Title

Trajectory Modeling in the National Airspace System: Data Mining, Inference, and Deep Generative Models

Author

Liu, Yulin

Publication Date

2020

Trajectory Modeling in the National Airspace System: Data Mining, Inference, and Deep Generative Models

by

Yulin Liu

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Engineering – Civil and Environmental Engineering

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Mark Hansen, Chair
Professor Michael J. Cassidy
Professor Peng Ding

Spring 2020

Trajectory Modeling in the National Airspace System: Data Mining, Inference, and Deep Generative Models

Abstract

Trajectory Modeling in the National Airspace System: Data Mining, Inference, and Deep Generative Models

by

Yulin Liu

Doctor of Philosophy in Engineering – Civil and Environmental Engineering

University of California, Berkeley

Professor Mark Hansen, Chair

In this thesis, we present data-driven methods and algorithms to address the emerging challenges for performance assessment and predictive analysis in the national airspace system (NAS). Among these challenges, routing efficiency has received more attention in the research community due to its core position in aviation economics and environmental studies. This motivates us to seek answers to fundamental questions such as “what are the mechanisms and causes of inefficient en route operations?” More importantly, “what can we do to improve these operations? Presumably in a predictive way?”

Chapter 2 provides a macroscopic comparison for flight en route inefficiency among different departing airports, arrival airports, seasons, and flight lengths through a series of fixed-effects models. We show that flights from or to the airports in the east coast corridor and south coast, especially in the New York metropolitan and Florida area, are generally more inefficient than the others. Long-haul flights are more efficient mainly because one component of excess distance – which results from flights using fixed entry and exit points that are not on the great circle route – is roughly independent of distance. In addition, flights operated during the summer season, when convective weather is more frequent, are less efficient than those in other seasons.

In Chapter 3, we have first gathered information from the Federal Aviation Administration (FAA) database system, which includes convective weather, wind, miles-in-trail (MIT) restrictions, airspace flow programs (AFP), and special activity airspace (SAA), through data mining techniques. We then propose two mechanisms to ascribe flight en route inefficiency to these factors. Trajectory clustering, attribute matching, statistical modeling, and counterfactual analysis have been employed in our methodological framework. Using this approach, the contributions of wind, connective weather, MIT, AFP and SAA to flight inefficiency have been estimated. Results vary across airport pairs, but in general if we systematically clear the values of these causal factors in the whole airspace, convective weather and wind

make the greatest, yet not dramatic, contribution. However, we have also shown that the "marginal contribution estimates" might be caused by the largely homogeneous distribution of different attributes across routes. Therefore, if we "clear the sky" solely for one alternative route, flight en route inefficiency may substantially improve or deteriorate.

Finally, we have developed in Chapter 4 a novel approach to predict, in a real-time manner, the actual 4D (latitude, longitude, altitude, time) aircraft trajectories. Specifically, our framework consists of an efficient tree-based matching algorithm to construct feature maps from high-fidelity meteorological datasets, and an end-to-end convolutional recurrent neural network that includes a long short-term memory (LSTM) encoder network and a mixture density LSTM decoder network. The encoder network embeds last-filed flight plan information into fixed-size hidden state variables and feeds the decoder network, which further learns the spatiotemporal correlations from the historical flight tracks. Convolutional layers are integrated into the pipeline to learn representations from the high-dimension weather features. During the real-time inference process, beam search, adaptive Kalman filter, and Rauch-Tung-Striebel smoother algorithms have been implemented to prune the variance of generated trajectories. Our approach, which is trained on historical flights from the George Bush intercontinental airport (IAH) to Boston Logan international airport (BOS) in 2013, achieves an average of 50 nautical-mile horizontal and 2800 ft. vertical error, which is a new state-of-art method for trajectory prediction tasks.

Dedicated to my fiancée Shu Huang and my parents Yang and Hui.

Contents

Contents **ii**

List of Figures **iv**

List of Tables **vi**

1 Introduction **1**

2 Performance Analysis of Routing Efficiency in the National Airspace System **4**

2.1 Introduction 4
2.2 Background 5
2.3 Data Description and Exploratory Analysis 6
2.4 Regression Analysis 8
2.4.1 Airport Level Comparison 8
2.4.2 Airport-Month Level Comparison 10
2.5 Conclusions 10

3 Causal Analysis of En Route Flight Inefficiency **17**

3.1 Introduction 17
3.2 Preliminaries 20
3.2.1 Data Sources and Summary Statistics 20
3.2.2 Motivations from Previous Work 25
3.3 Nominal Routes, Attribute Identifications and Matching Algorithms 27
3.3.1 Trajectory Clustering 27
3.3.2 Mapping Framework 33
3.3.3 Convective Weather Mapping Algorithm 35
3.3.4 Wind Mapping Algorithm 36
3.3.5 Traffic Management Initiatives and Special Activity Airspace Mapping Algorithm 38
3.4 Statistical Analysis 41
3.4.1 Modeling Philosophy 41

