# Automated Data-Driven Prediction on Aircraft Estimated Time of Arrival

Zhengyi Wang
Ecole Nationale de l'Aviation Civile
Toulouse, France

Man Liang
University of South Australia
Adelaide, Australia

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 aimed at improving the operational ability and the predictability of air traffic. In this paper, we introduce a novel automated data-driven model to deal with the short-term trajectory prediction problem in Terminal Manoeuvring Area (TMA). The proposed model consists of data mining and Deep Neural Networks (DNNs). Firstly, the dataset is analyzed and cleaned by several criterions. Then, the flights in the dataset are split into partitions according to the runway in use (QFU). Prediction models of each QFU will be trained by the corresponding dataset. The experiments were firstly performed on real traffic data in Beijing TMA for 5 Neural Networks (NNs) models with nested cross validation. The results demonstrate that the DNNs perform better than shallow NNs. In addition, comparative study on data mining is conducted and proves that the data mining operation is robust in processing outliers, missing point and noise, which greatly improves the prediction accuracy in terms of Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE). We finally introduced ensemble learning model by combining well-performed individual models with certain strategies. Compared to other models, the minimum rule and mean rule applied to Deep Forward Neural Networks (DFNNs) with 3 hidden layers and DFNNs with 4 hidden layers after data mining performs the best in terms of MAE and RMSE.**

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

## 1. Introduction

4D trajectory and flight information are crucial factors for the Trajectory Based Operation (TBO) and Collaborative Decision Making (CDM). High-fidelity 4D Trajectory Prediction (TP) capability is the cornerstone of the deployment of TBO concept. Through accurate 4D TP, the air traffic operational efficiency will be improved, and the flight cost and adverse environmental impact will be lowered. More importantly, the workload of controllers will be alleviated, which means that the maximum Air Traffic Management (ATM) capacity will be augmented [1].

There are several studies on TP, which can be divided into model-driven methods and data-driven methods. Classical model-based TP methods made the ideal assumptions about the motion of the aircraft, the atmospheric environment, and the flight performance, along with either parametric or physics-based trajectory models. They generally do not take the intersections between different trajectories and the real Air Traffic Control (ATC) human behavior factors into account. In addition, lacking of sufficient data support, computation resources and learning ability, the model-based TP method is much less effective when facing with massive real-time data in large-scale ATM system. With the purpose of overcoming drawbacks of model-driven approaches, nowadays, the focus of 4D TP has been gradually shifted to data-driven approaches.

The Data-driven AiRcraft Trajectory prediction research (DART) project is one of the recently launched research projects supported by SESAR joint undertaking, aiming to explore the applicability of data-driven approaches to the ATM domain [2]. As pointed by DART, data-driven techniques are able to train appropriate models from all relevant and actual historical data with no or few prior assumptions and few requirements for data quality. Compared to classical model-driven approaches, contextual features can be extracted, including ATC information, meteorological condition, human factors, which will be beneficial in modelling the ATM socio-technical system and taking operational constraints into account.

Machine learning models are the most prevailing techniques in data-driven approaches. Typical machine learning models for 4D TP field include linear regression models and Neural Networks (NNs) models. Linear regression models have solid and widely accepted mathematical foundations and can provide insights on the air traffic dynamics [3–7]. However, they have difficulties in handling 4D TP scenarios with high-complexity, multi-dimensions and high-nonlinearity. On the other hand, NNs models are much better to deal with such problems, even with outliers, missing or noisy data [8]. Recently, NNs models, as the most representative model, have been successfully applied to 4D TP [9–12]. However, there are still some shortcomings in the current NNs models. Firstly, in the aforementioned researches, NNs models generally have only 1 hidden layer. Although a number of theorems show that shallow NNs can approximate any function with arbitrary precision, there are no statements as to the efficiency of the representation. It is also indicated that most functions that can be represented compactly by deep architectures cannot be represented by a compact shallow architectures [13]. Thus, it is not clear whether shallow NNs structure is sufficient and effective for the 4D TP problem. Secondly, some studies require complicated preprocessing step, which is a trade-off between prediction performance and computational efficiency. Thirdly, most current models lack generalizability and automaticity. They are only applicable to one or few flights, aircraft types or departure/arrival procedures. If the problem is extended to other flights, airports, airspaces or scenarios, the defined model architectures and parameters need to be greatly modified. Last but not least, the data used in some researches is not openly shared due to security reasons and business interests. Besides, some flight data sources are not possible to implement real-time 4D TP, e.g., Quick Access Recorder (QAR) data, Flight Data Recorder (FDR) data, etc.

