

Predicting Aircraft Trajectory Choice – A Nominal Route Approach

Yulin Liu, Mark Hansen

Institute of Transportation Studies
University of California, Berkeley
Berkeley, CA, USA
liuyulin101@berkeley.edu,
mhansen@ce.berkeley.edu

David J. Lovell

Department of Civil and
Environmental Engineering and
Institute for Systems Research
University of Maryland, College Park,
MD, USA
lovell@umd.edu

Michael O. Ball

Robert H. Smith School of Business
and Institute for Systems Research
University of Maryland, College Park,
MD, USA
mball@rsmith.umd.edu

Abstract — In this work, we propose a novel approach to predict aircraft trajectory choice. A trajectory clustering technique is used to consolidate historical flight tracks into a small set, and the cluster assignment results are then used as the ground truth of the route choice. Three types of features are used to predict the trajectory choice: convective weather, wind, and Miles-In-Trail (MIT) restrictions. Dimension of the features is greatly reduced by matching them with the representative trajectories of different clusters, which we call Nominal Routes. Four popular machine learning models are explored and compared: logistic regression, support vector machine, random forest, and gradient boosting. We apply our methods to five airport pairs: IAH to BOS, BOS to IAH, FLL to JFK, JFK to FLL, and LAX to SEA. The random forest approach has the best performance for all pairs except IAH to BOS, where gradient boosting has slightly better performance. Based on the best models, we rank the importance of features for different airport pairs. Results vary, but in general, wind has the largest effect, followed by thunderstorm, rain, and MIT.

Keywords – Aircraft Trajectory Prediction; Trajectory Clustering; Miles-In-Trail; Convective Weather; Random Forest; Gradient Boosting; Support Vector Machine; Logistic Regression

I. INTRODUCTION

With the emerging demand for air traffic, it is crucial to monitor and control air traffic flow to ensure the safety and efficiency of the National Airspace System (NAS). The Federal Aviation Administration (FAA) developed the Traffic Flow Management System (TFMS) to estimate sector traffic loads, make planning decisions, and evaluate historical performance. In the Europe, EUROCONTROL adopted the PREDICT system ([1]) to forecast the pre-tactical traffic load. While the vast majority of the decision-making processes (e.g., Miles-In-Trail restrictions) relies on traffic load estimation, the estimated demand is calculated mostly based on deterministic flight trajectory prediction processes, using, for example, filed flight plans or historical routes ([2], [3]). However, demand forecasting based upon deterministic processes barely considers uncertainties such as unsteady weather conditions, which could lead to overestimating the traffic load for sectors with bad weather and

induce unnecessary Traffic Management Initiatives (TMIs). Additionally, from a system point of view, sector demand is not only the result of planned flight routes, but also reroutes in response to TMIs and weather changes. Therefore, this paper aims to provide a trajectory prediction tool that can incorporate different sources of uncertainties – weather, wind, TMIs, etc., and further understand how those uncertainties affect the prediction results.

Current literature on aircraft trajectory prediction tools focuses mostly on predicting individual flight trajectories. Both deterministic and probabilistic approaches have been employed. The former approach (e.g., [4], [5], [6]) usually applies specific aerodynamic models and optimal control theory to estimate the state of an aircraft and then propagate the estimated states into the future. Although it accounts for specific aircraft parameters and kinematic equations, this approach, without considering any uncertainties such as unsteady weather conditions, can either predict only a specific phase of a flight or suffer from degraded prediction accuracy. Moreover, this approach cannot be used to predict trajectories prior to flight initiation. In contrast, the probabilistic approach relies on statistical tools to “learn” how aircraft fly from point A to point B from historical trajectory datasets. Ref. [7] trains a Generalized Linear Model (GLM) to use wind and aircraft initial state to predict the trajectories within the arrival terminal area. However, instead of predicting a 4D trajectory, the authors predict the time of arrival to each of a set of given fixed significant points (a.k.a., waypoints). Ref. [8] proposes a Hidden Markov Model (HMM) to predict full 4D trajectories, given the observed weather conditions (temperature and wind). In their model, the position of the aircraft is defined as the hidden state, and they assume that the weather cube observed around the track point is a realization of such hidden state. By training the HMM on a historical trajectory and weather dataset, the authors obtain the transition matrix and thus use it to predict any flight trajectories based on the observed or inferred weather sequence. Both models take the uncertainties of weather into account, but do not use any information about TMIs. More importantly, they only consider the local effect of weather, while in practice, flight routing accounts for a much

larger region of weather conditions. Therefore, the prediction error is typically large.

This paper proposes an approach that, instead of directly predicting individual flight tracks, predicts the aircraft route choices by consolidating a large number of trajectories into several clusters within which flight tracks are similar to each other. These predictions can support individual flight trajectory prediction tools by predicting the cluster to which a future flight trajectory will belong. Moreover, by identifying the relative importance of disparate factors in determining cluster assignment, it enhances our understanding of existing air traffic patterns and ability to understand how these factors affect en route efficiency. Marcos et al. ([9]) propose a similar framework that also applies trajectory clustering algorithm to predict aircraft route choice. However, their model utilizes a small set of features based on visual exploration, and does not incorporate uncertain factors such as convective weather and wind. As we will discuss further in section II, our approach is capable of efficiently utilizing a rich variety of features to predict route choices. Finally, our method establishes a baseline for predictive performance that may be improved upon in subsequent studies.

