Short-term 4D Trajectory Prediction Using Machine Learning Methods

Zhengyi Wang, Man Liang, Daniel Delahaye

Short-term 4D Trajectory Prediction Using Machine Learning Methods

Zhengyi WANG
Sino-european Institute
of Aviation Engineering
Civil Aviation University of China
Tianjin, China

Man LIANG
Ecole Nationale
de l'Aviation Civile
Toulouse, France

Daniel DELAHAYE
Ecole Nationale
de l'Aviation Civile
Toulouse, France

***Abstract*—4D trajectory prediction is the core element of future air transportation system, which is intended to improve the operational ability and the predictability of air traffic. In this paper, we introduce a novel model to address the short-term trajectory prediction problem in Terminal Manoeuvring Area (TMA) by application of machine learning methods. It consists of two parts: clustering-based preprocessing part and Multi-cells Neural Network (MCNN)-based machine learning part. First, in the preprocessing part, Principle Component Analysis (PCA) is applied to the real 4D trajectory dataset for reducing the vector variable dimensions. Then, the trajectories are clustered into partitions and noises by Density-Based Spatial Clustering of Applications with Noise (DBSCAN) method. After that, the Neural Network (NN) model is chosen as machine learning method to find out the good predicting model for each individual cluster cell. Finally, with the real traffic data in Beijing TMA, the predicted Estimated Time of Arrival (ETA) for each flight is generated. Experiment results demonstrate that our proposed method is effective and robust in the short-term 4D trajectory prediction. In addition, it can make an accurate trajectory prediction in terms of MAE and RMSE with regards to comparative models.**

***Keywords*—Air Traffic Management, 4D Trajectory Prediction, Data mining, Machine Learning, Clustering, Neural Network**

I. Introduction

4D trajectory prediction refers to the calculation and prediction of longitude, latitude, altitude and time on the future waypoint sequence based on the existing data. During the development of Trajectory Based Operation (TBO) concepts in Single European Sky ATM Research (SESAR) and Next Generation Air Transportation System (NextGen) programs, trajectory prediction is intended to improve the predictability of air traffic, it is the core element of future air transportation system.

The 4D trajectory prediction can be influenced by several factors, such as aircraft weight, pilot actions, wind and temperature. These uncertainties will not only make it difficult to improve the prediction accuracy, but also will decrease the prediction process efficiency as the prediction time becomes longer[1]. According to the time scale, 4D trajectory prediction can be divided into two categories [2]:

1) Tactical (short-term) trajectory prediction: A prediction in a short period within several minutes or even shorter. Since the prediction scale is relatively small, minor change may have great impact on prediction results. Therefore, tactical trajectory prediction require as much information as possible. Flight-related information contained in radar or ADS-B data is usually taken;
2) Strategical (long-term) trajectory prediction: A kind of prediction before departure based on the flight plan, which provides the prediction from a macroscopic view. It is mainly applied to fuel consumption and airspace flow evaluation.

In this paper, we propose a novel short-term trajectory prediction model, which combines the different machine learning techniques to address the problem of 4D trajectory prediction in Terminal Maneuvering Area (TMA). This model can be divided into two main parts: preprocessing part and machine learning part. The preprocessing part contains several steps: data cleaning, filtering, re-sampling, Principle Component Analysis (PCA), density-based clustering and training. In the machine learning part, Multi-Cells Neural Networks (MCNN) technique will be applied to generate the predicted trajectory for different patterns.

II. Literature review

4D trajectory prediction can be mainly classified into aircraft performance models and machine learning models, according to input parameters models[3].

Aircraft performance models belong to physics-based approaches. The model structure is based on kinetic assumptions. The model parameters are determined based on a model of the aircraft performance, the planned flight routes, the predicted atmosphere condition, and the expected command and control strategies given by pilots or FMS (known as Aircraft Intent). The most precise aircraft performance model is Base of Aircraft Data (BADA) Family 4, which provides increased levels of precision in aircraft performance parameters for modelling and simulation [4]. A variety of researches based on BADA and Aircraft Intent have been conducted. In 2008, Lin Xi et al. presented a classified ADS-B-based trajectory prediction algorithm [5]. Based on the state estimation by Kalman filter and intent information captured by a pretreatment and probability method, the aircraft trajectory can be predicted with computation efficiency and less errors. M.Porretta et al. presented a novel aircraft performance model in consideration

of effects of wind, for aircraft lateral guidance and a new procedure for speed estimation [6]. The model input includes navigation data and aircraft intent information, based on EUROCONTROL BADA set. Simulation results show that the model is suitable for reliable trajectory prediction. In 2014, J. Kaneshige et al. described the implementation and evaluation of a motion-based trajectory prediction function, which can increase the resiliency and robustness of TBO [7]. Based on the performance index such as the fuel consumption, flight time, the algorithm computes the difference between with trajectory prediction and without trajectory prediction. Although, aircraft performance models have made great contributions to trajectory prediction, most of these models made ideal assumptions, rarely considered the real constraints, human behaviour factors, and the intersection of trajectories.

As a branch of Artificial Intelligence (AI), machine learning has been developed over 30 years, aims to learn from experiences and make predictions. The trend of recent years show that machine learning is widely used in trajectory prediction domain. Compared with those aircraft performance models, machine learning models were constructed with weak assumptions or even without assumptions. In some case, it shows better prediction performance. For example, in 1999, Yann Le Fablec et al. used Neural Networks to predict an aircraft trajectory in the vertical plane. The model is trained by a set of real historical trajectory, where two different method were adopted: in the first method, the input is current altitude, the remaining altitude to reach Request Flight Level (RFL) and n past vertical speeds, the output is the next speed; while in the second method, it is built with the starting altitude and the remaining altitude to reach, the RFL as input, the n first initial speeds as output. Simulation result showed that the Neural Networks give better results than classical prediction functions based on model of aircraft [8]. In 2013, De Leege et al. introduced Generalized Linear Models (GLMs) for trajectory prediction at a prediction horizon of 15NM to 45NM on fixed arrival route. The inputs of the model are aircraft type, ground speed at the Initial Approach Fix (IAF), altitude over the IAF, surface wind and altitude winds. All inputs come from surveillance data and meteorological data [9].

