Transactions on Computer Science and Technology
Volume 10, 2023, PP. 1-10

A Flight Trajectory Prediction Method Based on Internal Relationships between Attributes

Liwei Wu, Yuqi Fan

School of Computer Science and Information Engineering/Hefei University of Technology, Hefei, Anhui, 230000, China

Email: utcs142857@tom.com

Abstract

The rapid development of the aviation industry urgently requires airspace traffic management, and flight trajectory prediction is a core component of airspace traffic management. Flight trajectory is a multidimensional time series with rich spatio-temporal characteristics, and existing flight trajectory prediction methods only target the trajectory point temporal relationships, but not the implicit interrelationships among the trajectory point attributes. In this paper, a graph convolutional network (AR-GCN) based on the intra-attribute relationships is proposed for solving the flight track prediction problem. First, the network extracts the temporal features of each attribute and fuses them with the original features of the attribute to obtain the enhanced attribute features, then extracts the implicit relationships between attributes as inter-attribute relationship features. Secondly, the enhanced attribute features are used as nodes and the inter-attribute relationship features are used as edges to construct the inter-attribute relationship graph. Finally, the graph convolutional network is used to aggregate the attribute features. Based on the full fusion of the above features, we achieved high accuracy prediction of the trajectory. In this paper, experiments are conducted on ADS-B historical track data. We compare our method with the classical method and the proposed method. Experimental results show that our method achieves significant improvement in prediction accuracy.

Keywords: Deep Learning; Graph Convolution Neural Network; Flight Trajectory Prediction

1 INTRODUCTION

In recent years, with the rapid growth of the aviation industry, air traffic monitoring has become a common focus of society. Therefore, the United States, the European Union, and China have proposed a plan to jointly build a next-generation air traffic management (ATM) system. The air traffic management system is a dynamic, complex, information-driven automated system[1]. The main objective of the program is to make full use of the available airspace and routes to ensure safe and efficient flights[2]. Trajectory prediction (TP) is a core component of ATM[3], so the trajectory prediction problem is an important issue for air traffic management.

The flight's trajectory points contain several attributes, such as time, longitude, latitude, altitude, and airspeed. The purpose of trajectory prediction is to predict the values of each attribute of the aircraft's future trajectory points. The current trajectory prediction methods mainly include trajectory prediction based on aerodynamic and aircraft performance models[4,5,12], trajectory prediction based on traditional time series prediction methods[9,10,24], and trajectory prediction based on machine learning[8,16,17,18]. Among them, the trajectory prediction methods based on aerodynamic and aircraft performance models require a large amount of accurate a priori knowledge, and the trajectory prediction methods based on traditional time-series prediction methods lack the ability to capture the complex nonlinear features of flight data.

Machine learning-based methods for trajectory prediction have become a hot research topic. The literature[17] based on Long Short-Term Memory (LSTM) network for trajectory, but LSTM only focuses on capturing the features of the trajectory points in the time dimension. In the literature[17], a CNN (Convolutional Neural Network) and LSTM based trajectory method was proposed to extract features in the spatial and temporal dimensions of the trajectory by CNN and LSTM, respectively, but failed to fully explore the implied interrelationships between the attributes of the trajectory points. In the literature[8], the K-means clustering algorithm is used to cluster the trajectory points with similar features into one flight state, and the corresponding GRU (Gated Recurrent Unit) prediction model is trained for each flight state, which integrates all the attributes contained in the trajectory points when calculating the distance values from the clustering centre. The existing work on deep learning-based trajectory prediction has explored a lot in terms of mining the temporal relationships of the trajectory points. However, the implicit interrelationships between the attributes of the trajectory points are ignored. Flight trajectories contain multiple attributes, so the trajectory problem is a multivariate time-series prediction problem[9]. In this paper, an Inter-Attribute Relationships based Graph Convolutional Networks (AR-GCN) is proposed for solving the trajectory prediction problem. Our contributions are summarized as follows:

(1) AR-GCN fuses attribute features and inter-attribute relationship features. Firstly, the augmented attribute features are obtained by extracting the attribute timing features and fusing the original attribute features to obtain the augmented attribute features, and the implied relationships between attributes are extracted as the inter-attribute relationship features. Secondly, the inter-attribute relationship graph is constructed by using the augmented attribute features as nodes and the inter-attribute relationship features as edges. Finally, the attribute features are aggregated by a graph convolutional network, and the attribute features are input to a multilayer perceptron (MLP) to achieve flight trajectory prediction.