The rest of the paper is organized as follows. In section II, we summarize our data sources and feature engineering, which introduces the concept of choice set construction and nominal route. Section III describes the predictive models and model selection rules. In Section IV, we visualize the route choice sets for different airport pairs and present the prediction results. Section V offers conclusions, as well as suggestions for further research.

II. DATA AND FEATURE ENGINEERING

In this section, we first summarize the datasets used in the research, and then introduce how we construct the choice set, which induces the concept of a nominal route. Finally, we describe the process of extracting features for route choice prediction.

A. Data Sources

In this study, we use four datasets from different sources. The **flight tracks dataset**, which comes from the FAA Traffic Flow Management System (TFMS), contains 4D position – latitude, longitude, altitude, and time, whose resolutions are respectively 1 minute, 1 minute, 100 feet and 1 minute – of each aircraft throughout its flight. For this study, we obtained flight tracks for the 5 airport pairs listed in TABLE I. for the calendar years 2013, 2014, and 2015. During the preprocessing, we excluded tracks where spatial or temporal discontinuities were detected and ones that started or ended outside of the selected terminal areas.

The **convective weather dataset** was obtained from the National Oceanic and Atmospheric Administration (NOAA), and includes hourly summaries for convective weather

conditions like thunderstorm, rain, hail, etc. at approximately 2,500 ground stations in the US. Each record is a vector of binary variables indicating if there was a certain type of weather occurring at a specific time and location. For thunderstorms, it also indicates three levels of strength – light, medium and heavy. The collected data represented the years 2013 to 2015.

TABLE I. SUMMARY STATISTICS FOR FIVE AIRPORT PAIRS

Airport Pair	Number of tracks after preprocessing		
	2013	2014	2015
IAH → BOS	1679	1710	1673
BOS → IAH	1742	1782	1698
FLL → JFK	4010	3847	4515
JFK → FLL	4042	3705	4567
LAX → SEA	6991	8066	8964

The **wind dataset** used in this work comes from the National Center for Atmospheric Research (NCAR). It records the southerly and westerly wind speeds four times a day – 0:00, 6:00, 12:00, and 18:00 UTC. For each instance, the wind speed is given at a 3D grid with $2.5^\circ \times 2.5^\circ$ latitude/ longitude resolution, and 17 pressure levels ranging from 10 mbar to 1000 mbar ([10], [11]). By collecting the wind data from CY 2013 to 2015, we obtained 4,380 raster files.

The **Miles-In-Trail (MIT) dataset** comes from the National Traffic Management Log (NTML), and contains information about where, when, and why MIT restrictions were implemented. In the US, the MIT specifies the spacing required between aircraft flying along a certain path at an altitude range [12]. Fields of interest include providing/ requesting facility, NAS element, start/end time, altitude, and the actual MIT spacing parameter. For this study, we excluded restrictions that were cancelled before initiation and those whose distance thresholds were less than 0, and we also merged MITs that were extended or modified from the initial MIT. Lastly, we only included restrictions with known facility and NAS element geometries. TABLE II. provides a few summary statistics about MIT restrictions from CY 2013 to 2015.

TABLE II. SUMMARY STATISTICS FOR MIT FROM 2013 TO 2015

	TOTAL	ENROUTE
Cancelled Before Initiation Rate	5.48%	5.24%
Number Initiated	543,643	231,298
Total Number of MIT after preprocessing	465,966	203,457
Average Duration/ hours	2.04	1.82
Average MIT Value/ miles	18.79	20.59

B. Constructing the Route Choice Set

To fulfill the goal of modeling the aircraft trajectory choice, we first need to construct the route choice set for a given Origin - Destination (OD) pair. One approach would be to treat every historical route as one of the alternatives. However, this is computationally intractable due to the volume of the traffic and challenges of computing the features for each member of such a large choice set. Moreover, it is difficult to validate such a

model. We therefore opt for a simpler approach. We reduce the choice set by grouping similar flight trajectories and use the cluster assignment of a given flight as the route choice, or label. To implement this idea, we must apply a trajectory clustering algorithm to classify flight tracks. A brief summary of the algorithm is shown in Algorithm 1. Interested readers may refer to [13] for the details of the algorithms.

Algorithm 1. Trajectory Clustering (TC)
INPUT Flight track dataset FT .
PARAMETER Maximal distance threshold eps in DBSCAN algorithm Minimal samples threshold $Msamp$ in DBSCAN algorithm
OUTPUT Number of Clusters, cluster assignment, and nominal routes
Step 1 For every flight, linearly interpolate 100 points with equal distance along the route traveled.
Step 2 Apply Principal Component Analysis (PCA) on all the interpolated trajectory data, and get the first five components as the feature vector.
Step 3 Apply DBSCAN($eps, Msamp$) algorithm [14] to cluster similar trajectories.
Step 4 For each cluster, solve a 1-median problem to get the nominal route.

a. The use case for step 4 is described in section II.C.

C. Framework of Feature Engineering

From section II.B, we have constructed a set clusters, and know the cluster assignment for any given flight in our data set. To predict the route choice, four types of predictors are used: the presence of weather conditions at the 2,500 US ground stations, wind information, MIT restrictions, and flight features such as operating airline and departure season. The description of variables is summarized in TABLE III.