The objective of this paper is to test the effectiveness of Deep Neural Networks (DNNs) model for 4D TP, and weigh the computational efficiency of preprocessing and the prediction performance. Based on open historical ADS-B data, the model in this study will predict the Estimated Time of Arrival (ETA) of flights on the runway from the entry point of TMA, where is one of the most challenging places for ETA prediction due to complex traffic patterns, meteorological conditions, ATC command and human behavior. Besides, air traffic controllers have great interests about the information at the entry of TMA. Several preliminary efforts have been made by authors. Reference [12] introduced an ETA trajectory prediction framework including clustering-based preprocessing and Multi-Cells Neural Networks (MCNNs). However, this model is not fully automated. Firstly, only flights with the same magnetic orientation of the runway in use (QFU) can be handled by the model. If flights with multiple QFUs are taken into account, the trajectories will overlap and mix together, which will greatly bring difficulties for the clustering algorithm to extract meaningful traffic patterns. Secondly, hyperparameters need to be retuned if we generalize the model to other TMAs. In this paper, we will solve the previous problems and target the following contributions:

1) An automated 4D trajectories prediction model will be developed to handle routine traffic in TMA. Deep neural networks will be utilized to deal with the highly dense arrival trajectories in TMA.
2) A refined data mining method should be applied to handle more complicated traffic patterns. The proposed model should be robust and generalizable. It should be capable of processing 4D trajectory data of all landing flights in TMA, even with outliers, missing points and noise. Besides, the approach could be extended to other TMAs, without manually tuning the hyperparameters.
3) The preprocessing step is supposed to improve the prediction performance with very few computation complexities.
4) A comparative study will be conducted for selecting the good model structures.

## 2. Data Mining

### A. Data preparation

*1) Airport and TMA:* Beijing Capital International Airport (BCIA, ICAO: ZBAA) is selected as the study case. It is one of the busiest airports in the world, with 3 parallel runways: 18R/36L, 18L/36R and 01/19. The cumulative flight time of BCIA has reached $785,200$ hours in 2017. Besides, BCIA is the most irregular airports in China by the year 2017, with 87,300 irregular flights [14]. According to China Electronic Aeronautical Information Publication (EAIP) [15], Beijing TMA is illustrated in Fig. 1.

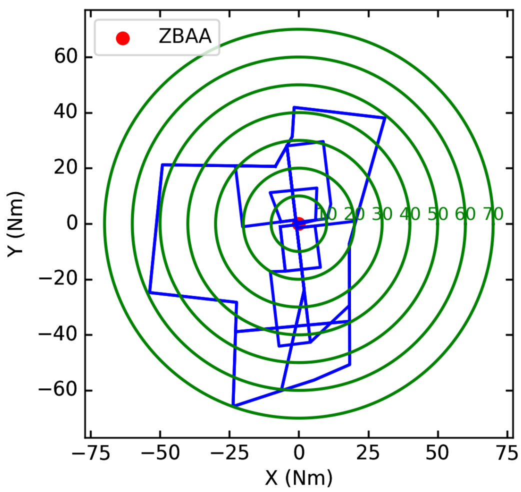



Figure 1: Beijing TMA

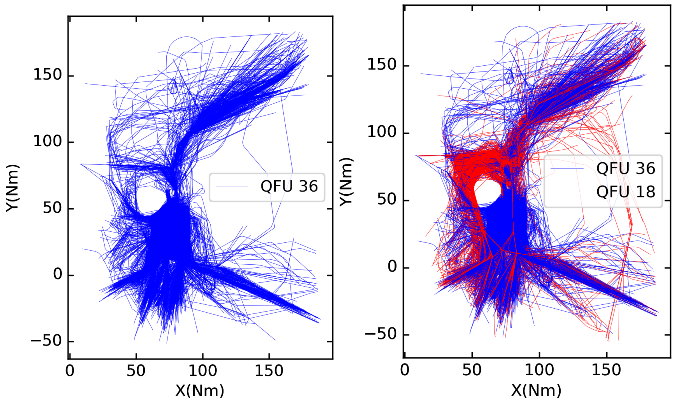
*2) Data type:* The data source of this study is ADS-B flight data, which is easily accessible and can provide accurate real-time report of aircraft's information. The dataset used in this study includes ADS-B records in July, 2017 that belong to Beijing TMA. Since the range of Beijing TMA is relatively small, Geographic Coordinate System (GCS) can be projected into Projection Coordinate System (PCS). Each record of ADS-B data contains the following information:

- Type of operation (departure/arrival),
- Runway in use,
- Record beginning time $t$,
- Aircraft ID,
- Position $(X, Y, Z)$,
- Heading $H$,
- Horizontal ground speed $V_h$
- Vertical ground speed $V_v$, etc.

Here, each record with the same aircraft ID $i$ belongs to a flight, and the collection of all records for that flight forms the trajectory $T_i$, $i = 1, ..., n$, where $n$ is the total number of trajectories in the raw dataset. In this dataset, $n = 12775$.

*3) Data volume:* Unlike the dataset used in Reference [12], which considered only one traffic operational direction, the dataset used in this study consists of the whole arrival flights in Beijing TMA, that is all landing directions. Note that the runway in use information about each flight can be obtained from the flight plan.