In the view of improving the accuracy in prediction tasks, S. Trivedi et al. carried out a study on the feasibility of utilizing clustering as a preprocessing approach [11]. Their research shows that the improvement on prediction accuracy is significant on large-scale cluster-able datasets by combining the clustering with even some simple machine learning predictors. Under routine traffic situation, in the TMA, the aircraft follows the standard arrival/departure procedure and regular ATC instructions, which makes trajectories cluster-able. Thus, application of machine learning together with clustering for 4D trajectory prediction in TMA is a valuable and interesting research topic. Several efforts on combining clustering with simple machine learning predictors have been investigated. For example, in 2014, K. Tastambekov et al. considered the short to mid-term aircraft prediction problem, namely, the prediction with a horizon of 10-30 min [1]. The model firstly searches similar trajectories in terms of shape and time, then uses wavelet decomposition to solve the linear regression model in the relationship between time and trajectory projection onto one of the three axis X, Y and Z. This method produces efficient results with high robustness. In 2015, S. Hong et al. introduced a new framework for predicting aircraft arrival times by combining the ATC intent information [12]. The training stage of the method contains two steps: trajectory pattern identification and regression models construction for each pattern. The prediction of arrival times can be achieved by applying different regression models for each trajectory pattern of target aircraft.

However, most of the aforementioned existing models still fall short. Some models neglect the prediction steps, directly consider clustering results as prediction results. A majority of trajectory pattern identification approaches are not robust, require high-quality flight data that follow the same departure/arrival procedure. If there are some noise and overflights, the results will be far less effective. In addition, the machine learning approaches that have been used are relatively simple and shallow in structure.

In this paper, we will extends the trajectory clustering method, which is introduced by Gariel et al. in reference [13], to study the short-term trajectory prediction model with machine learning methods. The main contributions of this paper are threefold:

1) A novel hybrid 4D trajectory prediction model based on clustering and MCNN is developed.
2) The proposed model is robust. The preprocessing part of the model can effectively and efficiently process the data, provide the high-quality inputs to the prediction part.
3) It can improve the accuracy of prediction. A comparative study is conducted to demonstrate the effectiveness of our model, compared with Multiple Linear Regression (MLR) model.

III. Methodology

A. Overview

The flow chart of the proposed trajectory prediction approach is demonstrated in Fig. 1. Our novel trajectory prediction approach includes two parts: clustering-based preprocessing part and MCNN-based machine learning part.

The DBSCAN method together with PCA form the preprocessing step. In this part, our model aims to identify the 4D trajectories into different clusters and remove noises in an efficient way. Each cluster symbolizes that the corresponding trajectories have the similar pattern. Noises contain trajectories with holding patterns, trajectories with large vectoring, the trajectory in special cases and overflight trajectories. After identifying the trajectory pattern and removing noises, the trajectory data quality will be highly increased.

In the part of machine learning, we apply the MCNN method to process different traffic data. First, for each partition of trajectories, there is a predictor, in which there is an

individual NN-based learning cell. Each individual learning cell will be trained with the associated cluster of trajectories. Consequently, each classified partition of trajectories will have its corresponding predicting model. Second, for the new input data, we will classify them into different corresponding clusters, then with our proposed multi-cells predicting model, trajectory prediction of the input data is generated.

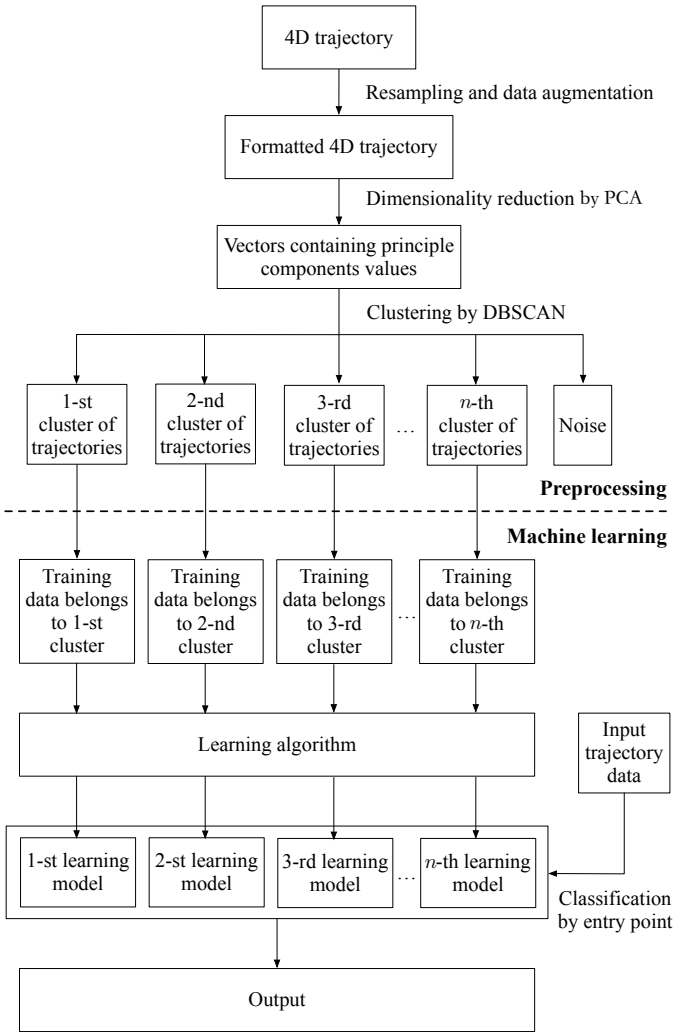


Figure 1: Proposed 4D trajectory prediction approach

B. Data preparation

The available dataset includes ADS-B records in July, 2017 over the TMA of Beijing Capital International Airport (BCIA), which is one of the busiest airport in the world, with three parallel runways: 18R/36L, 18L/36R and 01/19.