Before we build the predictive models, it is crucial for us to convert the underlying data into features that provide the basis for the subsequent modelling. One naïve alternative would be for each flight, to use the full dataset of predictors covered by the flight time period. However, this runs the danger of overfitting. For example, a normal flight from IAH to BOS takes around three hours of airborne time. Within the time period, the dimensions for the convective weather alone would be approximately $2482 \times 3 \times 7 \approx 50,000$, where 2,482 is the number of weather stations, 3 the time instances (a 3-hour flight matches roughly 3-hour weather instances), and 7 the number of weather phenomena. However, the total number of flights from IAH to BOS is 5062, which is much fewer than the dimensions of the naïve features. Therefore, we need to construct highly descriptive features that not only represent the conditions in the relevant airspace, but with controllable dimensions.

To tackle the dimensionality issue, we introduce the concept of *Nominal Route*. Among the trajectories that form a given cluster, one can identify a specific trajectory that is the “center” of the cluster (see step 4 in Algorithm 1), which we call the nominal route. Such routes are highly representative for the associated clusters so that we can use them as the bases for determining cluster features (e.g., exposure to thunderstorms) that help predict to which cluster a given flight belongs.

Moreover, we assume cluster assignments of individual flights depend on nominal route-specific features, such as convective weather and wind. To determine these features for a specific flight, we fix its departure time at its actual value, which we assume would be the same regardless of its flight route. Based on these two assumptions, we determine the features for each designated flight by concatenating route-specific features for all nominal routes. The general framework of the feature engineering is described by Algorithm 2.

Algorithm 2. General Framework for Feature Engineering (FE)
INPUT Trajectory of a designated flight T_{flight} . Trajectories of a set of nominal routes $NR = \{NR_1, NR_2, \dots\}$, where $NR_i = \{p_1, p_2, \dots\}$ and p_i is a 4-dimensional point constituted of latitude, longitude, altitude and time.
OUTPUT Mapped features of the set of nominal routes with replaced time stamps
Step 1 Get the departure time DT and flight specific features FSF – airline and departure season – from T_{flight} .
Step 2 For each nominal route NR_i , (a) replace its departure time with DT ; (b) calculate the subsequent time stamps based on the new departure time DT and NR_i .
Step 3 Call Algorithm 3, Algorithm 4 and Algorithm 5 to construct route specific features for each nominal route.

D. Constructing Features for Nominal Routes

There are three types of route specific features – convective weather, wind and MIT, each of which has different data format and construction rules. The convective weather records obtained from NOAA are recorded at individual weather stations, and are represented as time-stamped vectors of the form $[I_t, I_r, I_s, I_h, I_i, I_p, I_{sh}]$, where the binary indicator variables $\{I_j\}$ represent the presence or absence of thunderstorms, rain, squalls, hail, ice, precipitation, and showers, respectively. Moreover, in the original dataset, thunderstorm is marked as $TS-, TS, TS+$, which stands for light, medium and heavy thunderstorms, respectively. Therefore, we also use a variable $I_t^s = [0.5, 1, 1.5]$ to specify the strength of the thunderstorm (a smaller value is for lighter conditions). For a given flight and a set of given nominal routes, we first use steps 1 and 2 of Algorithm 2, and then compute the weather-related metrics using Algorithm 3. At a high level, the algorithms compute a weighted average of each binary weather variable at each station within 150 nautical miles of a given point on the route based on the weather at the nearest time stamp to when the flight would have traversed that point, where the weight is the inverse of the distance between the station and the point.

The wind data from NCAR are stored as raster files. Each raster file corresponds to a single snapshot of wind vectors $WD = [U_{ijk}, V_{ijk}]$ at a specific time, where U_{ijk} and V_{ijk} are, respectively, the westerly and southerly wind speed components at the grid point $(lat, lon, alt) = (i, j, k)$. Similar to Algorithm 3, we construct the wind features for transformed nominal routes using a tree-based search technique. To summarize, this

algorithm matches each track point on the route with both spatially and temporally closest wind speed, and uses it to calculate the equivalent still air distance and average wind speed for each nominal route. The details are described in Algorithm 4.

Algorithm 3. Mapping Convective Weather (MCWX)

INPUT Trajectory of a nominal route with replaced time stamps $NR' = \{p'_1, p'_2, \dots\}$. Note that only the time dimension of NR' is different from NR . A tensor of convective weather data $WX = [WX_{tjk}]$, where WX_{tjk} represents the binary indicator for weather phenomenon k^c of station j^a at time t .
PARAMETER Maximum query distance Max_Dist (default: 150 nautical miles)
OUTPUT Weather-related features for NR' .
Step 1 Construct two kd-trees [15] for spatial (s_tree_wx) and temporal (t_tree_wx) dimensions, respectively. The spatial tree is built based on the location of the weather stations. Step 2 Query s_tree_wx to find all stations within a circle with radius of Max_Dist for each track point p'_i of NR' . Step 3 Query t_tree_wx to find the nearest time stamp in the weather data to NR' . Step 4 Combine query results from steps 2 and 3, and get a list of matched weather variables MI_{ik} , where k represents the weather phenomenon, and a list of distances away from matched weather stations (MSD_i) to each track point p'_i . Step 5 Compute the weighted average MI_{ik} and get weather scalar MW_{ik} for track point p'_i , where the weights are a decreasing function MSD_i and sum to 1. (Currently the function is $1/MSD_i$) Step 6 Average MW_{ik} along the nominal route NR' and get the feature metric Wx_k for weather phenomenon k . Step 7 Return Wx , where Wx is a list of Wx_k and Wx_k is the weather metric for weather activity k .

a. A station is represented by (latitude, longitude).

b. For the purpose of computational efficiency, we set a base time and use the difference between the base time and time stamps in both the weather and trajectory data to construct and query the kd-trees. This also applies to the subsequent algorithms.

c. k includes thunderstorm, thunderstorm level, rain, squall, hail, ice, precipitation and shower.