(2) We conduct single-step prediction and multi-step prediction experiments on real flight data, respectively. The experimental results show that the model proposed in this paper effectively improves the prediction accuracy of the trajectory and reduces the root mean square error and the mean absolute error of the prediction.

2 RELATED WORKS

Currently, the prediction methods can be divided into three categories based on the differences in the trajectory prediction methods: trajectory prediction based on aerodynamic and aircraft performance models, trajectory prediction based on traditional time-series prediction methods, and trajectory prediction based on machine learning.

The first approach bases the trajectory prediction on the a priori knowledge provided by aerodynamics and flight models. For example, the literature[4] proposed a four-dimensional trajectory prediction method based on the basic flight model to construct horizontal profile, altitude profile and velocity profile models of the aircraft for trajectory prediction according to the characteristics of different flight phases. The literature[5] proposed a new four-dimensional trajectory prediction model based on the statistical analysis of the actual radar trajectory data of the aircraft, combined with the aircraft intention model and the aircraft dynamics kinematic model. The literature[12] proposed an algorithm to dynamically adjust the flight model weights based on the observed trajectory data to improve the accuracy of trajectory prediction. The aerodynamic-based approach and flight performance model require a large number of accurate prior assumptions and a priori knowledge, which leads to a model that cannot adapt to the complex and variable airspace environment, with high computational complexity and low prediction accuracy.

The second approach is the traditional time series prediction methods including Kalman filter algorithm, autoregressive integrated moving average method, particle filter, etc., which treats the trajectory prediction problem as an estimation problem of a stochastic linear mixed system. A new UAV attitude estimation algorithm is proposed in the literature[6]. The algorithm uses attitude quaternions to represent the attitude of the UAV and achieves the UAV attitude estimation by the Extended Kalman Filter (EKF). The literature[7] proposed an interactive multimodal prediction algorithm for vehicle motion diversity and uncertainty based on hybrid systems theory. The literature[13] proposed a hybrid state estimation method based on wind speed and wind direction to achieve the trajectory prediction. The traditional time-series prediction methods to solve the trajectory prediction problem are inadequate in acquiring the complex nonlinear time-varying characteristics of the trajectory.

The third method is a machine learning based trajectory prediction method. This method has now become a research hotspot because it does not need to build a complex aircraft kinematic model. The literature[15] proposed a BP neural network-based radar track prediction method in a complex electromagnetic environment. The literature[40] constructed a 4D track prediction model using LSTM neural network, and used LSTM to focus on capturing the features of track points in the time dimension. The literature[18] proposed a CNN and LSTM-based track prediction method, which extracts the implied relationships within the track points and features in the temporal dimension by CNN and LSTM, respectively. The literature[8] proposed a track prediction model combining K-means clustering and gated recurrent unit (GRU) neural network, which clusters the track points with similar features into one flight state by K-means clustering algorithm, and trains the corresponding GRU (Gated Recurrent Unit) prediction model for each flight state. The literature[16] proposed three constraints for the climb, cruise and landing phases based on the dynamics of the aircraft and embedded them into the LSTM. In the literature[14], a four-dimensional joint cube structure was designed to fuse weather information with trajectory point information as the input of Hidden Markov Model (HMM) to predict the flight trajectory under environmental uncertainty.

Compared with the traditional methods, the existing deep learning-based track prediction methods have shown some superiority. However, these works ignore the interrelationship information between multiple attributes of the trajectory points, resulting in the prediction accuracy still needs to be improved. Therefore, we propose a graph convolutional network AR-GCN based on inter-attribute interrelationships to fuse attribute features through inter-attribute implicit relationships to improve the accuracy of trajectory prediction.