3.4.2 Model Specification 43
3.4.3 Estimation Results 45
3.4.4 Contributions of Causal Factors 47
3.5 Conclusions 52

4 Predicting Aircraft Trajectories: A Deep Generative Convolutional Recurrent Neural Networks Approach 56
4.1 Introduction 58
4.2 Related Work 59
4.3 Preliminaries 61
4.3.1 Definitions and Notations 61
4.3.2 Problem Formulation 61
4.3.3 Data Sources and Preprocessing 62
4.4 Feature Engineering 64
4.4.1 Feature Cube Referencing System 66
4.4.2 Feature Cube Matching 68
4.5 Module Design 69
4.5.1 Network Architecture 72
4.5.2 Inference Process 74
4.5.3 Implementation Details 76
4.6 Experimental Results 80
4.7 Conclusions 83

Bibliography 87

Appendix A List of US 34 Core Airports 93

Appendix B List of European 34 Core Airports 95

List of Figures

2.1 Summary of Horizontal En Route Inefficiency 7
2.2 Kernel Density Estimation of the Gap Between Actual and Flight Plan Distance $(A - A_p)$ 8
2.3 Horizontal Inefficiency Box Plots by OD pairs 12
2.4 Model I-I: Airport Fixed Effects Estimates 13
2.5 Model I-I: Distance and Monthly Fixed Effects Estimates 13
2.6 Model I-I: Departure Airport Fixed Effects Estimates 14
2.7 Model I-I: Arrival Airport Fixed Effects Estimates 14
2.8 Model I-II: Fixed Effects Estimates 15
2.9 Model I-III: Fixed Effects Estimates 15
2.10 ATL to EWR 16

3.1 Flight tracks from IAH to BOS in 2013 21
3.2 QCLCD snapshot data from 18:00 to 21:00 Zulu on 08/08/2013 22
3.3 Wind Speed Map (m/s) at 200 mbar Derived From the NCAR Datasets at 18:00 Zulu on 02/04/2013 23
3.4 An Actual Miles-In-Trail Implemented at ONL From the Denver ARTCC (ZDV) to Protect the Salt Lake City ARTCC (ZLC) 24
3.5 Actual AFP data 25
3.6 Spatial Distribution of Special Activity Airspace 26
3.7 Route Choice Mechanism 27
3.8 Reroute Mechanism 27
3.9 Trajectory Clustering for Flights from IAH to BOS 29
3.10 Trajectory Clustering for Flights from BOS to IAH 29
3.11 Trajectory Clustering for Flights from JFK to FLL 30
3.12 Trajectory Clustering for Flights from FLL to JFK 30
3.13 Trajectory Clustering for Flights from SFO to MCO 31
3.14 Trajectory Clustering for Flights from SFO to MSP 31
3.15 Trajectory Clustering for Flights from EWR to LAS 32
3.16 Trajectory Clustering for Flights from LAX to CLE 32
3.17 Illustration of the Mapping Framework 35
3.18 Pearson Correlation of Attributes Across Alternatives 53

4.1 Wind Speed Map at 200 mbar Derived From the Atmospheric Datasets 63
4.2 Comparison of NCWF datasets 64
4.3 Georeferencing System **G** 65
4.4 Actual Flight Trajectories vs. Their Flight Plan 66
4.5 Illustration of Feature Cube Creations 67
4.6 Feature Cube Grid Path Projected on the 2D Space 67
4.7 Training Framework 71
4.8 Inference Framework 78
4.9 Structure of the Convolutional Layers 79
4.10 Examples of our Predictions 81
4.11 Histogram of Prediction Errors 82
4.12 Error Propagation 82
4.13 CNN Feature Representation 85
4.14 Outputs from the Third Convolutional Layer 86

List of Tables

3.1 Summary statistics for representative city pair inefficiency analysis 21
3.2 Summary statistics for MIT restrictions 23
3.3 Average Inefficiency within Clusters 34
3.4 Variance Explained by Clusters 34
3.5 Description of Variables 45
3.6 Nesting Structures 46
3.7 Estimation Results for the Logit Models 48
3.8 Estimation Results for the Inefficiency Model 49
3.9 Estimation Results for the Inefficiency Model (Continued) 50
3.10 Contribution of Causal Factors 52
3.11 Percentage Change In HIE by Zeroing Out Factor(s) of One Alternative 55

4.1 Average of the Four Absolute Errors 83

Acknowledgments

I would like to thank my thesis advisor, Dr. Mark Hansen, for his valuable insight and guidance through years of my PhD career. I would also like to thank Dr. Michael O. Ball and Dr. David J. Lovell from the University of Maryland for their advice and support of many contents in this thesis. Thank you also to the rest of my committee members, Dr. Michael J. Cassidy, Dr. Peng Ding, and Dr. Joan Walker, for their encouragement and constructive comments on my work. I'd also like to thank our sponsors John Gulding and Joseph Post from the Federal Aviation Administration, and John Schade from the ATAC, without whom this research would not be possible.