Since the studied airspace is relatively small, the longitude, latitude and altitude of trajectory points can be transformed into 3D Cartesian coordinates. Each sample of data contains:

1) Type of operation (departure/arrival),
2) Record beginning time t,
3) Aircraft number,
4) Position (X, Y, Z),
5) Heading Ψ,
6) Horizontal velocity V_h
7) Vertical velocity V_v, etc.

Each record with the same aircraft number belongs to an aircraft i, and the collection of all records for that aircraft forms the trajectory T_i, $i \in [\![1, n]\!]$, where n is the total number of trajectory in the dataset. Note that, in this paper, only flights that correspond to runways 18R/36L and 18L/36R are taken into consideration. These part of data consist of 36288 flights and 3242384 trajectory points.

Fig. 2 depicts the four traffic patterns in the 18R/36L and 18L/36R configuration, roughly clustered according to route nodes passed. Here, QFU means the magnetic orientation of runway-in-use. QFU 36 is to North, and QFU 18 is to South.

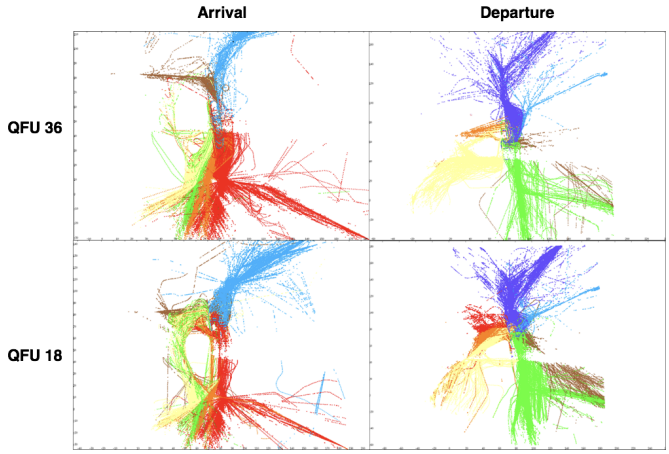


Figure 2: Runways 18R/36L and 18L/36R traffic patterns in Beijing capital international airport

C. Clustering-based preprocessing

The preprocessing part can be divided into the following steps:

1) Data cleaning and formatting,
2) Dimensionality augmentation,
3) Principal component analysis,
4) Clustering via DBSCAN.

Data Cleaning and Formatting: Due to the instability of ADS-B data receiver, our collected ADS-B data is not complete. Some trajectories have missing parts. It is necessary to filter them out. To solve this problem, a low pass filter is applied by the following function:

$$\tilde{x}_i^1 = x_i^1, \tag{1}$$

$$\tilde{x}_i^l = \alpha x_i^1 + (1-\alpha)\,\tilde{x}_i^{l-1}, l \in [\![2, m_i - 1]\!]. \tag{2}$$

Where the 3D coordinates and heading of the l-th point of i-th trajectory are substituted into x_i^l. α is a smoothing factor in $[0, 1]$. In this study, α is set to 0.5 to provide better results without too much delay. m_i is the number of points in i-th trajectory.

Trajectories with less than 50 points were eliminated due to statistical insufficiency. In order to make dataset suitable for

clustering, each trajectory should be represented as a vector. All the trajectory vectors are re-sampled into the same length, then their distance can be computed. The re-sample method for i-th trajectory is given as follow:

$$T_i = \left\{ T_i^l \,\middle|\, l = \text{round}\left(\frac{k \cdot m_i}{50}\right), k \in [\![1, 50]\!] \right\} \tag{3}$$

Dimensionality augmentation: This step aims to augment the dimensionality of dataset. The existing dimensions may not be sufficient and will result in lack of information, which can't completely reflect the differences between each trajectory. The augmentation of dimensions will help improve the clustering performance. Therefore, the following dimensions will be added into the dataset:

1) Distance from the reference point R, which indicates the convergence degree of trajectory. Due to the runway configuration, we define the reference point $(X_{\text{ref}}, Y_{\text{ref}}, Z_{\text{ref}})$ as (73.5, 65.5, 0). For each trajectory point, R_i^l is given as:
$$R_i^l = \sqrt{(X_i^l - X_{\text{ref}})^2 + (Y_i^l - Y_{\text{ref}})^2 + (Z_i^l - Z_{\text{ref}})^2} \tag{4}$$
2) Distance from the corner point D. According to the dataset, the corner point $(X_{\text{cor}}, Y_{\text{cor}}, Z_{\text{cor}})$ is assigned as (-50, 200, 0). The corner point will help solve the identifying problem when two trajectories are symmetric. The D_i^l is calculated by the function below:
$$D_i^l = \sqrt{(X_i^l - X_{\text{cor}})^2 + (Y_i^l - Y_{\text{cor}})^2 + (Z_i^l - Z_{\text{cor}})^2} \tag{5}$$
The reference point and corner point play the role as multilateration.
3) Angular position from the reference point Θ. It shows the variation (turning status) of trajectory with respect to the reference point. Θ is defined as:
$$\Theta_i^l = \arctan\left(\frac{Y_i^l - Y_{\text{ref}}}{X_i^l - X_{\text{ref}}}\right) \tag{6}$$

To sum up, the re-sampled dataset includes original features: position (X, Y, Z), heading Ψ and additional features: distance from the reference point R, distance from the corner point D, angular position from the reference point Θ. To avoid the discontinuity at $\pm\pi$, the sine and cosine values of Θ and Ψ is adopted.