3 METHOD

3.1 Problem Definition

Given a series of historical trajectory points:

$$T = \{P_1, P_2, \ldots, P_K\} \tag{1}$$

where, P_i denotes the trajectory point at the moment i, and K denotes the time span of the trajectory. Each trajectory point consists of multiple attributes, denoted as:

$$P_i = \{a_{i1}, a_{i2}, \ldots, a_{iM}\}, i = 1, 2, \ldots, K \tag{2}$$

where, a_{im} is the m-th attribute in trajectory point P_i, M is the number of attribute, $1 \le m \le M$. Attributes include longitude, latitude, altitude and airspeed. For ease of presentation, we use A_m to denote the sequence composed by the m-th attribute in the historical time:

$$A_m = \{a_{1m}, a_{2m}, \ldots, a_{Km}\}, \quad m = 1, 2, \ldots, N \tag{3}$$

Therefore, from the perspective of attributes, the historical trajectory points can be expressed as:

$$T = \{A_1, A_2, \ldots, A_M\} \tag{4}$$

The historical trajectory points described from the temporal perspective and the historical trajectory points described from the attribute perspective are denoted as T^{time} and T^{attr}.

Based on these time-series data, we aim to predict the values of multiple attributes from the future trajectory points P_{K+1} to P_{K+Hor}, where H represents the prediction step size. To address this problem, we propose a graph convolutional network (AR-GCN) based on inter-attribute dependencies, and its general architecture is shown in Fig. 1. First, the attribute temporal features are extracted and the temporal features and the original attribute features are fused to obtain the enhanced attribute features. Secondly, the inter-attribute relationship features are extracted by the attention mechanism. Again, the enhanced attribute features are used as nodes and the inter-attribute relationship features are used as edges to construct the inter-attribute relationship graph. From the next, the aggregated attribute features are extracted from the inter-attribute relationship graph using a graph convolutional network to synthetically capture each attribute and the associated attribute features. Finally, the aggregated attributes are input to the multilayer perceptron to obtain the final prediction results.

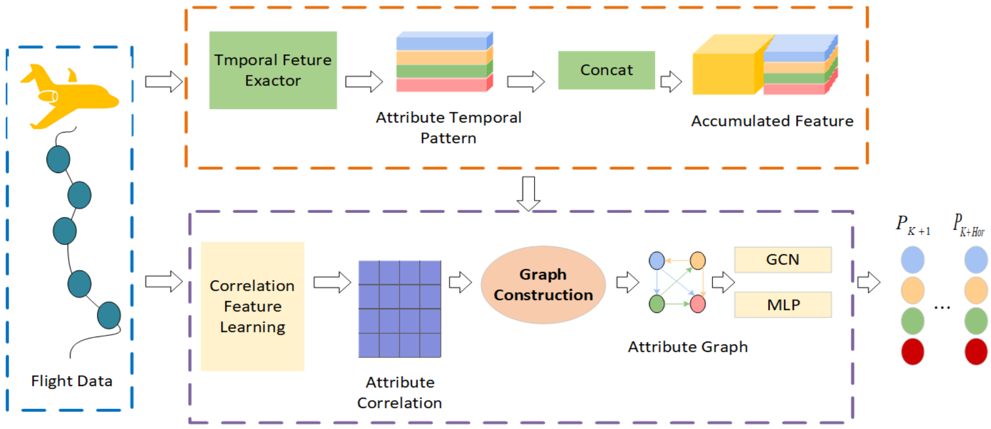

FIG. 1 THE FRAMEWORK OF AR-GCN

3.2 Enhanced Attribute Feature Learning

In this section, the original features of each attribute are extracted by performing an embedding operation on the attributes of each waypoint. The temporal features of each attribute are extracted by performing a multi-scale convolution operation on the temporal data of each attribute. These two features are fused to learn the augmented attribute features. The learning process of the enhanced attribute features is shown in Figure 2.

First, the embedding is done for each attribute and the embedding of all attributes is stitched to obtain the original features of the attributes. The calculation process is shown in Equation (5).

$$E^{ORI} = \phi\left(A_m, W_{Embed}^{ORI}\right) \tag{5}$$

where, $E^{ORI} \in R^{M \times D^{ORI}}$, D^{ORI} denotes the dimensionality of the embedding. $\phi(\cdot,\cdot)$ represents the linear transformation. $W_{Embed}^{ORI} \in R^{K \times D^{LEA}}$ is the weight coefficient of linear variation.

