA, 23–29 October 2009; pp. 2.A.6-1–2.A.6-7.
6. Wang, X.; Jiang, X.; Chen, L.; Wu, Y. KVLMM: A Trajectory Prediction Method Based on a Variable-Order Markov Model With Kernel Smoothing. *IEEE Access* **2018**, *6*, 25200–25208. [[CrossRef](#)]
7. Chiou, Y.S.; Wang, C.L.; Yeh, S.C. Reduced-complexity scheme using alpha–beta filtering for location tracking. *IET Commun.* **2011**, *5*, 1806–1813. [[CrossRef](#)]
8. Yu, W.; Yu, H.; Du, J.; Zhang, M.; Liu, J. DeepGTT: A general trajectory tracking deep learning algorithm based on dynamic law learning. *IET Radar Sonar Navig.* **2021**, *15*, 1125–1150. [[CrossRef](#)]
9. Gao, C.; Yan, J.; Zhou, S.; Varshney, P.K.; Liu, H. Long short-term memory-based deep recurrent neural networks for target tracking. *Inf. Sci.* **2019**, *502*, 279–296. [[CrossRef](#)]
10. Kui, Q.; Ying, Z.; Liujing, Y.; Rongping, X.; Xidian, H. Aircraft target track prediction model based on BP neural network. *Command. Inf. Syst. Technol.* **2017**, *8*, 54–58. [[CrossRef](#)]
11. Shi, Z.; Xu, M.; Pan, Q.; Yan, B.; Zhang, H. LSTM-based Flight Trajectory Prediction. In Proceedings of the 2018 International Joint Conference on Neural Networks, IJCNN, Rio de Janeiro, Brazil, 8–13 July 2018; pp. 1–8.
12. Graves, A.; Schmidhuber, J. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Netw.* **2005**, *18*, 602–610. [[CrossRef](#)] [[PubMed](#)]
13. Siami-Namini, S.; Tavakoli, N.; Namin, A.S. The Performance of LSTM and BiLSTM in Forecasting Time Series. In Proceedings of the 2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, 9–12 December 2019; pp. 3285–3292.
14. Xu, Z.; Zeng, W.; Chu, X.; Cao, P. Multi-Aircraft Trajectory Collaborative Prediction Based on Social Long Short-Term Memory Network. *Aerospace* **2021**, *8*, 115. [[CrossRef](#)]
15. Ma, L.; Tian, S. A Hybrid CNN-LSTM Model for Aircraft 4D Trajectory Prediction. *IEEE Access* **2020**, *8*, 134668–134680. [[CrossRef](#)]
16. Zhou, J.; Zhang, H.; Lyu, W.; Wan, J.; Zhang, J.; Song, W. Hybrid 4-Dimensional Trajectory Prediction Model, Based on the Reconstruction of Prediction Time Span for Aircraft en Route. *Sustainability* **2022**, *14*, 3862. [[CrossRef](#)]
17. Zhang, Z.; Gao, J.; Mao, J.; Liu, Y.; Anguelov, D.; Li, C. STINet: Spatio-Temporal-Interactive Network for Pedestrian Detection and Trajectory Prediction. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, Seattle, WA, USA, 13–19 June 2020; pp. 11343–11352.
18. Park, J.; Jeong, J.; Park, Y. Ship Trajectory Prediction Based on Bi-LSTM Using Spectral-Clustered AIS Data. *J. Mar. Sci. Eng.* **2021**, *9*, 1037. [[CrossRef](#)]
19. Sun, J.; Ellerbroek, J.; Hoekstra, J. Flight Extraction and Phase Identification for Large Automatic Dependent Surveillance–Broadcast Datasets. *J. Aerosp. Inf. Syst.* **2017**, *14*, 566–572. [[CrossRef](#)]
20. Vaughan, N.; Gabrys, B. Comparing and Combining Time Series Trajectories Using Dynamic Time Warping. *Procedia Comput. Sci.* **2016**, *96*, 465–474. [[CrossRef](#)]

21. Chen, Z.; Yan, Y.; Ellis, T. Lane detection by trajectory clustering in urban environments. In Proceedings of the IEEE International Conference on Intelligent Transportation Systems, Qingdao, China, 8–11 October 2014; pp. 3076–3081.
22. Bergroth, L.; Hakonen, H.; Raita, T. A survey of longest common subsequence algorithms. In Proceedings of the International Symposium on String Processing & Information Retrieval, A Coruna, Spain, 27–29 September 2000; pp. 39–48.
23. Laxhammar, R.; Falkman, G. Online Learning and Sequential Anomaly Detection in Trajectories. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *36*, 1158–1173. [[CrossRef](#)] [[PubMed](#)]
24. Arora, P.; Deepali; Varshney, S. Analysis of K-Means and K-Medoids Algorithm for Big Data. *Procedia Comput. Sci.* **2016**, *78*, 507–512. [[CrossRef](#)]
25. Aranganayagi, S.; Thangavel, K. Clustering Categorical Data Using Silhouette Coefficient as a Relocating Measure. In Proceedings of the International Conference on Computational Intelligence and Multimedia Applications, ICCIMA 2007, Sivakasi, India, 13–15 December 2007; pp. 13–17.
26. Liu, L.; Zhai, L.; Han, Y. Aircraft trajectory prediction based on convLSTM. *Comput. Eng. Des.* **2022**, *43*, 1127–1133. [[CrossRef](#)]
27. Mnih, V.; Heess, N.; Graves, A.; Kavukcuoglu, K. Recurrent Models of Visual Attention. *Adv. Neural Inf. Process. Syst.* **2014**, *3*. [[CrossRef](#)]
28. Zhang, S.; Wang, L.; Zhu, M.; Chen, S.; Zhang, H.; Zeng, Z. A Bi-directional LSTM Ship Trajectory Prediction Method based on Attention Mechanism. In Proceedings of the 2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference, IAEAC, Chongqing, China, 12–14 March 2021; pp. 1987–1993.