Many thanks to my colleagues and collaborators: Dr. Alexander S. Estes, Dr. Lei Kang, and Dr. Mogeng Yin, who have shared their research experiences and insights to help me develop interest and confidence in my early academic career. Thank you also to my other colleagues: Dr. Bo Zou from the University of Chicago, Dr. Yu Zhang from the University of South Florida, Dr. Yanjun Wang from the Nanjing University of Aeronautics and Astronautics, Cara Chuang from the University of Maryland, and Dr. Yi Liu, Dr. Lu Hao, Lu Dai, Ivan Tereshchenko, Ang Li from our research group.

I would also like to thank my friends in the ITS Berkeley community: Dr. Chao Mao, Dr. Yanqiao Wang and Dr. Sheng Liu for all the fun we had together. I will miss many of our ping-pong nights and tennis afternoons. Special thanks to my college friend Xi Lin from the Tsinghua University for his consistent encouragement regardless of the 15-hour time difference. Thank you also to Mengqiao Yu, Ke Xu, Meng Qi, Dr. Ying Cao, Bowen Yang, Dr. Wei Ni, Dr. Renyuan Xu, Dr. Zhe Ji, Dr. Xu Wu for all their support.

Lastly, I would like to thank my parents Yang and Hui, not only for their unconditional love and support over the years, but also for their selfless dedications to fighting against the COVID-19 on the front line. Thank you also to my fiancée, Shu Huang, for her love, care and support.

Chapter 1

Introduction

Recent years have witnessed growing interest in data analytic techniques to assess the overall performance in the national airspace system (NAS). In the past decades, air navigation service providers (ANSPs), including the Federal Aviation Administration (FAA) and EUROCONTROL, have spent tremendous effort on system level projects that seek to identify key performance indicators (KPIs), define metrics, assess trends, discover patterns, and compare performance. Among these assessments, the en route performance of the airspace, which is closely related to flight trajectory analysis, is one of the most important KPIs due to its significant impact on aviation economics and the environment. Therefore, ANSPs have made many attempts to improve it. For example, since 2008, EUROCONTROL has begun implementing "free route airspace (FRA)" in the European airspace such that flights may freely plan a route, if available, without reference to the air traffic service (ATS) route network ([55], [20]). They report that this initiative has benefited the flight operators by reducing average routing inefficiency from 3.17% in 2012 to 2.77% in 2017. In the US, much of the attentions has been focused on balancing air traffic demand and capacity instead of improving the en route performance. These traffic flow management initiatives, including miles-in-trail (MIT) restrictions and airspace flow programs (AFPs), however, might be harmful to the routing efficiency. Therefore, investigating their effects on the en route performance is one of the major aims in this work.

On the other hand, the concept of trajectory-based operations (TBO) has received more attention in the development of the next generation of the national airspace system (NextGen). In 2016, the FAA announced the "TBO Vision 2025" program that aimed to achieve greater predictability, higher levels of automation, more efficient strategic planning, and fewer tactical interventions ([26]). As a fundamental requirement, TBO seeks the development of tools to both strategically plan and tactically predict accurate 4D aircraft trajectories (latitude, longitude, altitude and time), such that air traffic control can manage traffic in a timely manner. Recent years have seen many developments on systems that focus on optimizing and controlling aircraft 4D trajectories ([5], [59], [60]), yet, there are far fewer studies on resolving the fundamental task of the TBO – developing tools to predict 4D flight trajectories in real time.

These challenging tasks demand individual flight trajectory modeling techniques, which can be addressed through machine learning, data mining and deep learning methods thanks to recent enhancements in computing power and data storage. In particular, the FAA established a System Wide Information Management (SWIM) system ([52], [68], [67]) that archives massive real-time information that covers flight track surveillance data, weather activities, traffic management initiatives and airport operational status. This system, therefore, offers us the opportunity to develop data-driven methodological frameworks that scientifically understand the mechanisms and causes behind degraded NAS performance, and further, to build predictive models that improve decision making.

In this thesis, we present three papers around the topic. In Chapter 2, we explore patterns of flight en route inefficiency for domestic flights into and out of the US 34 core airports using a dataset of approximately six million flights from the years 2013 and 2014. Our inefficiency metric, which is defined by the FAA and EUROCONTROL, compares the distance flown between airport terminal exit and entry points with a benchmark distance, and further isolates the effects of pre-specified, and often not ideal, entry and exit points and excess distance flown between these points. Macroscopic fixed effect models are proposed to estimate the variations of routing inefficiency across different departure/arrival airports, seasons, and flight length.

Chapter 3 continues the discussion of uncovering the mechanisms behind the flight en route inefficiency. In this chapter, we first identify the possible factors that cause the inefficiency through comprehensive literature review and data mining techniques. These factors include convective weather, wind, miles-in-trail (MIT) restrictions, airspace flow programs (AFP) and special activity airspace (SAA). Second, we propose two mechanisms that directly link en route inefficiency with the routing decision-making process, in which flight operators make strategic routing decisions prior to departure and perform tactical rerouting airborne. Both phases of decision making can be ascribed to the list of identified causal factors. Third, we have developed a framework, which includes trajectory clustering and attribute matching, to reveal nominal traffic flows and their route-specific attributes within a specified airport pair. Based on this discovery, we are able to construct a scenario whereby each flight needs to choose a route from the nominal flows, which mimics the strategic flight planning process. Fourth, we establish logit models, including multinominal logit and nested logit models, to quantitatively understand, for each airport pair, how different causal factors affect the choice of a nominal route. A similar strategy is applied to model the tactical rerouting through a set of inefficiency (linear) models. Lastly, we have conducted counterfactual analysis to estimate the contributions of each of these causal factors to en route inefficiency.