Second, for each attribute sequence A_m, the global time-varying features of the attribute are extracted by multiple multi-scale convolutions with different convolution kernel sizes. Given the set of values of an attribute m at all historical time points $A_m \in R^{1 \times K}$, the 1D convolutions of 1×3, 1×4 and 1×5 are used to extract the temporal feature information at different time steps, and the results obtained from each convolution are stitched to obtain the multi-scale time-varying feature representation vector $U_m \in R^{1 \times D^u}$ of the corresponding attribute , where D^u denotes the dimensionality of this feature vector, and the extraction equation is as follows.

$$U_m = Conv(A_m, \kappa_{(1\times3)}) \| Conv\left(A_m, \kappa_{(1\times4)}\right) \| Conv\left(A_m, \kappa_{(1\times5)}\right) \tag{6}$$

Finally, the original attribute features E^{ORI} are spliced with the time-varying attribute features U_m. Finally, an enhanced attribute feature $F \in R^{M \times \left(D^u + D^{ORI}\right)}$ containing both the original and time-varying features is obtained.

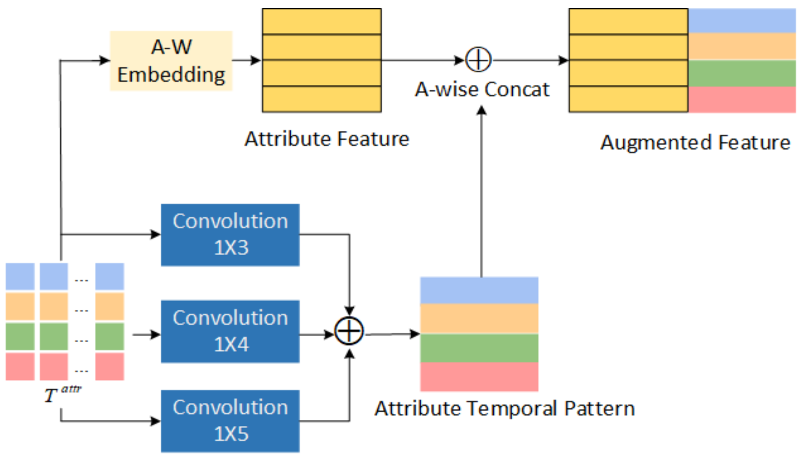

FIG. 2 ENHANCED ATTRIBUTE FEATURE LEARNING

3.3 Inter-Attribute Relationship Feature Learning

There are implicit interrelationships between attributes. To enrich the information contained in the inter-attribute relationship features, this section uses a self-attentive mechanism for each attribute's feature to obtain the inter-attribute correlations. The learning process of inter-attribute relationship features is shown in Figure 3.

The attribute attention score matrix is calculated for each attribute feature using the self-attention mechanism, called the attribute relationship feature matrix $R \in R^{M \times M}$, and the process of calculating the self-attention is shown in Equations (7) to (9).

$$Q = \phi\left(E^{ORI}, W_{\text{Query}}\right) \tag{7}$$

$$Key = \phi\left(E^{ORI}, W_{\text{Key}}\right) \tag{8}$$

$$R = soft\max\left(\frac{Q(Key)^T}{\sqrt{d}}\right) \tag{9}$$

where, $\phi(\cdot,\cdot)$ represents the linear transformation. Q and Key are the query and key of the attention mechanism. $W_{\text{Query}} \in R^{D^{ORI} \times D_{\text{Query}}}$, $W_{\text{Key}} \in R^{D^{ORI} \times D_{Key}}$ are the weighting factors of the linear transformation, respectively. $\sqrt{d} = \sqrt{D_{Query}}$ is the scale factor that ensures the stability of the values[18]. R reflects the correlation between multiple attributes internally.