Next, to make every feature on the same scale, each feature is normalized in $[0, 1]$. The general formula is given as:

$$x^* = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{7}$$

where x is the original feature and x^* is the normalized feature. Replacing x with our features, finally, the trajectory is organized as follows:

$$T_i = [P_i^* \quad R_i^* \quad D_i^* \cos(\Theta)^* \sin(\Theta)^* \cos(\Psi)^* \sin(\Psi)^*] \tag{8}$$

$$T = \begin{bmatrix} T_1 \\ \vdots \\ T_n \end{bmatrix} \tag{9}$$

where $P_i^* = [X_i^* \;\; Y_i^* \;\; Z_i^*]$. Then, each trajectory is re-sampled with 450 dimensions. Matrix T is $n \times 450$.

Principal Component Analysis: As shown in Eq. (8), trajectories are related to various of factors. Nevertheless, among these factors, some is more related, while the other is less related. Redundant elements will decrease computational efficiency, even lead to larger errors. To solve this problem, Principal Component Analysis (PCA) is introduced. PCA is a powerful tool used to reduce the dimension of dataset without losing too much information. The main idea of PCA is to derive an orthogonal linear transformation to project each of the vector variables into principal components for the maximum amount of variance that can be presented in lower dimensions[14].

PCA performs a linear transform on the $n \times m$ (in this case $m = 450$) matrix T:

$$Y = E \cdot T \tag{10}$$

Where E is a rotation matrix, Y is the new principal component matrix. The variance of Y is:

$$\text{var}(Y) = E^T \cdot C \cdot \tag{11}$$

Where C is the covariance matrix of T, which can be written as:

$$C = \frac{1}{n-1} \cdot T \cdot T^T \tag{12}$$

The eigenvalues of C can be calculated as $\{\lambda_i | i \in [\![1, m]\!]\}$, which correspond to the variances in Y as $\{v_i | i \in [\![1, m]\!]\}$, with $\lambda_1 > \lambda_2 > ... > \lambda_n$.

To map a dataset $X \subset \mathbb{R}^m$ to a dataset $Y \subset \mathbb{R}^q$ with $q \in [\![1, m[\![$, a rotation matrix $E = (v_1, ..., v_q)$ can be used. The dimension can be reduced by choose the number of q. It is required that the projection should better covers 95% of the variances, i.e., the cumulative percentage or variance explained $G(q)$ is greater than 95%:

$$G(q) = \frac{\sum_{i=1}^{q} \lambda_i}{\sum_{i=1}^{m} \lambda_i} \geq 95\% \tag{13}$$

Clustering via DBSCAN: As an unsupervised learning approach, Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is a commonly used density-based clustering algorithm [15]. The core concept of DBSCAN is to evaluate the density according to the number of points within the ϵ-neighbourhood. DBSCAN classifies the points into three types: core point, density-reachable point and noise point. The algorithm expands to density-reachable areas from a selected core point, then obtaining a maximum area including the core point and density-reachable points. Being robust to the quality of datasets, DBSCAN can divide the dataset into several clusters and noises, where the a-priori selection of the number of clusters is not required. Besides, DBSCAN is able to find arbitrarily shaped clusters. The advantages of DBSCAN make it fits well with trajectory clustering scenarios.

There are two principle parameters in DBSCAN algorithm: the neighbourhood radius ϵ and the minimum number of points required to form a cluster $MinPts$. These two parameters should be well chosen: The value of ϵ will affect the size of clusters. The value of $MinPts$ will affect the noise identification and the significance of clusters. After the proposed processing approach, the dataset for machine learning model will have better quality and the performance will be increased.

D. MCNN-based learning model

The machine learning used in our short-term trajectory prediction is supervised learning method. Supervised learning finds a mapping function from the input to the output based on the training data. The prediction can be achieved by applying the mapping function to the new inputs. As one of the most classical machine learning algorithms, regression model is commonly used in 4D trajectory prediction problem [16], [12], [1], [9]. A regression model can be expressed as:

$$y \approx f(x, \beta), \tag{14}$$

where y is dependent variable, x is independent variable, β represents parameters. More specifically, the Multiple Linear Regression (MLR) model is the most common form of regression analysis, frequently applied to prediction [12]. Given n multiple independent variables $\{x_i | i \in [\![1, n]\!]\}$ and corresponding dependent variable y, the model can be formalized as following:

$$y = \sum_{i=1}^{n} \beta_i x_i + \beta_0, \tag{15}$$

where $\{\beta_i | \beta \in [\![0, n]\!]\}$ are parameters, which can be approximated by least squares approach.

In this paper, we use MCNN model to predict the Estimated Time of Arrival (ETA) based on preprocessed real 4D trajectory data. The advantage of the usage of Neural Network (NN) in each prediction cell is that they are able to learn the hidden and non-linear dependencies from the training data. The architecture of proposed NN model for each cell is composed of an input layer, a hidden layer and an output layer, shown in Fig. 3. Given input $\{x_j | j \in [\![1, n]\!]\}$ and the hidden layer node number m, the network output can be calculated as:

$$y = \sum_{i=1}^{m} w_i^2 f\left(\sum_{j=1}^{n} w_{ij}^1 x_j + b_i\right) + c \tag{16}$$