The miles-in-trail data from NTML are recorded as $MIT = \{st, et, reqfac, provfac, nas_elem, alt, mit_value\}$. It can be interpreted as facility $reqfac$ (e.g., ZLA) requests a MIT to manage the area's traffic, so that facility $provfac$ (e.g., ZOA) provides a MIT restriction enforced at the NAS element (e.g., BTY) with value mit_value (e.g., 15 miles) and altitude range specified by alt from time st to time et . Note that the NAS element can be a jet route, TRACON or fix; therefore, each MIT restriction considered must be assigned to an influence area. In this study, we must assess the airspace regions within which flight traffic is impacted by:

1) If the NAS element is a fix, then the influence area we use is a circle of radius R_{fix} around the fix. A trajectory is considered to be potentially influenced by this MIT if it intersects this circle.

2) If the NAS element is a jet route, then we define a swath centered on the jet route with overall width D_{jet} . A trajectory is

considered to be potentially influenced by this MIT if it intersects this swath for at least a "distance" L_{jet} .

3) If the NAS element is a center/TRACON, we use its existing geometry. A trajectory is considered to be potentially influenced by this MIT if it intersects this polygon.

Algorithm 4 Mapping Wind (MWD)

INPUT Trajectory of a nominal route with replaced time stamps $NR' = \{p'_1, p'_2, \dots\}$. Note that only the time dimension of NR' is different from NR . A tensor of wind data $Wind = [U_{ijkt}, V_{ijkt}]$, where U_{ijkt} and V_{ijkt} respectively represent the westerly and southerly wind speeds at the grid $(i, j, k)^a$ and time t .
OUTPUT Wind-related features for NR' .
Step 1 Construct two kd-trees for spatial (s_tree_wind) and temporal (t_tree_wind) dimensions, respectively. The spatial tree is built based on the horizontal grids. Step 2 Find the closest altitude level from the 17 standard pressure altitudes in the wind data to each track point p'_i of NR' . Step 3 Query s_tree_wind to find the 2D grid point nearest to each track point p'_i of NR' . Step 4 Query t_tree_wind to find the time stamp of the wind data nearest to NR' . Step 5 Combine query results from steps 2, 3 and 4, and get the matched westerly and southerly wind speeds (u_i and v_i) for each track point p'_i . Step 6 For each track point p'_i , calculate the azimuth θ with respect to the previous adjacent point, then calculate the headwind/tailwind by $HTW_i = u_i \cos \theta + v_i \sin \theta^c$. Calculate the average of HTW_i along the trajectory (mean wind speed $MWSP$). Step 7 Use the ground speed in the flight track data and headwind/tailwind speed to calculate the airspeed, then calculate the distance traveled for NR' with respect to the air (equivalent still air distance WD). Step 8 Return WD and $MWSP$.

a. A grid is represented by (latitude, longitude, pressure altitude).

b. The temporal tree is constructed in the same way as in Algorithm 3.

c. If the HTW is negative, then it represents headwind speed.

4) If the altitude of a MIT is specified, then we define an altitude buffer B_{alt} . A trajectory is considered to be potentially influenced by this MIT if its crossing altitude is within the altitude range of $alt \pm B_{alt}$.

The detailed feature construction algorithm is described in Algorithm 5. Different from Algorithms 3 and 4, this algorithm intersects each route with all MIT geometries and finds the one(s) that are both spatially and temporally matched. The relevant features of the MIT include the number of matched MIT restrictions, average of matched MIT durations, values, and stringency.

Algorithm 5. Mapping MIT (MMIT)

INPUT Trajectory of a nominal route with replaced time stamps $NR' = \{p'_1, p'_2, \dots\}$. Note that only the time dimension of NR' is different from NR . A matrix of MIT dataset $MIT = \{st, et, reqfac, provfac, nas_elem, alt, mit_value\}$ where each column is a 1-dimensional vector.
PARAMETER Radius of a circle around a fix: R_{fix} (default: 0.25-degree of latitude/longitude. See Ref. [16] for further information)

III. PREDICTIVE MODELS AND MODEL SELECTION

A. Candidate Models

To model the aircraft route choice, we compare the performance of four machine learning algorithms: Logistic Regression (LR), Support Vector Machine (SVM), and two tree-based ensemble methods – Random Forest (RF) and Gradient Boosting (GB).

Logistic regression assumes that the Logit transformation of the probability of each choice has a linear relationship with the predictors. By maximizing the likelihood of the outcome in the training dataset, LR learns the probabilistic assignment of each alternative given the feature vector. SVM tries to find, instead of the probabilistic distribution of the data, the separating hyperplane that maximizes the margins of different classes [17]. In this study, we also apply a Gaussian Radial Basis Function (RBF) kernel in the SVM algorithm to further capture nonlinearities. Random forest and gradient boosting are both based on decision trees, which map the feature vector to the target choice in the leaves of the tree by splitting the data recursively [18]. While decision trees are inherently indifferent to nonlinear features and have low bias, the variance is prone to be high. Therefore, to overcome the issue of high variance, random forest and gradient boosting algorithms are proposed. While the RF learns a cluster of fully grown decision trees (usually shallow and using only a random subset of the features) and use the average of the output for all trees, GB starts with a weak decision tree (e.g., shallow trees) and sequentially learns a set of trees that reduce the error (bias) of the previous trees [19], [20], [21].