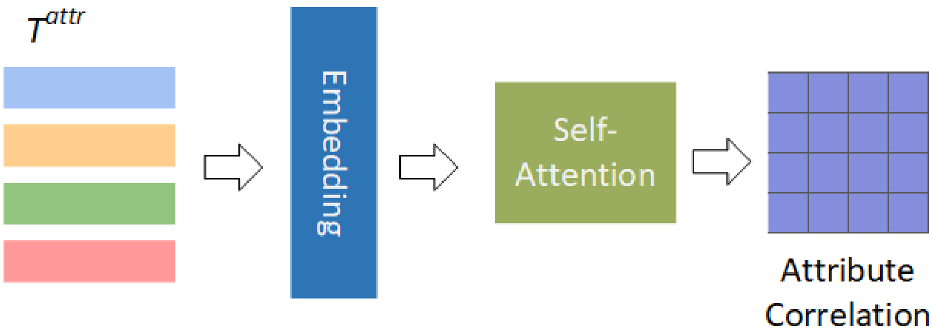

FIG. 3 INTER-ATTRIBUTE RELATIONSHIP FEATURE LEARNING

3.4 Inter-Attribute Relationship Graph Construction and Aggregation Attribute Learning

In order to learn the fusion features containing rich global and local attribute interrelationships, this section constructs the inter-attribute relationship graph and learns the aggregated attribute features of each attribute from this graph.

First, we construct the inter-attribute relationship graph for each moment by using the enhanced attribute features as nodes and the inter-attribute relationship features as edges at each moment. The inter-attribute relationship graph of each moment is stitched by time points to obtain the inter-attribute relationship graph $G = (F, R)$.

Next, the node information in the graph convolutional network inter-attribute relationship graph is aggregated using to learn the aggregated attribute features H, which is calculated as shown in Equation (10).

$$H^{(l)} = \sigma\left(RH^{l-1}(W)^l\right) \tag{10}$$

where, l denotes the number of layers of the graph convolutional network. $(W)^l$ is the coefficient of the first layer of the graph convolutional network. $\sigma(\cdot)$ denotes the activation function. $H^{(0)}$ is generated by the initialization G. After the l layers' graph convolutional neural network, the learned feature representation incorporates the time-varying features of each attribute and the associated attribute features.

3.5 Prediction Module

The aggregated attribute features obtained from the graphical convolutional neural network are fed into the MLP, and the output of the MLP is the final prediction result $\left\{P_{K+1}, P_{K+2}, \ldots, P_{K+Hor}\right\}$.

4 EXPERIMENT

4.1 Dataset

The version of this template is V2.6. Most of the formatting instructions in this document have been compiled from LaTeX style files. This offers both A4 templates and US Letter templates for LaTeX and Microsoft Word.

The trajectory attributes we consider include longitude, latitude, altitude and airspeed. The flight data include the waypoints (including time, longitude, latitude, altitude and airspeed) of the aircraft during the whole process from take-off to landing. The units of longitude and latitude are degrees, the units of altitude are meters and the units of speed are kilometres per hour.

Automatic Dependent Surveillance Broadcast (ADS-B) data is used to evaluate the performance of the AR-GCN. ADS-B automatically obtains parameters from relevant on-board equipment and reports the aircraft's position, altitude, speed, identification number, etc. to other aircraft or ground stations.

We downloaded the ADS-B data of several flights from the flightradar24 website. In the obtained ADS-B dataset, the time intervals of the trajectory points are unequal. Therefore, in this paper, the original data are segmented and the missing data are complemented by cubic spline interpolation, and the track point data are processed as equal time interval data (track point interval is 50 seconds). The take-off and landing phases include a small number of pre-take-off and post-landing taxiing records, which are not meaningful for the trajectory prediction problem, so this part of the data is excluded. The trajectory data used in the experiments are shown in TABLE 1.

The flights included three flights with flight numbers CZ6180 (Beijing-Changchun), MU5435 (Hefei-Chengdu) and DL528 (Washington-Atlanta). The flight data sets are as follows.

(1) Flight CZ6180 (Beijing-Changchun) flight data from April 15, 2022 to June 15, 2022, including 10 flights with a total of 1,241 track points.

(2) Flight data for flight MU5435 (Hefei-Chengdu) between April 15, 2022 and June 15, 2022, including 10 flights with 1,169 waypoints.