Raw trajectories studied in two cases are compared in Figure 2. In the previous model, the clustering algorithm used in Reference [12] cannot deal with too complex and overlapping scenarios. It can only handle trajectories coming from a specific QFU: QFU-36, see Figure 2a. The new model, which will be used in this study, will overcome the shortcoming of the previous model. It is able to handle two different runway-in-use directions, both QFU-36 and QFU-18, see Figure 2b. Remark that unusual flights can be seen as negative contributing factors to the prediction performance, which will be discussed later. All these raw trajectory data show that it is quite hard to provide an accurate 4D trajectory prediction in Beijing TMA with classic approaches.




(a) In previous model (b) In current model

Figure 2: Dataset used in researches of authors

## B. Data Analysis and Cleaning

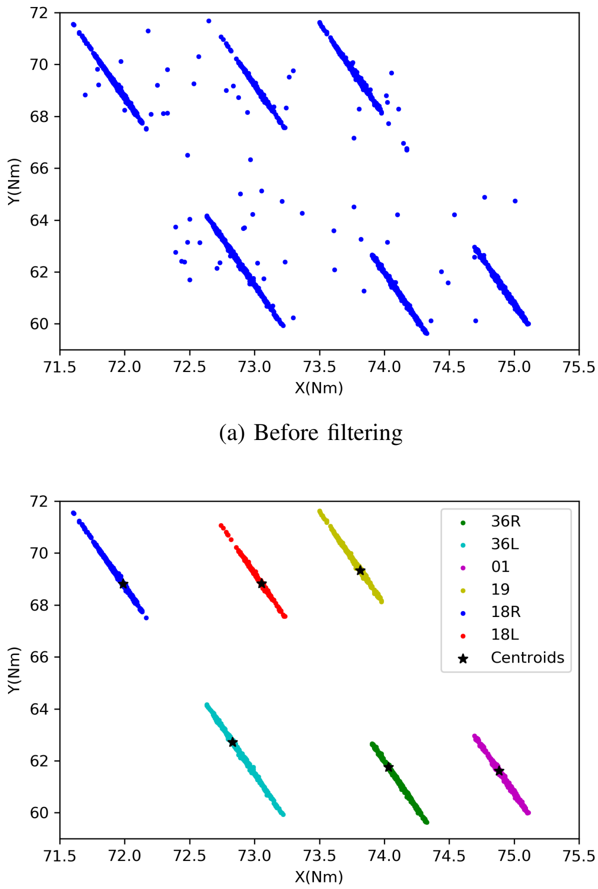
The dataset is firstly cleaned by the following criterions:

*1) Data receiving problem:* Different points of the same trajectory that have duplicated timestamps were removed. The problem lies in the integration of multiple data sources. Besides, trajectories with little recording points were eliminated from the dataset. In this study, 50 points are set to be the threshold. A total of 201 trajectories were removed in these 2 steps.

*2) The last recoding points:* In order to get the Actual Time of Arrival (ATA), we consider the first recording point on the ground as the landing point for each trajectory. Although most last recording points are along the runway, there are few out of runway. Those points have to be filtered. Figure 3a presents the last points of all trajectory. The points form 6 partitions, with little noise. After filtering by runway position and configuration, 110 trajectories whose last points are out of the runway were removed. Overall, the data validity rate reaches 97.57%. 6 partitions were labelled with corresponding magnetic orientation. In addition, centroids of each cluster are calculated and plotted, see Figure 3b.

*3) Transit time in TMA:* The nearest distance between entry points of TMA and runways of BCIA is approximately 20 to 25 Nm. This range is chosen to be focused in this study. Therefore, all trajectory points beyond the circles (with centers of corresponding centroid and radius of 25 Nm) were firstly removed from the dataset.

After data cleaning, the distribution of transit time in TMA in terms of QFUs is respectively plotted in Figure 4, where the transit time of an aircraft stands for the elapsing time between entering the TMA and landing. For each landing orientation, there are few flights that have long transit time.




(a) Before filtering

(b) After filtering

Figure 3: Last points of trajectories

Through visualization and analysis of 63 flights in Figure 5, most flights with approximately top 0.5% longest transit time are unusual flights, including holding patterns, flights with large vectoring, flights with go-around procedure, etc. Being negative contributing factors both for training and test of 4D trajectory prediction model, these flights are stochastic and irregular, which need to be filtered. In addition, new input trajectories whose predicted transit times are longer than the threshold (0.5% longest transit time of corresponding QFU, listed in Table I) are regarded as unusual trajectories. These trajectories cannot reflect the actual performance of this model and should be neglected. In this study, the unusual flights are not included in the training set. In the validation set and test set, if the ETA predicted by models exceeds the transit time threshold in Table I for each QFU, the related data will be discarded. Figure 6 portrays the remaining trajectories of different QFUs. In view of each QFU, it can be seen that the trajectories are much more regular after removing the abnormal ones. Furthermore, most of them follow the similar pattern, which can lead to better predictability.