Where w_{ij}^1 is the weight between the j-th input node and the i-th hidden node, w_i^2 is the weight between the i-th hidden node and the output node, b_i is the bias to the i-th hidden layer, c is the bias to the output layer. f is the activation function, in which Sigmoid function is commonly used. To find suitable weights such that the NN is in good performance, the cost function should be minimized. To increase the efficiency of updating the gradients, a prevailing cost function: cross-entropy cost function J is used:

$$J = -\frac{1}{N} \sum_{x} [t \ln y + (1 - t) \ln(1 - y)] \tag{17}$$

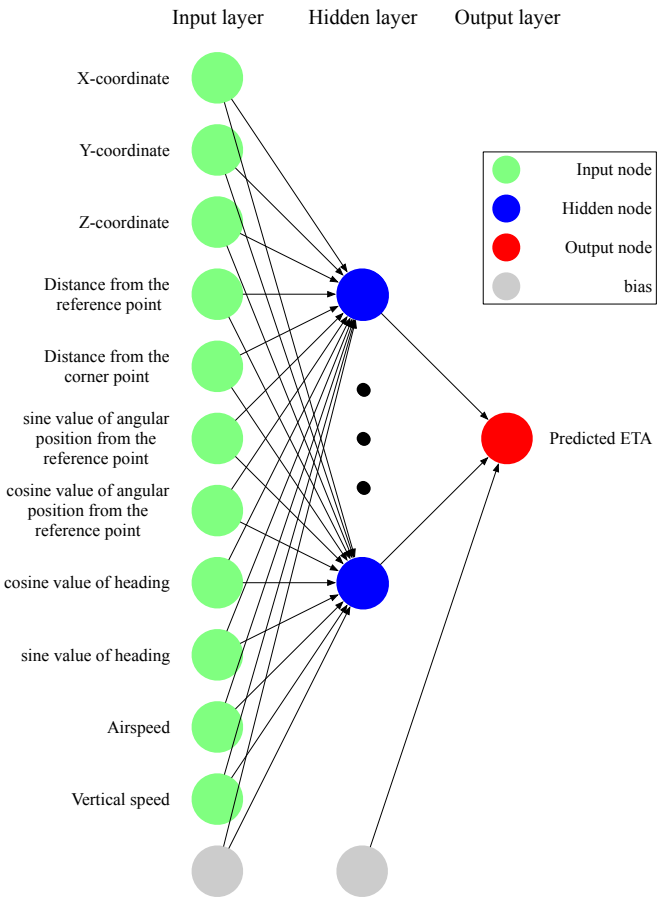


Figure 3: Neural network architecture used in this paper

where N is the number of training data, t is the target output. The steep descent is used to update and obtain the optimized parameters, which can be computed by well-known back propagation algorithm.

The new input can be classified according to the initial point of each trajectory. In view of arrival flights in TMA, initial points of trajectories in each cluster belong to a certain range in 3D Cartesian coordinate system. This character of dataset can be used to realize an effective classification on each new input trajectory.

E. Nested cross validation

In order to well select the parameters of prediction model, and to achieve an unbiased performance of the prediction model, this paper utilizes nested cross validation method. It consists of the outer loop and the inner loop. In the outer loop, there is a k_1-fold cross validation that splits the data into $k_1 - 1$ folds of training sets and one fold of test set. Then in the inner loop, there is another k_2-fold cross validation, which will further split the training set into $k_2 - 1$ fold of training sets and one fold of validation set. Taking $k_1 = 5$, $k_2 = 5$, the concept of the whole process is demonstrated by Fig. 4. The proportion of training sets, validation sets and test sets is $64\%/16\%/20\%$. The purpose is that the inner loop is for parameters selection, such as learning rate, number of hidden

nodes, and the outer loop is to validate the robustness of our prediction model.

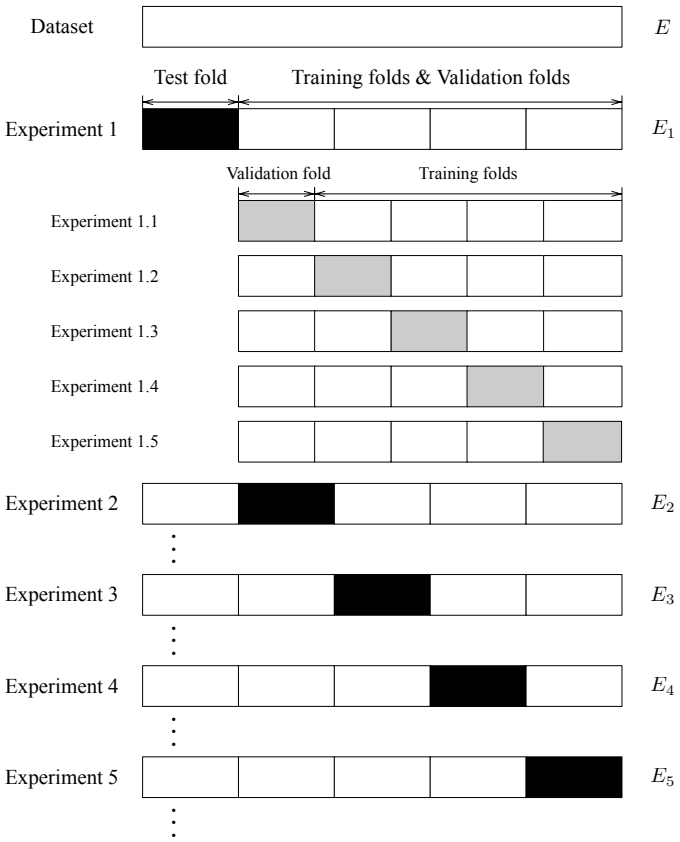


Figure 4: Nested cross validation procedure

Here, we use Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) to evaluate our trajectory prediction model performance:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i| \tag{18}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}, \tag{19}$$

where $\hat{y}_i$ is the i-th predicted value and y_i is the i-th observed value of ETA. A smaller value of MAE or RMSE represents a better accuracy of prediction.

Given that each outer iteration produces a $\mathrm{MAE_i}$, and a RMSE_i, $i \in [\![1, k_1]\!]$, the average MAE and RMSE can be computed as follows:

$$\mathrm{MAE} = \frac{1}{k_1}\sum_{i=1}^{k_1}\mathrm{MAE}_i, \tag{20}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{k_1}\sum_{i=1}^{k_1}(\mathrm{RMSE}_i)^2}, \tag{21}$$

IV. Simulation and Result

A. Dataset

The dataset that we used in the experiments contains 8677 arrival flights of QFU 36 extracted from the available dataset described in section III-B.