(3) Flight DL528 (Washington-Atlanta) from April 15, 2022 to June 15, 2022, including 10 flights with 1,595 waypoints.

All flights were divided into a training set (60%), a validation set (20%), and a test set (20%).

TABLE 1 DATASET OVERVIEW

Flight	Number
CZ6180（Beijing-Changchun）	10
MU5435（Hefei-Chengdu）	10
DL528（Washington-Atlanta）	10

4.2 Evaluation Indicators

Two commonly used performance metrics for evaluating track prediction methods are used in the experiments: Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) to measure the accuracy of model prediction. The MAE is the average of the absolute error between the predicted and actual values. The evaluation index is calculated by the following formula.

$$RMSE = \sqrt{\sum_{i=1}^{N} \left(a_{im} - \hat{a}_{im} \right)^2} \tag{11}$$

$$MAE = \frac{1}{N} \sum_{i=1}^{N} \left| a_{im} - \hat{a}_{im} \right| \tag{12}$$

Where N denotes the total number of predicted waypoints, the predicted value of the first attribute on the first waypoint, and the true value of the first attribute on the first waypoint. the smaller the value of RMSE and MAE, the better the performance of the method. a_{im} denotes the predicted value of the m -th attribute on the i -th trajectory point, $\hat{a}_{im}$ indicates the true value of the m -th attribute on the i -th trajectory point. Smaller values of RMSE and MAE indicate better performance of the method.

4.3 Parameter Setting

The prediction task in this paper is to predict the longitude, latitude, altitude, and velocity of one or more trajectory points in the future, given K trajectory points.

In the experiment, K is set to 5, i.e., given the first 250 seconds of the 5 track points data. The embedding dimension of the local attribute features is set to 64, and the Query and Key in Self-Attention, which calculates the feature matrix of local attribute relationships, are set to 64 dimensions. The sizes of the convolution kernels used for multi-scale convolution are 1×3, 1×4 and 1×5. The Embedding in Self-Attention, which calculates the attribute-relationship feature matrix, is set to 128 dimensions, and the Query and Key are set to 64 dimensions. The number of layers of the convolutional network is set to 2, and the nonlinear activation function is ReLU function. In this paper, the Adam optimizer [20] is used to minimize the mean absolute error MAE with a training period of 200 epochs and an initial learning rate set to 0.001, decaying by 0.1 at 100 epoch intervals. experiments are implemented on Pytorch [21].

4.4 Comparison Experiments

The AR-GCN is experimentally compared with the following three baseline methods.

LSTM [11]: a classical method for processing temporal data, a kind of recurrent neural network, which introduces a gating component on the basis of RNN and overcomes the problem of RNN gradient disappearance or gradient explosion to some extent.

Clustering-GRU [8]: abbreviated as C-GRU in the experiments, it is one of the most advanced track prediction algorithms, combining clustering algorithm and GRU recurrent neural network to segment the whole track for prediction.

CNN-LSTM [18]: one of the current advanced track prediction algorithms, which extracts the spatial and temporal features of the track data using CNN and LSTM, respectively

The performance on the single-step prediction problem is shown in TABLE 2, where the best values are marked in bold. The model in this paper outperforms the compared baseline model in terms of prediction accuracy for each attribute on both evaluation metrics, MAE and RMSE.

Compared with LSTM, AR-GCN reduces MAE and RMSE by 9% and 16%, respectively, in longitude prediction, 28% and 23%, respectively, in latitude prediction, 20% and 21%, respectively, in elevation prediction, and 13% and 9%, respectively, in velocity prediction. LSTM model has advantages in capturing the temporal features of the sequence of track points, but lacks the extraction of the information of each attribute within the track and the features of the relationship between attributes, which is compensated by the model in this paper, and the prediction results also show that the features of the attributes and the relationship between attributes extracted in this paper are effective.

Compared with C-GRU, the MAE and RMSE of GLAR-GCN are reduced by 2% and 4% for longitude prediction, 6% and 17% for latitude prediction, 7% and 17% for altitude prediction, and 2% and 5% for speed prediction, respectively. C-GRU combines the information of all attributes when extracting the features of each flight segment, but ignores the interrelationship between attributes, so it cannot provide accurate and rich trajectory representation to the prediction module, which makes the prediction accuracy of C-GRU still needs to be improved.