In Chapter 4, while still focusing on the en route airspace, we shift our focus to modeling individual flight trajectories. As one of the most direct solutions to the fundamental task of TBO, we have developed a framework to predict real-time 4D aircraft trajectories. To this end, we first propose a highly generalizable efficient tree-based matching algorithm to construct image-like feature maps from high-fidelity meteorological datasets — wind, temperature and convective weather. We then model the track points on trajectories as conditional Gaussian mixtures with parameters to be learned from our proposed deep generative model, which is an end-to-end convolutional recurrent neural network that consists of a long short-term memory (LSTM) encoder network and a mixture density LSTM decoder network. The encoder network embeds last-filed flight plan information into fixed-size hidden state variables and feeds the decoder network. The decoder network further learns the spatiotemporal correlations from the historical flight tracks and outputs the parameters of Gaussian mixtures. Convolutional layers are integrated into the pipeline to learn representations from the high-dimension weather features. During the real-time inference process, beam search, adaptive Kalman filter, and Rauch-Tung-Striebel smoother algorithms are used to prune the variance of generated trajectories.

Chapter 2

Performance Analysis of Routing Efficiency in the National Airspace System

2.1 Introduction

In recent years, the aviation research community has witnessed growing interest in comparative performance assessment of air navigation service providers (ANSPs). Such efforts, however, face difficult challenges, in particular the need to identify key performance indicators (KPIs) that are precisely defined and can capture the major dimensions of aviation system performance. The potential payoffs from comparative performance assessment more than justify such work, however. These include identifying opportunities for performance improvement, determining the benefits from modernization, and more fundamentally understanding the linkages between structure and performance in the air navigation service domain.

Comparative assessments typically emphasize macroscopic comparisons. For example, the most recent comparison of US and European air navigation system performance [29] offers conclusions such as "Europe continues to demonstrate less additional time in the taxi-out phase than in the US" and "the US continues to show a lower level of inefficiency in the airborne phase of flight". From the standpoint of decision makers, such high-level conclusions represent the ultimate payoff from vast amounts of data collection and analysis. To researchers, however, they beg more detailed questions. For example, are taxi-out times in the US high everywhere or are the results skewed by a few highly congested airports? Similarly, is airborne inefficiency fairly constant across space and time, or are there pronounced patterns of variation and, if so, what are they? Answers to these questions, in addition to satisfying basic curiosity, may be more relevant to improving system performance than macro comparisons. For example, it may be far easier to import best practices from one region of a single air navigation system to another than to do so across systems.

The reminder of this chapter is organized as follows. Section 2.2 introduces the background and definition of our inefficiency metrics. Section 2.3 presents data sources and exploratory data analysis. In Section 2.4, we use regression techniques to estimate fixed effects of airports, seasons, and flight lengths. Section 2.5 offers conclusions.

2.2 Background

The en route phase of a flight is defined as the portion between a 40 nautical-mile circular boundary around the departure airport (D40) and a 100 nautical-mile circular boundary around the arrival airport (A100). This definition is intended to exclude the portions of the flight path that are strongly influenced by terminal operations. Horizontal en route inefficiency, which evaluates actual flight trajectories against a benchmark trajectory, has received considerable attention in the literature. Ref. [29] calculates the horizontal inefficiency based on the extra distance flown in the en route phase with respect to an ideal distance known as "achieved distance". The latter represents the average of how much further the flight has gotten from its origin and how much closer it has gotten to its destination over the en route portion of the flight ([21], [31]). This method, instead of choosing great circle distance[1] between OD airports as the benchmark, excludes the effect of terminal inefficiency. Equation 2.1 explicitly expresses the definition, where HIE is the horizontal en route inefficiency of a flight, A is the actual flown distance, and H is the benchmark distance, or "achieved distance".

$$HIE = \frac{A-H}{H} \tag{2.1}$$

The en route inefficiency can be further decomposed into two parts: direct route (DIR) extension inefficiency; and terminal (TMA) extension inefficiency. While the first component is primarily driven by the efficiency of the path between the actual terminal area entry and exit points, the TMA extension inefficiency reflects the inefficiency that derives from locations of these points, which are usually not on the great circle route between the origin and destination. The decomposition is illustrated by Equation 2.2, where D is the great circle distance from the exit point (of the origin airport) to the entry point (of the destination airport).

$$HIE = DIR + TMA = \frac{A-D}{H} + \frac{D-H}{H} \tag{2.2}$$

Based on these metrics, the US-Europe Performance Report [29] compares en route inefficiency in the US and Europe. Both markets have been on a downward trend for en route inefficiency, but the US in general is more efficient than Europe. In 2013, the flight inefficiency calculated from equation 2.1 is 2.7% for the US and 2.9% for Europe. The report documents certain patterns for the US and Europe. It suggests that flights to New York and

[1] The great circle distance is the shortest distance between two points on a spherical surface

the Florida area are systematically more inefficient, mostly due to the avoidance of special activity airspace (SAA) and long transcontinental operations. In Europe, the implementation of free route airspace (FRA) improves the en route efficiency significantly, especially for flights through those FRA areas.