B. Model Selections and Experimental Setup

As addressed in the previous section, the conflict between bias and variance makes it crucial to properly tune the hyper parameters for the prediction models. While all four algorithms have an extensive number of hyper parameters to tune, we select the ones that are commonly used in the open literature. The notations and parameter grids are summarized in TABLE IV.

Selecting hyper parameters for different models involves deciding appropriate model performance metrics. Although prediction accuracy is an important metric in this study, our data in general are heavily imbalanced so that using accuracy alone is no longer reliable. For example, for the pair LAX → SEA, one of the classes has over 75% of samples, therefore, a naive model that always predicts this class will yield over 75% accuracy. Thus, in this study, we use the unweighted average F1 score, which is the average of F1 scores across all classes without considering the class weights, as the performance metric to select the best hyperparameters and the best model. The F1 score is twice the harmonic mean of the precision—the ratio of true positives to predicted positives—and the recall—the ratio of true positives to actual positives. To be more specific, for each city pair, we first randomly split the full dataset into a training set (80%) and testing set (20%). Second, for each candidate model, we use a 3-fold cross-validation over the training set to

Algorithm 5. Mapping MIT (MMIT)

Width of a swath surrounding a jet route: D_{jet} (default: 0.5-degree of latitude/ longitude)
 Minimal length of within a intersected swath: L_{jet} (default: 1-degree of latitude/ longitude)
 Altitude buffer: B_{alt} (default: 1000 ft.)

OUTPUT

MIT-related metric for NR' .

Step 1 Convert NR' to a "LineString" geometry object. Convert the $reqfac$ and $provfac$ to polygon geometry objects. Define influencing area based on nas_elem and parameter R_{fix}, D_{jet} and L_{jet} , and convert them into corresponding geometry object.

Step 2 Filter MITs whose et is before the departure time of NR' and whose st is after the arrival time of NR' . The remaining MIT restrictions are retained as a candidate.

Step 3 Intersect NR' with every candidate MIT. If NR' intersects the MIT_i 's influencing area, $reqfac_i$ and $provfac_i$ in a correct order, and the time of intersecting with influencing area is within $[st_i, et_i]$, then MIT_i will go to Step 4, otherwise it will be discarded.

Step 4 For the remaining MITs, if the altitude restrictions with buffer B_{alt} are satisfied at the intersection point with the influencing area, then we gather the list of such MITs as $VMIT$ and pass it to step 5.

Step 5 Count the number of matched MITs in $VMIT$ as $CMIT$, and Calculate average of the MIT value, MIT duration, MIT stringencies^a in $VMIT$ as $MMIT_val$, $MMIT_dur$ and $MMIT_str$.

Step 6 Return $CMIT$, $MMIT_val$, $MMIT_dur$ and $MMIT_str$.

a. The MIT stringency is defined as the product of MIT hour and MIT value.

Through Algorithm 2 to Algorithm 5, the feature space for each flight includes a vector of features for different nominal routes as in the format of $F = [F_0, F_1, F_2, \dots, F_N]$, where F_i is the feature vector (e.g., $[CMIT, WD, \dots]$) for nominal route i with transformed time stamps. The description of the full feature vector for any flight is summarized in TABLE III.

TABLE III. DESCRIPTION OF FEATURES

Category	Notation	Description
Flight level generic variable	<i>Airline</i>	A list of dummy variables. Each category is a major airline that covers at least 1% of the total traffic.
	<i>LocalHour</i>	Binary variable. Equals 1 if the departure time is before 12 pm local time
	<i>Season</i>	A list of dummy variables. Categorized by the departure month: winter (Dec - Feb), spring (Mar - May), summer (Jun - Aug), fall (Sep - Nov)
Convective weather	<i>TS</i>	Average of the thunderstorm
	<i>TSLevel</i>	Average of the thunderstorm level
	<i>R</i>	Average of the rain
	<i>S</i>	Average of the shower
	<i>H</i>	Average of the hail
	<i>I</i>	Average of the ice
	<i>P</i>	Average of the precipitation
Wind	<i>SH</i>	Average of the shower
	<i>WD</i>	Wind distance
	<i>MWSP</i>	Average of wind speed
MIT	<i>CMIT</i>	Number of MIT matched
	<i>MMIT_val</i>	Average of MIT value
	<i>MMIT_dur</i>	Average of MIT duration
	<i>MMIT_str</i>	Average of MIT stringency

perform exhaustive search over the parameter grid to select the best parameter(s). To estimate the 3-fold cross-validation average F1 score, we randomly partition the training set into three subsets, and repeatedly use two sets for training and one set for calculating average F1 score three times. The final F1 score is the average of three F1 scores over different subsets. Lastly, we use the selected hyperparameters to train all four models on the full training set, and evaluate them on the testing set. The model with the highest average F1 score was selected for further analysis.