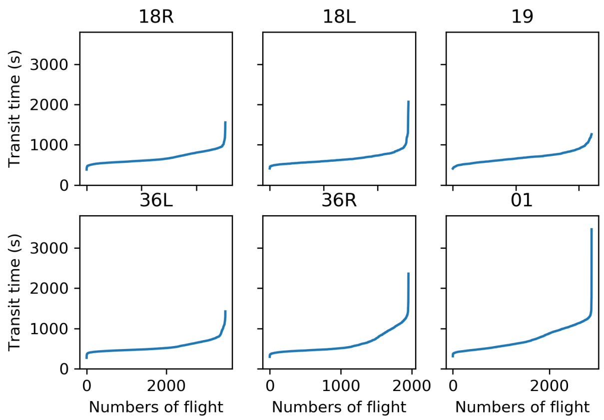



Figure 4: Distribution of transit time in TMA

TABLE I: Transit time threshold for all QFUs in Beijing TMA

| QFU | Transit time threshold (s) |
|---|---|
| 18R | 1151.0 |
| 18L | 1765.0 |
| 19 | 1227.0 |
| 36L | 1149.0 |
| 36R | 1396.0 |
| 01 | 1403.0 |

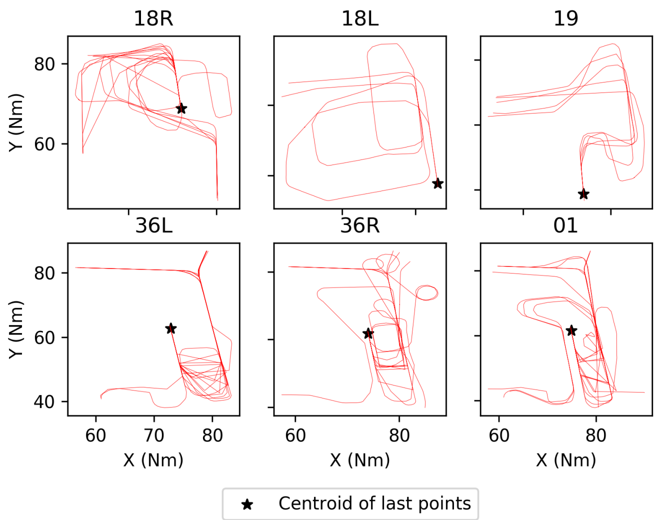



Figure 5: Trajectories of flights with top 0.5% longest transit time in TMA

## 3. Automated Prediction with Deep Neural Networks

### A. Deep Feed-forward Neural Networks

The goal of this part of research is to predict the ETA of aircraft at the entry of TMA. This is a regression problem, which can be expressed as the following fixed-design regression model:

$$y_n = f(x_n) + \epsilon_n, \quad n = 1, ..., N \tag{1}$$

where $N$ is the number of statistical units of input variables, $y_n$ are random variables that follow a mean function $f(\cdot)$ with errors $\epsilon_n$, which are independent and identically distributed

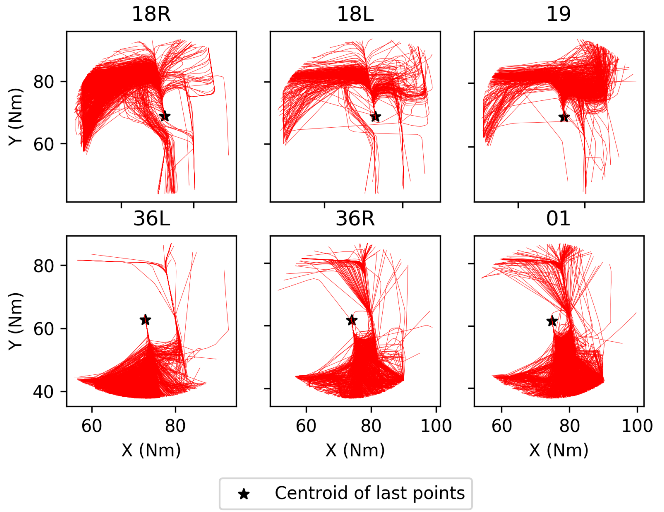



Figure 6: Other 99.5% trajectories

(i.i.d.) random variables, such that $E(\epsilon_n) = 0$ and $\text{var}(\epsilon_n) = \sigma^2$.