However, one major criticism of the en route inefficiency metric is the selection of the "achieved distance" as the benchmark distance. Although this benchmark takes into account the deviation between exit/entry points with a direct route from origin to destination, it is not an "optimum" trajectory distance for most flight operations, when considering meteorological conditions. For instance, Calvo et al. [7] proposed using the fuel efficiency as the metric to evaluate flight efficiency. Instead of using absolute distance, [7] calculated the inefficiency based on the additional fuel burn of the actual trajectory over the great circle trajectory between the exit and entry points. Although this metric performs quite differently from the route extension metric for some flights, the two are highly correlated most of the time. Therefore, route extension metric, while not perfect, has the virtue of simplicity and appears to correlate well with more refined metrics.

2.3 Data Description and Exploratory Analysis

We obtained flight level performance data from the Federal Aviation Administration (FAA) enhanced traffic management system (ETMS). The data cover around twelve million flights arriving at 34 US core airports (see Appendix A) from January 1st, 2013 to December 31st, 2014. Of these, 87% are domestic flights and less 1% are diverted flights[2] or missing records. Each record includes flight level information containing the departure and arrival times, origin and destination airports, and a list of distance metrics derived from the actual radar tracks, such as actual flown distance (A), flight plan distance (A_p), achieved distance (H), and direct route distance fro exit to entry points (D).

In this study, we limit our scope to flights that only flew into and from the 34 US core airports, which represents most of US instrumental flight rule (IFR) flights[3] ([29]). After removing all international flights and diverted flights, we obtained a dataset with six million flight records in 2013 and 2014. This encompasses about 50% of the total traffic in the ETMS database. Based on the data, we first used Equations 2.1 and 2.2 to calculate the en route inefficiency and its decomposition for each flight. We then compared inefficiencies from the perspective of flight length, airport pair and season.

Figure 2.1 shows the average horizontal en route inefficiency based on a flight's great circle distance category and year. We first observe that there is no substantial difference in inefficiency across all flight length groups between years 2013 and 2014. Overall, the average inefficiency across all flights is 3.413% for 2013, which is only 0.006% higher than 2014.

[2] A diverted flight is one that does not arrive at its original destination, but a new, usually temporary, airport

[3] An IFR flight "depends upon flying by reference to instruments in the flight deck, and navigation is accomplished by reference to electronic signals" ([24])

We also observe that long-haul flights (e.g., > 1000 nautical-mile flights) tend to be more efficient than short-haul flights (e.g., 0 – 200 nautical-mile flights). This is primarily the result of decreasing TMA extension inefficiency with the flight distance; i.e., excess distance from inefficient placement of entry and exit points is independent of great circle distance, while the DIR inefficiency is roughly independent of great circle distance. The latter also implies that the excess distance between entry and exit points is roughly proportional to the great circle distance. On the other hand, long-haul flights are more likely to adjust their routes – typically take short cuts – from their original flight plans. To support this claim, we have shown in Figure 2.2 a kernel density estimation (KDE, [58]) of the gap between the actual flown and flight plan distances ($A - A_p$) in 2013. Notice that the bandwidth parameter of the KDE has been selected through cross validation. Each graph of Figure 2.2 illustrates the KDE for one distance category shown on the upper left corner. While all six estimated distributions are negatively skewed with modes smaller than 0, long-haul flight groups (e.g., > 1000 NM) have longer left tails and heavier weights toward the negative side. This suggests that the actual flown distance is generally shorter than the flight plan distance, and that this difference tends to be greater for longer flights.