TABLE IV. HYPER PARAMETERS FOR PREDICTIVE MODELS

Model	Notation	Description	Parameter grid
Logistic Regression	C	Penalty term for the l_2 regularization	$C = [0.1, 1, 10, 25, 100, 150]$
SVM	γ	Kernal coefficient for RBF	$\gamma = [0.01, 0.1, 1, 10]$
	C	Penalty term for error term	$C = [1, 10, 100, 150]$
Random Forest	D	Maximal depth of the trees	$D = [7, 9, 11, 15, 19, 27]$
	N	Number of trees	$N = 300$
Gradient Boosting	D	Maximal depth of the trees	$D = [7, 11, 15, 19, 27]$
	LR	Learning rate	$LR = [0.05, 0.1, 1]$
	S	Sampling rate of the training data to learn each tree	$S = [0.25, 0.75, 1]$
	N	Number of trees	$N = 300$

IV. RESULTS

In this section, we first present the results of trajectory clustering (route choice set) for the specified five OD pairs. Then we briefly summarize the performance metrics for different models with the best hyperparameters. Lastly, we use the selected model to estimate the importance of different features.

TABLE V. CLUSTERING STATISTICS

Cluster Color index	OD pair				
	IAH BOS	BOS IAH	FLL JFK	JFK FLL	LAX SEA
Red	34.12%	60.37%	83.30%	79.97%	13.75%
Green	22.55%	5.46%	14.20%	11.49%	75.51%
Magenta	35.76%	19.18%	0.82%	3.03%	2.39%
Cyan	0.55%	2.64%	-	1.00%	-
Blue	0.53%	1.26%	-	0.23%	-
Black	6.46%	11.06%	1.67%	4.28%	8.35%

A. Trajectory Clustering Results

Figure 1. show the clustering results for the OD pairs listed in TABLE I. Each color represents a cluster, within which flight trajectories are similar to each other. The black curves, however, are flight trajectories classified as “outliers”, which do not belong to any of the natural clusters. The nominal routes for natural clusters are marked as white solid curves in the figures. The traffic shares of different clusters are summarized in TABLE V. Obviously, the flights from IAH to BOS have a more

balanced cluster distribution, while the other four pairs always have one dominant cluster – usually the most direct path – that covers over 60% of traffic.

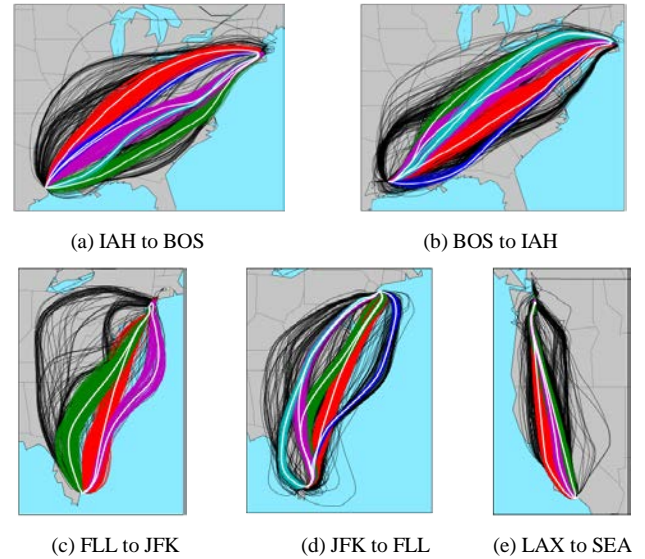


Figure 1. Clustering Results

B. Model Selection and Performance Statistics

We select the best hyperparameters based on the average F1 scores over 3-fold cross validation on the 80% training set. And we further compare models with the best parameters based on their performance in the 20% testing set. The best hyperparameters and the corresponding F1 scores on the testing set are recorded in TABLE VI. , where Param. records the best parameter(s) selected for the associated model. Since there are multiple classes for each OD pair, each model will have a F1 score for every alternative. Therefore, F1 max, F1 min and F1 mean are, respectively, the maximal, minimal, and average F1 scores across all alternatives in the testing set.

Generally speaking, the maximal F1 scores across different models are comparable for a specific OD pair, indicating that all four models perform considerably well in terms of precision and recall for at least one class – usually the one with the highest frequency. However, the minimal F1 scores differ significantly across models. Random forest and logistic regression in general perform better than the other two models, suggesting that RF and LR are more robust to overfitting. Lastly, random forest outperforms the other three models with respect to average F1 score for all pairs except the case from IAH to BOS, where the gradient boosting algorithm is slightly better. While pairs other than IAH to BOS do have more imbalanced class distributions, random forest applies bootstrapping and oversampling techniques to overcome this issue, which explains its overall improved performance.