B. Results and discussion

In this study, the cumulative percentage of variance is calculated and presented in Fig. 5. We can see that when the principal component reaches over 32, the variance explained will be more than 95%. Let $q = 32$, then the dimension of each trajectory was reduced to 32 from 450. To sum up, dimensionality augmentation enriches the features that principle components can choose. PCA reduce the dimension of the dataset, which makes the following clustering step more efficient and accurate in the projected principal component space.

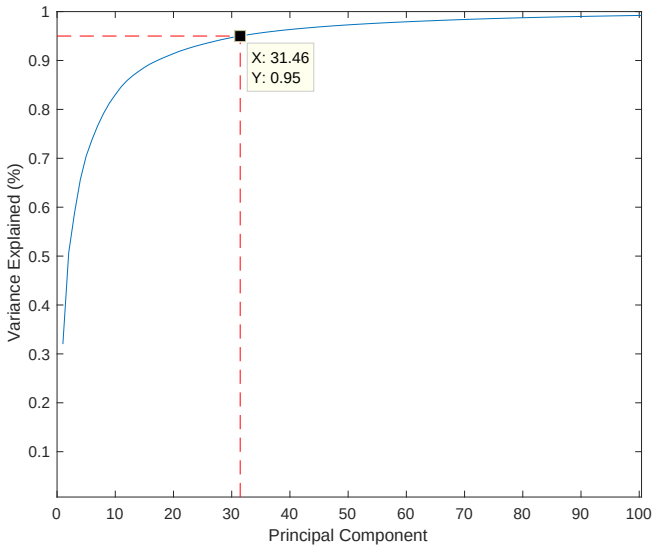
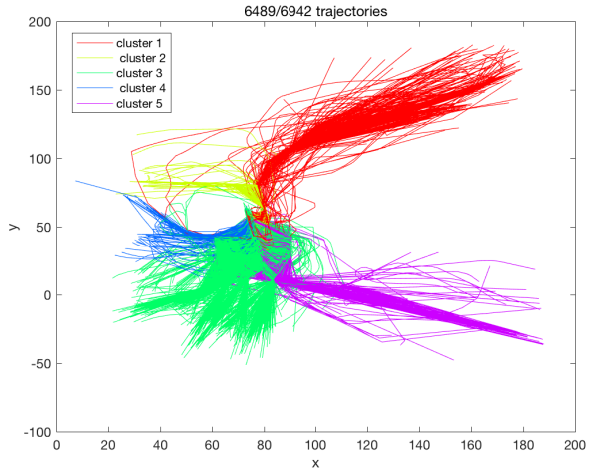


Figure 5: The cumulative percentage of variance in PCA

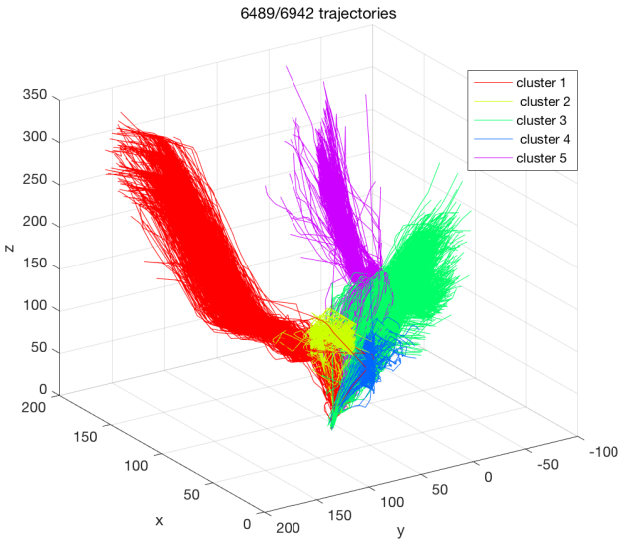
For DBSCAN step, experience shows that setting the parameters as $\epsilon = 1.8$ and $MinPts = 200$ is an optimum choice for this dataset. The distance metric used is Euclidean distance. taking a randomly generated training fold & validation fold for demonstration proposes. The resulting clusters is presented in Fig. 6a (trajectories in 2D) and Fig. 6b (trajectories in 3D), the noises is presented in Fig. 7.

According to Fig. 6 and Fig. 7, the trajectories are divided into 5 clusters. Clustered trajectories account for 93.47% of total trajectories. Noises represent 6.53%. Fig. 7 shows that the noise is mainly composed of holding patterns and trajectories with large vectoring, which will have an interference for prediction stage. Therefore, the noise should be removed from the dataset. In addition, there is no significant reduction on numbers of trajectory in the dataset.

The clustered partitions for each iteration is illustrated in Fig. 8, in which each trajectory is presented with its first 3 principle components. As we can see, 5 similar partitions



(a) 2D plot



(b) 3D plot

Figure 6: Cluster result of QFU 36 arrival trajectories example

were clustered for each iteration. The minimum proportion of clustered trajectories represent 93.22% and the corresponding noises account for 6.78% of all trajectories. The percentage is reasonable, which will not only eliminate the bad effect by noise, but also will keep most of the information.

To compare the performance of MCNN learning with the simple machine learning model, the Multiple Linear Regression (MLR) was proposed with the same clustering preprocessing step, and 5-fold cross validation is applied. The average proportion of test sets in each clusters and the ETA prediction errors of the proposed NN model and MLR were summarized in Tab. I. According to the Tab. I, with the same preprocessing procedure, the proposed NN model performs significantly better than MLR model in view of MAE and RMSE, not only

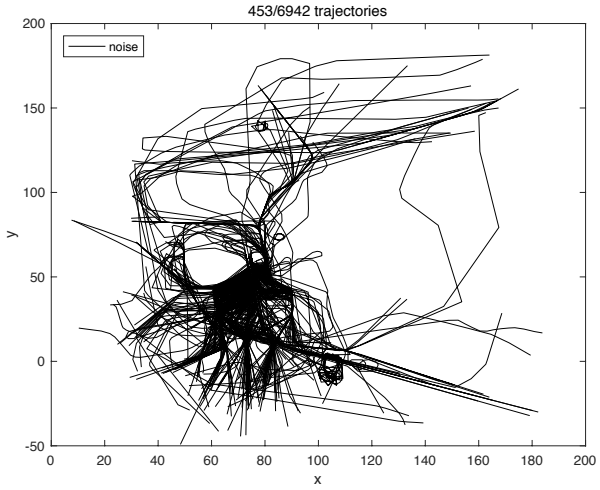


Figure 7: Noises result of QFU 36 arrival trajectories example

in total, but also for each cluster.