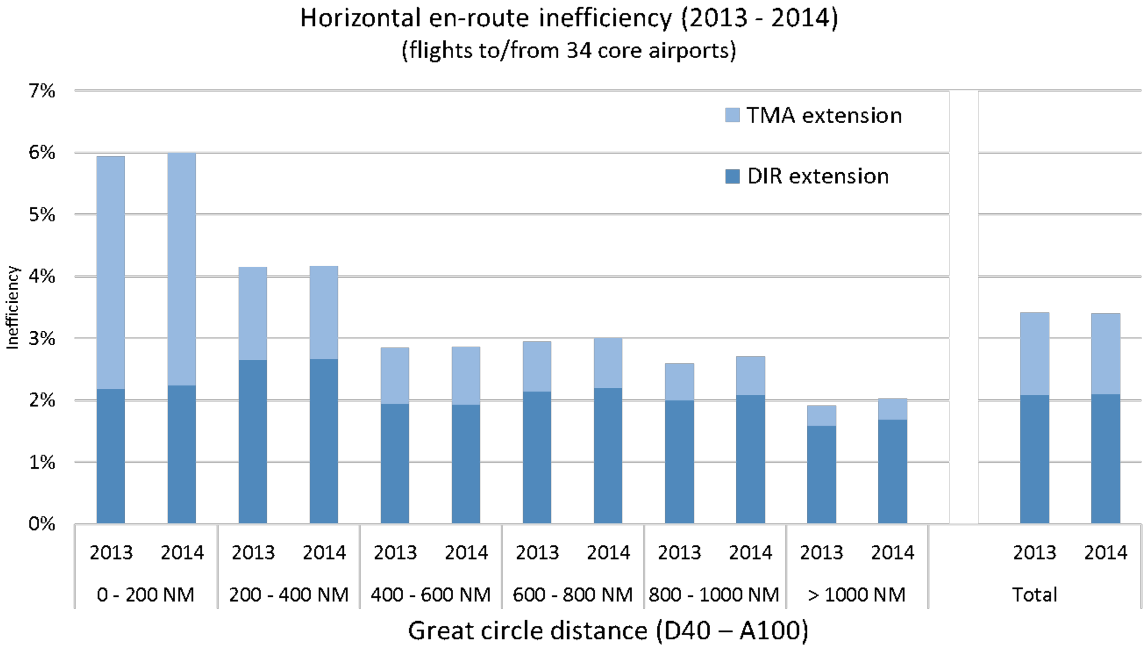


Figure 2.1: Summary of Horizontal En Route Inefficiency

We further summarize the monthly en route inefficiency performance for 1076 airport pairs in our dataset. To save space, we only present the results for six pairs. These have similar route distances, but significantly different inefficiency distributions. Figure 2.3 shows a set of box plots of HIE across months and airport pairs. All of the graphs show that

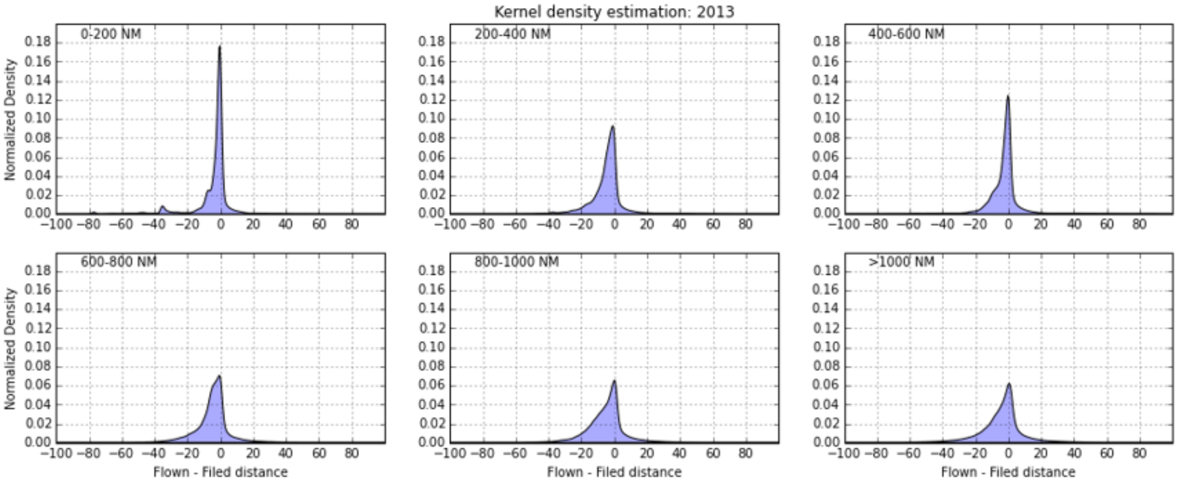


Figure 2.2: Kernel Density Estimation of the Gap Between Actual and Flight Plan Distance $(A - A_p)$

inefficiency is skewed to the top, and that the skewness is most pronounced for city pairs that are relatively efficient. However, the detailed patterns are drastically different across pairs. For example, in the first two subplots, 2.3a and 2.3b, EWR and JFK airports are both in the New York metropolitan area and share a great portion of airspace. However, flights from ATL airport to these two airports demonstrate clear differences in that inbound traffic to JFK has much higher inefficiency. In the next few sections, we will use basic fixed effect regression models to further conduct these types of macro comparisons among departure/arrival airports, and across different months.

2.4 Regression Analysis

The primary goal of this work is to investigate variation in the en route inefficiency metrics for domestic US flights among the core 34 airports. We focus on factors related to origin and destination airports, season, and flight distance. We apply linear regression techniques to explore and quantify factors that potentially impact flight horizontal en-route inefficiency. We are more interested in macro patterns than the specific circumstances that affect the inefficiency of individual flights.