TABLE VI. MODEL PERFORMANCE

Model		OD pair				
		IAH BOS	BOS IAH	FLL JFK	JFK FLL	LAX SEA
LR	Param.	C=0.1	C=1	C=150	C=0.1	C=0.1
	F1 max	0.53	0.73	0.88	0.87	0.81
	F1 min	0.00	0.11	0.07	0.00	0.18
	F1 mean	0.28	0.32	0.34	0.25	0.38
SVM	Param.	C=100 $\gamma=0.01$	C=100 $\gamma=0.01$	C=100 $\gamma=0.01$	C=150 $\gamma=0.01$	C=100 $\gamma=1$
	F1 max	0.54	0.74	0.91	0.89	0.83
	F1 min	0.00	0.16	0.00	0.00	0.09
	F1 mean	0.27	0.30	0.34	0.24	0.31
RF	Param.	D=9	D=9	D=11	D=11	D=15
	F1 max	0.52	0.69	0.88	0.86	0.79
	F1 min	0.00	0.15	0.08	0.12	0.16
	F1 mean	0.28	0.37	0.37	0.37	0.39
GB	Param.	D=15 S=0.75 LR=0.05	D=25 S=0.75 LR=0.1	D=25 S=1 LR=1	D=19 S=1 LR=1	D=25 S=1 LR=1
	F1 max	0.57	0.76	0.91	0.88	0.86
	F1 min	0.00	0.00	0.00	0.00	0.10
	F1 mean	0.29	0.30	0.35	0.27	0.37

C. Importance of Features

Feature importance is a metric that helps selecting relevant features as well as understanding the contributions of different predictors. For SVM and logistic regression, it is straightforward to use the estimated weights or coefficients ([22]) to evaluate the importance of features. For tree-based algorithms, however, since there are no weight coefficients, estimating the feature importance is based on the change of prediction error (or impurity) when omitting designated feature(s) ([23]). In this study, we use the best model for each OD pair (e.g., gradient boosting for IAH to BOS) to evaluate the importance of features for the variables listed in TABLE III. Notice that variables other than flight level generic variables, such as *ice* and *rain*, are N dimensional, where N is the number of nominal routes, so that we sum the N dimensional feature importance for each of those variables. Furthermore, variables like *WD* and *MWSP* are highly correlated in a way that they both characterize the wind effect to the route choice; therefore, we sum the importance for those correlated variables and use the summation as their feature importance. To be more specific, feature importance for thunderstorm is the summation of *TS* and *TSLevel*; for MIT it is the summation of *CMIT*, *MMIT_val*, *MMIT_dur*, *MMIT_str*; and for wind it is the summation of *WD* and *MWSP*.

Figure 2. shows the feature importance for 12 different features. Each color represents a OD pair, and the heights of the bars indicate the relative importance of the features. The rank of the estimates in general matches our expectation. Wind has the highest impact on the route choice for all five pairs. Thunderstorm and rain exhibit comparable and large effects for most of the pairs. Other meteorological conditions, such as hail and ice, seem to have negligible impacts. Among non-

meteorological factors, MIT has the greatest importance for pairs except LAX to SEA, which might because very few nominal routes (with transformed time stamps) are actually matched up with the MIT data (less than 2%). Thirdly, the airline variable has very limited effect to IAH to BOS and BOS to IAH, since over 95% of traffic between the two comes from United Airlines. However, as the market becomes less concentrated, the airline effect increases. For example, the market for LAX to SEA is composed of 52% Alaska Airlines flights, 14% Virgin America, 13% SkyWest and 12% Delta flights, and in this case the airline effect has the third highest importance. Lastly, the variable *LocalHour* and *Season* both have low effects, which could be largely absorbed by the weather variables.

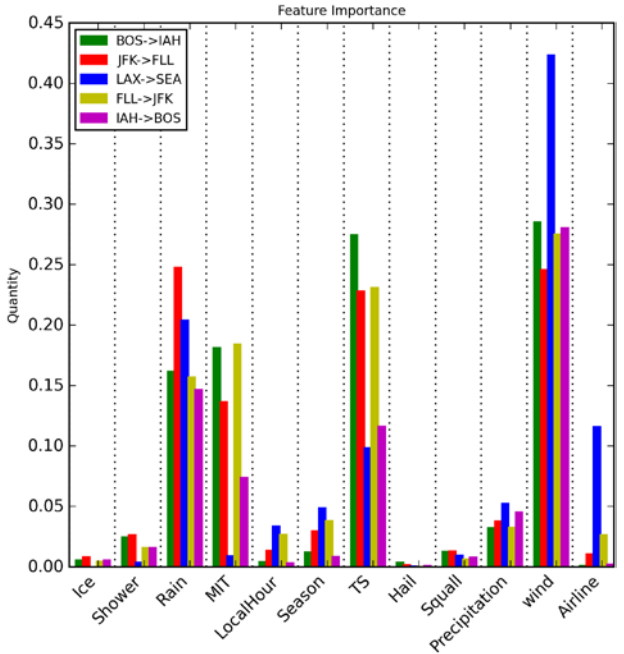


Figure 2. Feature Importance

V. CONCLUSIONS AND FUTURE RESEARCH

This research proposes a novel approach to predict the aircraft route choice and understand the relative importance of convective weather, wind and miles-in-trail restrictions. The method is based on trajectory clustering in which geographically similar trajectories are grouped into clusters that are represented by the “centers” of the groups, which we call *Nominal Routes*. By using the trajectory cluster assignment as the ground truth of route choice, we are able to reduce the dimension of the choice set significantly. Furthermore, we demonstrate several practical algorithms that efficiently construct highly descriptive features with significantly lower dimensions for each flight by matching the nominal routes with different data sources (e.g., convective weather). Via the constructed features, we trained four popular

machine learning models – logistic regression, support vector machine, random forest, and gradient boosting – to predict the aircraft trajectory choice.