To illustrate the importance of the proposed clustering preprocessing step mentioned in section III-C, the prediction errors of NN model and MLR model both without preprocessing are presented in Tab. II. We can see from the Tab. II and Tab. I that in view of the same machine learning method, the model with clustering preprocessing step has less prediction errors than the one without clustering preprocessing step, which proves that the clustering preprocessing is effective in improving the prediction accuracy. Besides, the NN model prevails against the MLR model.

We further observe the distribution of ETA prediction errors with different prediction methods. In Fig. 9, X axis is the value of prediction error, Y axis is the frequency, which presents the percentage of trajectories on the associated error. With four different prediction methods, large part of trajectory predictions are all with less than 100 seconds error. Moreover, NN method performs better than MLR method. MLR with preprocessing method can improve the accuracy of prediction. The method NN with preprocessing performs the best ETA prediction. In addition, Fig. 10 reveals the mean absolute error of ETA prediction with the fly time to destination (runway). With four different prediction methods, the results show the same trend, that is: when the time to destination is fewer, the absolute prediction error is smaller. The NN with preprocessing performs best. In conclusion, the proposed model in this paper is efficient and able to make an accurate 4D trajectory prediction.

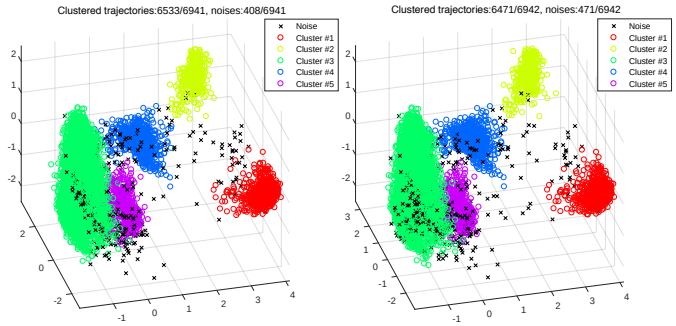
V. Conclusion

In this paper, a novel trajectory prediction approach that combines clustering with machine learning is proposed, implemented and simulated for ETA prediction.

The proposed model contains clustering-based preprocessing step and MCNN-based machine learning prediction step. First, it clusters different traffic flows, then it trains the associated prediction model for different clusters. After that, it is performed on real traffic data in Beijing TMA with nested

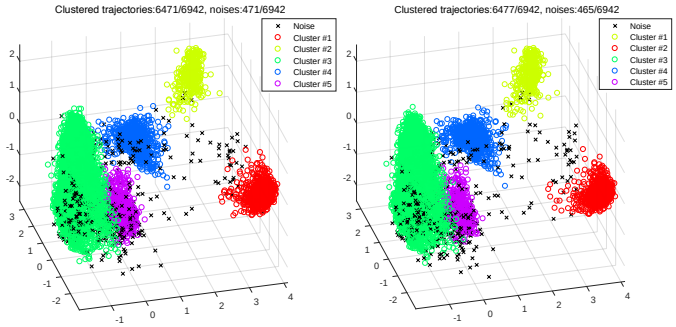
TABLE I. THE PERFORMANCE ON ETA PREDICTION OF NN AND MLR WITH PREPROCESSING STEP

Partition number	percentage	MAE for NN+P. (s)	RMSE for NN+P. (s)	MAE for MLR+P. (s)	RMSE for MLR+P. (s)
Cluster 1	13.85%	106.08	141.51	113.67	150.20
Cluster 2	5.62%	82.91	108.08	92.99	118.59
Cluster 3	58.39%	61.68	97.81	82.48	117.14
Cluster 4	13.64%	46.00	69.39	51.09	75.12
Cluster 5	8.51%	88.76	124.31	97.42	132.62
Total	100%	69.19	104.82	84.37	119.13



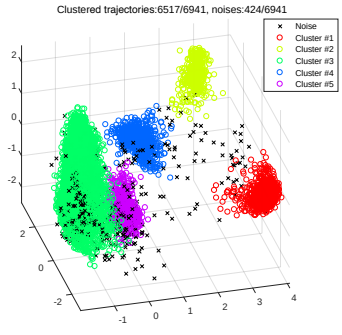
(a) 1st iteration

(b) 2nd iteration



(c) 3rd iteration

(d) 4th iteration

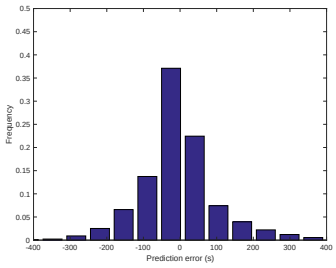


(e) 5th iteration

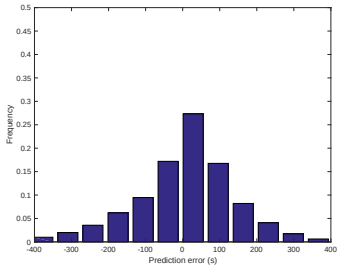
Figure 8: Illustration of clusters and noises for each outer iteration