2.4.1 Airport Level Comparison

We have estimated three fixed effects models to microscopically understand the variations in flight en route inefficiency with respect to airports and seasons. All three models have the same explanatory variables but different dependent variables — en route inefficiency (HIE), terminal extension (TMA) inefficiency and direct route (DIR) extension inefficiency.

In the models, we include four types of fixed effects. The first two types are the departure and arrival airports. We expect that flights to or from airports with higher traffic demand will be more inefficient than the others. The third type includes the monthly fixed effects, which are treated as proxies for convective weather. We expect estimates for summer months (June – August) to be systematically larger than the others. The last type considers the impacts of flight length. We separate flights into six categories based on their D40A100 great circle distance: a) 0 – 200 nm, b) 200 – 400 nm, c) 400 – 600 nm, d) 600 – 800 nm, e) 800 – 1000 nm and f) larger than 1000 nm. As stated in the previous section, short-haul flights are prone to more inefficient en route operations. Thus we expect a shorter distance category to have a larger fixed effect. Based on the specification, we can formulate the regression model as Equation 2.3, where $\mathbf{Y}_{HIE}, \mathbf{Y}_{TMA}, \mathbf{Y}_{DIR}$ (in percentage) are respectively the HIE, TMA, and DIR flight inefficiency, and $\mathbf{X}$'s are fixed effect variables.

$$\begin{aligned} \text{Model I-I:} \quad & \mathbf{Y}_{HIE} = \boldsymbol{\alpha}' \cdot \mathbf{X}_{dep} + \boldsymbol{\beta}' \cdot \mathbf{X}_{arr} + \boldsymbol{\gamma}' \cdot \mathbf{X}_{month} + \boldsymbol{\theta}' \cdot \mathbf{X}_{dist} \\ \text{Model I-II:} \quad & \mathbf{Y}_{DIR} = \boldsymbol{\alpha}' \cdot \mathbf{X}_{dep} + \boldsymbol{\beta}' \cdot \mathbf{X}_{arr} + \boldsymbol{\gamma}' \cdot \mathbf{X}_{month} + \boldsymbol{\theta}' \cdot \mathbf{X}_{dist} \\ \text{Model I-III:} \quad & \mathbf{Y}_{TMA} = \boldsymbol{\alpha}' \cdot \mathbf{X}_{dep} + \boldsymbol{\beta}' \cdot \mathbf{X}_{arr} + \boldsymbol{\gamma}' \cdot \mathbf{X}_{month} + \boldsymbol{\theta}' \cdot \mathbf{X}_{dist} \end{aligned} \tag{2.3}$$

We use the dataset obtained from Section 2.3, which contains about six millions flight records from 2013 to 2014, to estimate the above fixed effects models. Estimation results for the model of horizontal en route inefficiency, HIE (Model I-I), are shown in Figures 2.4 and 2.5. The baseline for airport fixed effects is the Tampa International Airport (TPA), and the baselines for month and flight length are respectively January and the $>$ 1000 NM. The error bars in both figures show the 95% confidence intervals. Estimation results for the majority of variables are significant. Figure 2.4 suggests that the fixed effects for arrival airports, which range from -1.7% to 3.4%, vary more than those for departure airports, which range from -2.8% to 0.2%. Meanwhile, airports such as LGA (LaGuardia airport) and FLL (Fort Lauderdale Airport) have negatively correlated estimates for departure and arrival fixed effects, indicating that some airports may have relatively efficient departure procedure but inefficient arrival procedure, or vice versa. To further demonstrate the geographic pattern of airport fixed effects, we present two 3D bar plots in Figures 2.6 and 2.7. In both figures, each bar shows the comparative magnitude of airport fixed effect at each airport, and the height is proportional to the magnitude of the airport fixed effects. We first observe that flights into or from airports in the east coast corridor and south coast are in general more inefficient than those in the west coast. This effect is most pronounced for JFK airport, as flights to JFK almost double their en route inefficiencies compared to those to FLL, which has the second highest arrival fixed effect. We also notice from Figure 2.5 that the monthly fixed effects have a concave shape, where June, July and August are systematically higher than the other months. This suggests less efficient en route performance for flights in the summer season. Lastly, the estimates for the flight length confirm that flight en route inefficiency decreases with the great circle distance between the airport pairs. For example, flights within

200 nautical miles are on average 4.26% less efficient than flights longer than 1000 nautical miles.

Figures 2.8 and 2.9 show the estimation results for models with direct route extension (DIR) and terminal extension (TMA) inefficiencies as dependent variables, respectively. While those airport-level fixed effects have similar patterns as in Model I-I, flight length has a much stronger impact on TMA inefficiency than the DIR. This also matches our expectation. In the meantime, the monthly fixed effects show more importance with larger magnitudes for DIR than for TMA. Lastly, by comparing Figures 2.8 and 2.9, we also find that there is only modest correlation between the DIR and TMA airport fixed effects. This implies that the same airport may feature relatively efficient terminal procedures and relatively inefficient direct route procedures, or vice versa.