Using this approach, we first compared four models (with parameters fine-tuned) on the testing set (20% of the full data) using the average F1 score. For the five city pairs considered in this study, random forest outperforms the other three methods for all pairs except IAH to BOS, for which gradient boosting has the best performance. For each city pair, we then use the best model to estimate the importance of different features. Results vary across OD pairs but in general, wind has the greatest importance, followed by thunderstorm and rain. MIT has moderately high importance for all pairs except LAX to SEA, where not many flights are matched with MIT restrictions. Airline has increasing importance as the market gets less concentrated.

In addition to the merits of our approach, we point out several potential applications and future extensions. First of all, by predicting the route choice for future flights where weather forecasts are available, we are able to predict sector demand and analyze the en route performance, for example en route efficiency. Furthermore, for pairs where MIT plays an important role, our method helps decision makers to plan traffic management initiatives to manage flows more efficiently. Secondly, our predictive models in general perform well on classes with moderately high weights, however, their prediction precisions on least chosen classes are low. While this is mostly due to the imbalanced nature of the data, one could further increase the sampling rate for the least weight class. Finally, our approach can be easily adapted to more features, including other traffic management measures, such as Airspace Flow Programs (AFP) and Special Use Airspace (SUA) activations. Also, given the importance of convective weather revealed by our study, it would also be desirable to incorporate higher fidelity convective weather information, such as the national convective weather forecast (NCWF), into our analysis.

ACKNOWLEDGMENT

This work was supported by the FAA through the NEXTOR-II Consortium.

REFERENCES

- [1] EUROCONTROL, "PREDICT," EUROCONTROL, [Online]. Available: <http://www.eurocontrol.int/articles/ddr-pre-tactical-traffic-forecast>. [Accessed 9 5 2018].
- [2] T. Mueller, J. Sorensen and G. Couluris, "Strategic aircraft trajectory prediction uncertainty and statistical sector traffic load modeling," in AIAA Guidance, Navigation, and Control Conference and Exhibit, Monterey, CA, 2002.
- [3] Federal Aviation Administration, "Traffic Flow Management System (TFMS) -- Improving Demand Predictions (IDP)," Department of Transportation, 2015.
- [4] G. Avanzini, "Frenet-based algorithm for trajectory prediction," *Journal of guidance, control, and dynamics*, vol. 27, no. 1, pp. 127-135, 2004.
- [5] G. B. Chatterji, "Short-term trajectory prediction methods," in AIAA Guidance, Navigation, and Control Conference, 1999.
- [6] 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.
- [7] A. de Leege, M. Van Paassen and M. Mulder, "A machine learning approach to trajectory prediction," in AIAA GNC Conference and Exhibit, Boston, MA, 2013.
- [8] S. Ayhan and H. Samet, "Aircraft Trajectory Prediction Made Easy with Predictive Analytics," in 22nd International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, 2016.
- [9] R. Marcos, O. G. C. Ros and R. Herranz, "Combining Visual Analytics and Machine Learning for Route Choice Prediction," in 7th SESAR Innovation Days, Beograd, 2017.
- [10] E. Kalnay, M. Kanamitsu, R. Kistler, W. Collins, D. Deaven, L. Gandin, M. Iredell, S. Saha, G. White, J. Woollen and others, "The NCEP/NCAR 40-year reanalysis project," *Bulletin of the American meteorological Society*, vol. 77, no. 3, pp. 437-471, 1996.
- [11] F. Cheng and J. Gulding, "Computing Wind-Optimal Routes for Flight Performance Benchmarking," in 16th AIAA Aviation Technology, Integration, and Operations Conference, Washington, D.C, 2016.
- [12] K. Sheth, S. Gutierrez-Nolasco and J. Petersen, "Analysis and modeling of miles-in-trail restrictions in the national airspace system," in 2013 Aviation Technology, Integration, and Operations Conference, Los Angeles, CA, 2013.
- [13] Y. Liu, M. Hansen, D. Lovell, C. Chuang, M. Ball and J. Gulding, "Causal Analysis of En Route Flight Inefficiency--the US Experience," in 12th USA/Europe Air Traffic Management Research and Development Seminar (ATM 2017), Seattle, WA, USA, 2017.
- [14] M. Ester, H.-P. Kriegel, J. Sander and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining, 1996.
- [15] J. L. Bentley, "K-d trees for semidynamic point sets," in Proceedings of the sixth annual symposium on Computational geometry, 1990.
- [16] T. A. Myers, M. Klopfenstein, J. Mintzer and G. Wilmouth, "A preliminary analysis of the impact of miles-in-trail restrictions on NAS flight operations," in 6th USA/Europe Air Traffic Management Research and Development Seminar (ATM2005), Baltimore, MD, USA, 2005.
- [17] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273-297, 1995.
- [18] L. Breiman, J. Friedman, C. J. Stone and R. A. Olshen, *Classification and regression trees*, Monterey, CA: CRC press, 1984.
- [19] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5-32, 2001.
- [20] J. H. Friedman, "Stochastic gradient boosting," *Computational Statistics & Data Analysis*, vol. 38, no. 4, pp. 367-378, 2002.
- [21] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of statistics*, pp. 1189-1232, 2001.
- [22] A. Rakotomamonjy, "Variable selection using SVM-based criteria," *Journal of machine learning research*, vol. 3, pp. 1357-1370, 2003.
- [23] A. Liaw and M. Wiener, "Classification and regression by randomForest," *R news*, vol. 2, no. 3, pp. 18-22, 2002.