In these experiments, a specific class of NNs is introduced to approximate the function $f(\cdot)$, referred to as DFNNs (Deep Feedforward Neural Networks), which is one of the most quintessential deep learning models. DFNNs have an input layer, $L$ hidden layers $h^{(1)}, ..., h^{(L)}$ ($L \geq 2$) and an output layer. The output $y(x; W)$ is expressed as a function of the input vector $x$ as follows:

$$h^{(1)} = \Phi^{(1)}({W^{(1)}}^\top x) \tag{2}$$

$$h^{(l)} = \Phi^{(l)}({W^{(l)}}^\top h^{(l-1)}), \quad l = 2, ..., L \tag{3}$$

$$y(x; W) = \Psi({W^{(L+1)}}^\top h^{(L)}) \tag{4}$$

Where $W^{(1)}$, $W^{(l)}$ ($l = 2, ..., L$), $W^{(L+1)}$ are respectively weights assigned to the connections between input layer and first hidden layer, between $(i-1)$-th hidden layer and $i$-th hidden layer, and between $L$-th hidden layer and output layer. $\Phi^{(l)}$ is the activation function applied to the weighted output of the $i$-th layer of NNs. $\Psi$ is the function applied to the weighted sum of the activations of the last hidden layer. Each input $x$ is a vector that contains the 3D position, heading, horizontal and vertical ground speed. The target variable $y$ in our case is the ETA. Note that if $L = 1$, DFNNs degenerate to shallow neural networks with 1 hidden layer.

In consideration of dataset size, we limit the maximum number of hidden layers to 5, since large number of hidden layers may cause overfitting and bring the difficulty for training. 5 structures of NNs were introduced in this paper: NNs with 1 hidden layer and DFNNs with 2, 3, 4, 5 hidden layers. All networks have 6 input nodes, 1 output node, and 15 nodes in each hidden layer. Thus, the structures of 5 NNs are respectively 6-15-1, 6-15-15-1, 6-15-15-15-1, 6-15-15-15-15-1 and 6-15-15-15-15-15-1, which are illustrated in Figure 7.

To make every feature on the same scale, the explanatory variables $(X, Y, Z, H, V_h, V_v)$ are normalized in [-1, 1] by the following formula:

$$x^* = 2\frac{x - \min(x)}{\max(x) - \min(x)} - 1 \tag{5}$$

Rectified Linear Unit (ReLU) function is applied to all layers of these models as the activation function $\Phi$. Compared to Tanh and Sigmoid functions, ReLU has no gradient vanishing problem and is less computationally expensive [16].

$$\Phi(z) = \max(0, z) \tag{6}$$

$\Psi$ is the identity function:

$$\Psi(z) = z \tag{7}$$

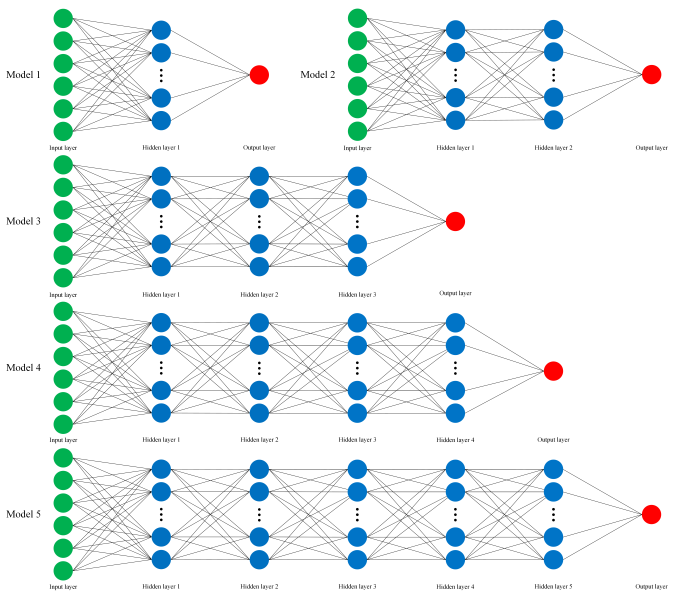



Figure 7: Structures of 5 NNs model used in this paper

The error functions on a training set $T$ of each model are defined as follows:

$$L_i(W) = \sum_{(x,t)\in T} (y_i(x;W) - t)^2 + \lambda_i \sum_l \left\| W_i^{(l)} \right\|_2^2,\ i = 1, ..., 3 \tag{8}$$

where $\lambda$ is the regularization coefficient. A properly selected $\lambda$ can avoid overfitting.

The backpropagation method was used to compute the gradient of the error function, and Adam [17] was used to minimize the error between our predictions of ETA and the ATA. We used an algorithm *reduce the learning rate on plateau* to reduce the learning rate when a metric stopped improving. The learning rate can be adaptively adjusted by this algorithm and need not to be tuned. In this paper, the initial learning rate is set to be 0.1. The learning rate will be reduced by a factor of 0.5 once there is no improvement of the error on the training set for 10 epochs. After learning rate has been reduced, wait 10 epochs before resuming normal operation. The lower bound on the learning rate is $10^{-4}$. All layers of each NNs are fine-tuned with 1000 epochs of training. All these methods were implemented in PyTorch [18].