Compared with CNN-LSTM, GLAR-GCN reduces MAE and RMSE by 8% and 5% for longitude prediction, 6% and 17% for latitude prediction, 6% and 16% for elevation prediction, and 3% and 2% for velocity prediction, respectively. CNN is mainly used to capture the spatial relationship between attributes, and LSTM is used to capture the temporality between individual attributes, but they do not model the internal relationship between attributes, which makes the learned feature information only consider the spatio-temporal characteristics of attributes in isolation.

TABLE 2 SINGLE-STEP PREDICTION COMPARISON RESULTS (HORIZON=1)

		Attribute			
Methods	Metrics	Lon	Lat	Height	Speed
LSTM	MAE	0.0220	0.0260	147.027	7.948
	RMSE	0.0351	0.0328	232.660	13.015
C-GRU	MAE	0.0203	0.0197	125.490	6.917
	RMSE	0.0303	0.0304	221.092	12.010
CNN-LSTM	MAE	0.0217	0.0244	132.981	7.843
	RMSE	0.0346	0.0297	223.841	12.732
AR-GCN	MAE	**0.0199**	**0.0186**	**117.537**	**6.890**
	RMSE	**0.0294**	**0.0252**	**184.228**	**11.905**

The performance of the different models on the multi-step prediction problem is shown in [Table 3], where the best values are marked in bold. The model in this paper also outperforms the compared baseline model in terms of prediction accuracy for each attribute in both MAE and RMSE evaluation metrics.

Compared with the recurrent neural network-based prediction model, the model proposed in this paper reduces MAE and RMSE by 11%-22% and 12%-21%, respectively, in longitude prediction, 6%-19% and 6%-13%, respectively, in latitude prediction, 4%-15% and 4%-12%, respectively, in elevation prediction based on the internal relationship between attributes and 4%-12%, respectively, and MAE and RMSE on velocity prediction were reduced by 2%-7% and 1%-7%, respectively. The prediction accuracy of several models decreased as the prediction time increased, but the performance of the prediction model based on inter-attribute relationships decreased less than that of the prediction model based on recurrent neural networks. The experimental results show that the prediction model considering inter-attribute interrelationship achieves better prediction results, indicating that inter-attribute correlation is an important component of waypoint features, and therefore increasing the extraction of inter-attribute relationship features can improve the model prediction.

By comparing the experimental results for prediction step 1 and prediction step 3, it can be seen that the performance of AR-GCN decreases better with increasing prediction step than other compared algorithms, indicating that learning only from inter-attribute correlations can provide the prediction module with richer information for long time prediction.

TABLE 2 SINGLE-STEP PREDICTION COMPARISON RESULTS (HORIZON=1)

		Attribute			
Methods	Metrics	Lon	Lat	Height	Speed
LSTM	MAE	0.0358	0.0498	162.737	8.606
	RMSE	0.0680	0.1103	291.054	15.293
C-GRU	MAE	0.0316	0.0460	143.290	8.175
	RMSE	0.0663	0.1082	263.548	14.373
CNN-LSTM	MAE	0.0325	0.0427	158.621	8.213
	RMSE	0.0608	0.1020	285.841	14.277
AR-GCN	MAE	**0.0279**	**0.0401**	**137.537**	**8.045**
	RMSE	**0.0595**	**0.0958**	**253.228**	**14.197**

CONCLUSIONS

In this paper, a graph convolutional network (AR-GCN) based on inter-attribute intra-relationship is proposed to solve the flight trajectory prediction problem. The network first extracts the temporal features of each attribute and fuses them with the original features of the attribute to obtain the enhanced attribute features, and extracts the implicit relationship between the attributes as the inter-attribute relationship features; secondly, the enhanced attribute features are used as nodes and the inter-attribute relationship features are used as edges to construct the inter-attribute relationship graph; finally, the graph convolutional network is used to aggregate the attribute features. Based on the full fusion of the above features, high accuracy prediction of trajectories is achieved. In this paper, single-step and multi-step experiments are conducted on real flight track data, and the experimental results show that the prediction accuracy is improved by outperforming the comparison algorithm in all evaluation indexes.