TABLE II. THE PERFORMANCE ON ETA PREDICTION OF NN AND MLR WITHOUT PREPROCESSING STEP

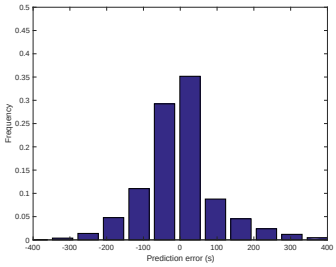
Model	MAE (s)	RMSE (s)
MLR without P.	108.03	160.40
NN without P.	76.28	127.76



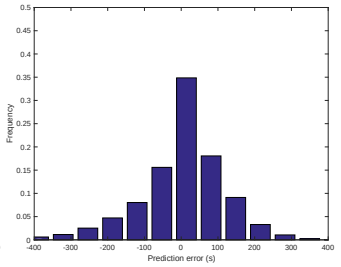
(a) NN without Preprocessing



(b) MLR without Preprocessing



(c) NN with Preprocessing



(d) MLR with Preprocessing

Figure 9: The distribution of ETA prediction errors with different methods

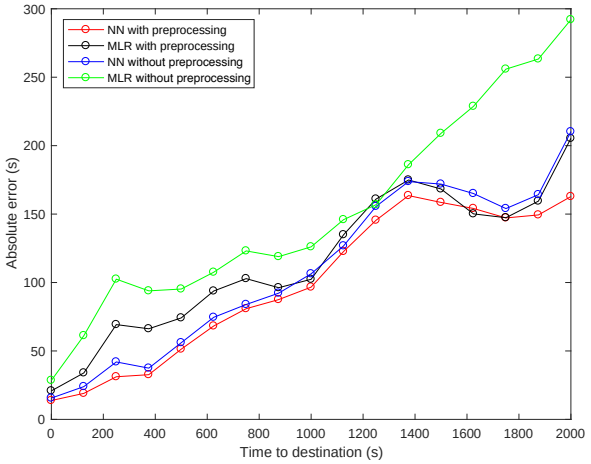


Figure 10: Mean absolute error of ETA prediction with the time to destination

cross validation. The numerical experiments demonstrate that the proposed method, NN with preprocessing, performs best in terms of MAE and RMSE, compared with other methods, such as NN without preprocessing, MLR without preprocessing, MLR with preprocessing. It can make an accurate 4D

trajectory prediction. In addition, the proposed method has a good robustness.

Future work could be conducted in different look-ahead times, on a comparison with results from model-based methods, as well as on studying prediction accuracy for other trajectory variables besides ETA. Moreover, more complex prediction model, such as deep learning approaches, would be very valuable.

VI. Acknowledgement

The authors would like to thank Serge Roux for his assistance with data collection, thank colleagues in the Optim group of ENAC and anonymous reviewers for their generous suggestions.

References

[1] K. Tastambekov, S. Puechmorel, D. Delahaye, and C. Rabut, "Aircraft trajectory forecasting using local functional regression in sobolev space," *Transportation research part C: emerging technologies*, vol. 39, pp. 1–22, 2014.

[2] X. Guan, X. Zhang, D. Han, Y. Zhu, J. Lv, and J. Su, "A strategic flight conflict avoidance approach based on a memetic algorithm," *Chinese Journal of Aeronautics*, vol. 27, no. 1, pp. 93–101, 2014.

[3] M. Hrastovec and F. Solina, "Prediction of aircraft performances based on data collected by air traffic control centers," *Transportation Research Part C: Emerging Technologies*, vol. 73, pp. 167–182, 2016.

[4] A. Nuic, D. Poles, and V. Mouillet, "Bada: An advanced aircraft performance model for present and future atm systems," *International Journal of Adaptive Control and Signal Processing*, vol. 24, no. 10, pp. 850–866, 2010.

[5] L. Xi, Z. Jun, Z. Yanbo, and L. Wei, "Simulation study of algorithms for aircraft trajectory prediction based on ads-b technology," in *System Simulation and Scientific Computing, 2008. ICSC 2008. Asia Simulation Conference-7th International Conference on*. IEEE, 2008, pp. 322–327.

[6] M. Porretta, M.-D. Dupuy, W. Schuster, A. Majumdar, and W. Ochieng, "Performance evaluation of a novel 4d trajectory prediction model for civil aircraft," *The Journal of Navigation*, vol. 61, no. 3, pp. 393–420, 2008.

[7] J. Kaneshige, J. Benavides, S. Sharma, L. Martin, R. Panda, and M. Steglinski, "Implementation of a trajectory prediction function for trajectory based operations," in *AIAA Aviation Atmospheric Flight Mechanics Conference, No. AIAA*, vol. 2198, 2014.

[8] Y. Le Fablec and J.-M. Alliot, "Using neural networks to predict aircraft trajectories." in *IC-AI*, 1999, pp. 524–529.

[9] A. de Leege0F, M. Van Paassen, and M. Mulder, "A machine learning approach to trajectory prediction," 2013.

[10] R. Alligier, D. Gianazza, and N. Durand, "Machine learning applied to airspeed prediction during climb," in *ATM seminar 2015, 11th USA/EUROPE Air Traffic Management R&D Seminar*, 2015.

[11] S. Trivedi, Z. A. Pardos, and N. T. Heffernan, "The utility of clustering in prediction tasks," *arXiv preprint arXiv:1509.06163*, 2015.

[12] S. Hong and K. Lee, "Trajectory prediction for vectored area navigation arrivals," *Journal of Aerospace Information Systems*, 2015.

[13] M. Gariel, A. N. Srivastava, and E. Feron, "Trajectory clustering and an application to airspace monitoring," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1511–1524, 2011.

[14] T. A. Runkler, *Data Analytics*. Springer, 2012.

[15] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *Kdd*, vol. 96, no. 34, 1996, pp. 226–231.

[16] M. Ghasemi Hamed, D. Gianazza, M. Serrurier, and N. Durand, "Statistical prediction of aircraft trajectory: regression methods vs point-mass model." ATM Seminar, 2013.