## B. *Performance evaluation and model selection*

To assess the performance of DFNNs models, we evaluate the performance of 3 proposed architectures of NNs on the raw dataset and preprocessed dataset.

In order to well select the hyperparameters and to achieve an unbiased performance of the prediction model, the nested cross validation method is introduced. It consists of the outer loop and the inner loop. A $K_1$-fold cross validation splits the dataset $S$ into $K_1$ subsets $S_i$, $i \in \{1, ..., K_1\}$. For each outer iteration $i$, $K_1 - 1$ folds $S_{-i} = S \backslash S_i$ act as training sets and one fold $S_i$ is test set. Then, there is another $K_2$-fold cross validation, which will further split the training sets $S_{-i}$ into $K_2$ subsets $S_{-i,j}$, $j \in \{1, ..., K_2\}$. For each inner iteration $j$, $K_2 - 1$ folds $S_{-i} \backslash S_{-i,j}$ play the part of training sets and the remaining fold $S_{-i,j}$ is validation set. The purpose of the inner loop is the selection of hyperparameters and the outer loop aims to assess the model performance.

Taking $K_1 = 5$, $K_2 = 5$, the proportion of training sets, validation sets and test is set as $64\%/16\%/20\%$. Furthermore, in order to obtain the best performance of the prediction models, the spatial distribution of these sets should be as close as possible. Therefore, the trajectory points between 20 and 25 Nm away from corresponding runways of BCIA make up the dataset of different models. For example, Figure 8 illustrates the trajectories points of training, validation and test set of one possible cross validation fold of QFU-01 dataset.

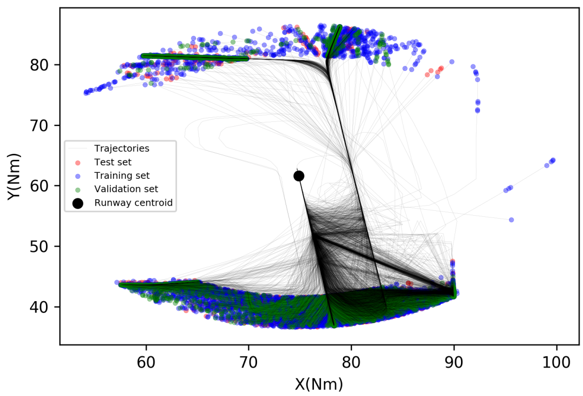



Figure 8: Illustration of training, validation and test set in a possible case

Here, we use a grid search algorithm [10] to tune hyperparameters. This algorithm aims at selecting the hyperparameter $\lambda$ of an algorithm $A_\lambda$ by performing a 5-fold cross-validation on a set of examples $T$, which is depicted by algorithm 1.

**Algorithm 1** Hyperparameters tuning

**function** TUNEGRID($A_\lambda$, $grid$)[$T$]
  $\lambda* \leftarrow \arg\min_{\lambda\in grid} CV_5(A_\lambda, T)$
  **return** $A_{\lambda*}[T]$
**end function**

The hyperparameter grids of aforementioned machine learning model are shown in table II.

TABLE II: Hyperparameter grids for machine learning algorithms

| Model | Hyperparameter grids |
|---|---|
| NNs | $\lambda = \{0.0001, 0.001, 0.01, 0.1, 0.5, 1, 2, 5\}$ |

Finally, we use Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) to assess the performance of trajectory prediction models:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n} |\hat{y}_i - y_i| \tag{9}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (\hat{y}_i - y_i)^2}, \tag{10}$$

where $\hat{y}_i$ is the $i$-th predicted value of ETA and $y_i$ is the $i$-th observed value of ETA.

## 4. Experiment results and discussion

Firstly, we evaluate the ETA prediction performance of 5 proposed NNs model with the data mining process. The proportions of flights for each QFU were presented in Table III.

The statistics of prediction results for 6 QFUs of BCIA were summarized in the first part of Table IV. The best MAE value among these 5 models of each QFU is bolded and colored in blue. Among 5 models, NNs with 1 hidden layer performs worst. Compared with shallow NNs, DFNNs are more suitable for ETA prediction task. However, it should be noted that excessive numbers of hidden layer may decrease the prediction performance due to overfitting. DFNNs with 3 and DFNNs with 4 layers are the best models in view of respectively 2 and 3 QFUs.