REFERENCES

[1] Slattery R, Zhao Y. Trajectory synthesis for air traffic automation[J]. Journal of guidance, control, and dynamics, 1997, 20(2): 232-238.

[2] Gui G, Zhou Z, Wang J, et al. Machine learning aided air traffic flow analysis based on aviation big data[J]. IEEE Transactions on Vehicular Technology, 2020, 69(5): 4817-4826.

[3] Lymperopoulos I, Lygeros J. Sequential Monte Carlo methods for multi - aircraft trajectory prediction in air traffic management[J]. International Journal of Adaptive Control and Signal Processing, 2010, 24(10): 830-849.

[4] Wang C, Guo J, Shen Z. Prediction of 4D trajectory based on basic flight models[J]. Journal of southwest jiaotong university, 2009, 44(2): 295-300.

[5] Zhang J F, Jiang H, Wu X G, et al. 4D trajectory prediction based on BADA and aircraft intent[J]. Journal of Southwest of Jiaotong University, 2014, 49(3): 553-558.

[6] Jing X, Cui J, He H, et al. Attitude estimation for UAV using extended Kalman filter[C]//2017 29th Chinese Control And Decision Conference (CCDC). IEEE, 2017: 3307-3312.

[7] Xie L, Zhang J, Sui D, et al. Aircraft trajectory prediction based on interacting multiple model filtering algorithm[J]. Aeronautical Computing Technique, 2012, 42(5): 68-71.

[8] Han P, Wang W, Shi Q, et al. A combined online-learning model with K-means clustering and GRU neural networks for trajectory prediction[J]. Ad Hoc Networks, 2021, 117: 102476.

[9] Jang J G, Choi D, Jung J, et al. Zoom-svd: Fast and memory efficient method for extracting key patterns in an arbitrary time range[C]//Proceedings of the 27th ACM International Conference on Information and Knowledge Management. 2018: 1083-1092.

[10] Paszke A, Gross S, Massa F, et al. Pytorch: An imperative style, high-performance deep learning library[J]. Advances in neural information processing systems, 2019, 32.

[11] Hochreiter S, Schmidhuber J. Long short-term memory[J]. Neural computation, 1997, 9(8): 1735-1780.

[12] Thipphavong D P, Schultz C A, Lee A G, et al. Adaptive algorithm to improve trajectory prediction accuracy of climbing aircraft[J]. Journal of Guidance, Control, and Dynamics, 2013, 36(1): 15-24.

[13] Maeder U, Morari M, Baumgartner T I. Trajectory prediction for light aircraft[J]. Journal of Guidance, Control, and Dynamics, 2011, 34(4): 1112-1119.

[14] Ayhan S, Samet H. Aircraft trajectory prediction made easy with predictive analytics[C]//Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2016: 21-30.

[15] Song L, Shengli W, Dingbao X. Radar track prediction method based on BP neural network[J]. The Journal of Engineering, 2019, 2019(21): 8051-8055.

[16] Chen Y, Sun J, Lin Y, et al. Hybrid N-Inception-LSTM-Based Aircraft Coordinate Prediction Method for Secure Air Traffic[J]. IEEE Transactions on Intelligent Transportation Systems, 2021.

[17] Z. Shi, M. Xu, Q. Pan, B. Yan, and H. Zhang, ‘‘LSTM-based flight trajectory prediction,’’ in Proc. Int. Joint Conf. Neural Netw. (IJCNN), Rio de Janeiro, Brazil, Jul. 2018, pp. 1–8.

[18] Ma L , Tian S . A Hybrid CNN-LSTM Model for Aircraft 4D Trajectory Prediction[J]. IEEE Access, 2020, PP(99):1-1.

AUTHORS

[1]**Liwei Wu,** Master's degree student, School of Computer and Information Technology, Hefei University of Technology. Research Interests: Trajectory prediction and flight delay analysis.

[2]**Yuqi Fan,** Associate Professor, School of Computer and Information Technology, Hefei University of Technology. Research interests: Artificial Intelligence and Blockchain.