TABLE III: The proportions of flights for each QFU

| QFU | Percentage |
|---|---|
| 18R | 31.01% |
| 18L | 5.94% |
| 19 | 13.83% |
| 36L | 19.55% |
| 36R | 11.89% |
| 01 | 17.77% |

To further improve the prediction performance, ensemble learning is introduced in this study. The main concept of ensemble learning is to combine multiple models with a strategy in order to obtain better results than any of the models. Based on the ETA prediction performance of 5 proposed models, we selected the best predictors (DFNNs with 3 and 4 layers) and used 3 rule-based combiners (mean rule, maximum rule and minimum rule) to generate 3 new ensemble models [19]. The 3 proposed combiners are defined as follows:

$$\tilde{y}_\alpha(x) = \left(\frac{1}{T}\sum_{t=1}^{T} y_t(x)^\alpha\right)^{\frac{1}{\alpha}} \tag{11}$$

$$\text{The combiner is} \begin{cases} \text{Minimum rule}, \ \alpha \to -\infty \\ \text{Maximum rule}, \ \alpha \to +\infty \\ \text{Mean rule}, \ \alpha = 0 \end{cases} \tag{12}$$

where $T$ is the number of ensembles, $y_t(x)$ is the output value of $t$-th ensemble model, $\tilde{y}_\alpha$ is the final output value.

The prediction results of these 3 ensemble models were summarized in the second part of table IV. The best MAE value of all models for each QFU is bolded and colored in red. It can be observed that minimum rule applied to DFNNs with 3 and 4 hidden layers have the least MAE among all the models for all QFUs. Other ensemble models also have better performances compared to 5 fundamental models. Although the improvement by ensemble models are quite small, it provides a new idea of improving the model performance. In other problems or with more individual models, ensemble models may perform better.

To evaluate the contribution of the data mining process introduced in this study, we compared the ETA prediction performance of models with and without the data mining process. According to Table IV and Table III, the ETA predictions performances of NNs models with data mining process on the whole dataset were calculated. Together with the prediction statistics of NNs models without data mining process, the prediction results were shown in table V. In general, the effect of data mining in this scenario is significantly positive. In view of 5 individual NNs models, data mining operation reduces the MAE by nearly 20 seconds and the RMSE by over 30 seconds. Especially for shallow NNs model, the MAE is reduced by over 30 seconds and the RMSE is reduced by nearly 50 seconds. The results demonstrate that the data mining operation is robust to outliers, missing data and noise. The dataset after data mining is able to train better prediction models. In consideration of the great difference of prediction accuracy, there is no use in conducting ensemble models without data mining. The best MAE value among all models is bolded and colored in red. The overall performances of mean and minimum ensemble models are very close.

In view of the prediction error of shallow NNs model and the best DFNNs model, with data mining, the MAE is reduced by 2.94 seconds and the RMSE is decreased by nearly 2.04 seconds. Without data mining, the MAE is decreased by 16.19 seconds and the RMSE is decreased by 15.16 seconds. The result indicates that DFNNs performs better than shallow NNs. Especially when the quality of training data is poor, the performance improvement becomes more obvious, which reveals that DFNNs are more capable of extracting hidden information in noisy features than shallow NNs.

## 5. Conclusion

In this paper, a novel automated data-driven trajectory prediction approach is presented, implemented and simulated

TABLE IV: ETA prediction performance of models with data mining process of each QFU

| Models | QFU-18R | | QFU-18L | | QFU-19 | | QFU-36L | | QFU-36R | | QFU-01 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | MAE (s) | RMSE (s) | MAE (s) | RMSE (s) | MAE (s) | RMSE (s) | MAE (s) | RMSE (s) | MAE (s) | RMSE (s) | MAE (s) | RMSE (s) |
| NNs with 1 hidden layer | 41.11 | 59.15 | 73.26 | 96.37 | 70.23 | 93.06 | 40.47 | 60.25 | 57.25 | 97.98 | 65.87 | 112.89 |
| DFNNs with 2 hidden layers | 40.09 | 57.70 | **71.74** | 96.49 | 68.42 | 91.15 | 36.51 | 58.70 | 54.92 | 96.64 | 62.98 | 111.01 |
| DFNNs with 3 hidden layers | **39.46** | 57.63 | 72.70 | 97.13 | 67.47 | 90.71 | **35.57** | 57.50 | 53.84 | 96.68 | 62.23 | 110.25 |
| DFNNs with 4 hidden layers | 39.59 | 57.30 | 73.46 | 97.74 | **67.34** | 90.70 | 35.92 | 57.05 | **53.55** | 95.85 | **62.17** | 110.32 |
| DFNNs with 5 hidden layers | 39.77 | 57.64 | 72.87 | 99.36 | 67.69 | 90.30 | 35.69 | 56.89 | 54.25 | 96.55 | 62.64 | 110.07 |
| **Ensembles of DFNNs with 3 hidden layers and DFNNs with 4 hidden layers** | | | | | | | | | | | | |
| Mean of ensembles | 39.46 | 57.31 | 72.06 | 95.90 | 67.01 | 90.12 | 35.56 | 57.06 | 53.41 | 95.89 | 61.66 | 109.85 |
| Min of ensembles | **39.40** | 57.40 | **69.54** | 94.26 | **66.91** | 90.65 | **35.46** | 57.31 | **53.24** | 96.03 | **61.43** | 110.47 |
| Max of ensembles | 39.65 | 57.52 | 75.81 | 100.32 | 67.91 | 90.76 | 36.06 | 57.24 | 54.24 | 96.51 | 62.98 | 110.16 |

TABLE V: ETA prediction performance of models with/without data mining process

| Models | With data mining | | Without data mining | |
|---|---|---|---|---|
| | MAE (s) | RMSE (s) | MAE(s) | RMSE (s) |
| NNs with 1 hidden layer | 53.24 | 83.37 | 86.58 | 131.94 |
| DFNNs with 2 hidden layers | 51.02 | 81.91 | 78.54 | 124.75 |
| DFNNs with 3 hidden layers | 50.30 | 81.53 | 71.53 | 117.96 |
| DFNNs with 4 hidden layers | 50.40 | 81.34 | 71.35 | 119.17 |
| DFNNs with 5 hidden layers | 50.74 | 81.48 | 70.39 | 116.78 |
| **Ensembles of DFNNs with 3 hidden layers and DFNNs with 4 hidden layers** | | | | |
| Mean of ensembles | 50.05 | 81.01 | - | - |
| Min of ensembles | **49.83** | 81.20 | - | - |
| Max of ensembles | 50.89 | 81.66 | - | - |

for ETA prediction.

The proposed model consists of data mining and DFNNs. The data mining operation is capable of processing highly dense 4D trajectory data of all arrival flights in TMA and handling complicated traffic patterns. It even can be extended to other TMAs without manually tuning the hyperparameters. The experiments were firstly performed on real traffic data in Beijing TMA for 5 NNs models with nested cross validation. The results demonstrate that the deep neural networks perform better than shallow neural networks. However, attention should be paid on selecting the hidden layer numbers. Excessive number of hidden layers may cause overfitting. In addition, comparative study on data mining is conducted and proves that the data mining operation is robust in processing outliers, missing point and noise, which greatly improves the prediction accuracy by nearly 20 seconds in terms of MAE and over 30 seconds in terms of RMSE for each NNs model. Furthermore, we introduced ensemble learning model by combining well-performed individual models with certain strategies. The result proves that ensemble models outperform individual models by accuracy and stability. Overall, the minimum rule applied to DFNNs with 3 hidden layers and DFNNs with 4 hidden layers after data mining performs best in terms of MAE.

In our future work, we will introduce more nonlinear prediction models as individual models for ensemble learning, such as Support Vector Machine (SVM), Gradient Boosting Machine (GBM), etc. We will also examine our approach on other TMAs, which may provide with larger dataset on larger time scale or special scale.

## 6. Acknowledgement

The authors would like to thank Serge Roux for providing the data used in this study.

## References

[1] G. Enea and M. Porretta, "A comparison of 4D-trajectory operations envisioned for nextgen and sesar, some preliminary findings," in *28th Congress of the International Council of the Aeronautical Sciences*, 2012, pp. 23–28.

[2] Dart – data-driven aircraft trajectory prediction research. [Online]. Available: http://dart-research.eu/

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

[4] W. Kun and P. Wei, "A 4-d trajectory prediction model based on radar data," in *Control Conference, 2008. CCC 2008. 27th Chinese*. IEEE, 2008, pp. 591–594.

[5] A. de Leege, M. van Paassen, and M. Mulder, "A machine learning approach to trajectory prediction," in *AIAA Guidance, Navigation, and Control (GNC) Conference*, 2013, p. 4782.

[6] S. Hong and K. Lee, "Trajectory prediction for vectored area navigation arrivals," *Journal of Aerospace Information Systems*, vol. 12, no. 7, pp. 490–502, 2015.

[7] 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.

[8] M. G. Karlaftis and E. I. Vlahogianni, "Statistical methods versus neural networks in transportation research:

Differences, similarities and some insights," *Transportation Research Part C: Emerging Technologies*, vol. 19, no. 3, pp. 387–399, 2011.

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

[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] ——, "Machine learning and mass estimation methods for ground-based aircraft climb prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 6, pp. 3138–3149, 2015.

[12] Z. Wang, M. Liang, and D. Delahaye, "Short-term 4d trajectory prediction using machine learning methods," in *SID 2017, 7th SESAR Innovation Days*, 2017.

[13] Y. Bengio, Y. LeCun *et al.*, "Scaling learning algorithms towards ai," *Large-scale kernel machines*, vol. 34, no. 5, pp. 1–41, 2007.

[14] CAAC Operation Monitoring Center, "Report on national civil aviation flight operation efficiency in 2017 (chinese)," Tech. Rep., 2018.

[15] *Electronic Aeronautical Information Publication of People's Republic of China*, Aeronautical Information Center, ATMB, CAAC, 2017.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[17] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[18] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.

[19] S. Haykin, *Neural networks: a comprehensive foundation*. Prentice Hall PTR, 1994.