2.4.2 Airport-Month Level Comparison

In previous models, airport and monthly fixed effects are estimated separately. However, we also want to compare the monthly variations of flight inefficiency within each airport. Therefore, in this section, we estimate a model that includes all interaction terms between airports and months. The new model, denoted as Model II, can be written as in Equation 2.4, where $\mathbf{Y}_{HIE}$ is the horizontal en route inefficiency, $\mathbf{X}_{dep-month}$ and $\mathbf{X}_{arr-month}$ are vectors that contain the tuple of departure/arrival airports and months, and $\mathbf{X}_{dist}$ is the flight length fixed effect variables, as before.

$$\text{Model II:} \quad \mathbf{Y}_{HIE} = \boldsymbol{\alpha}' \cdot \mathbf{X}_{dep-month} + \boldsymbol{\beta}' \cdot \mathbf{X}_{arr-month} + \boldsymbol{\gamma}' \cdot \mathbf{X}_{dist} \tag{2.4}$$

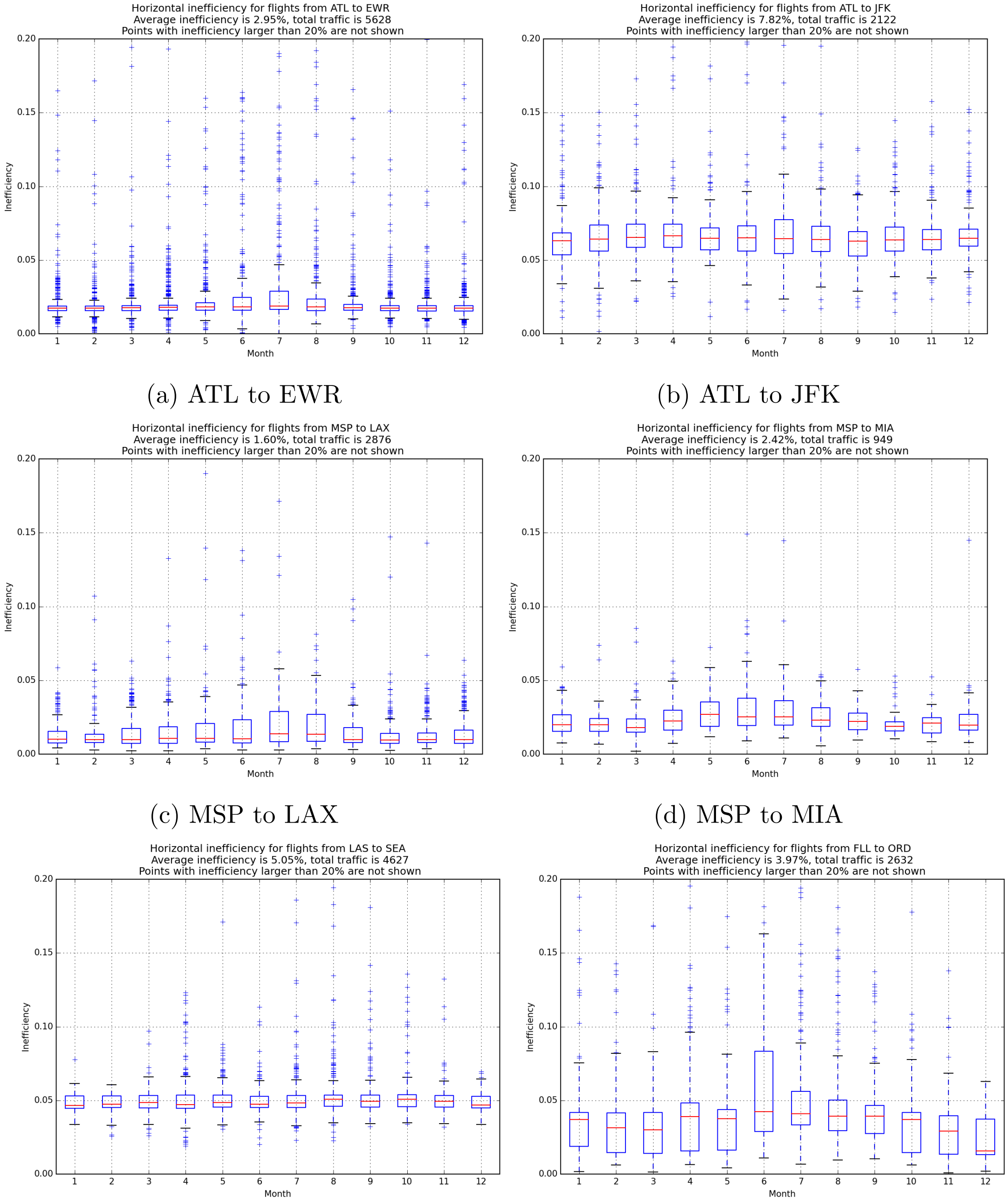
We use the dataset in Section 2.4.1 to estimate Model II, and the estimation results are presented in Figure 2.10. Each subplot shows the monthly fixed effects for a specific airport depicted in the title. The blue bars are the departure fixed effects, and the orange bars are the arrival fixed effects. We observe that the fixed effects for departure airports have a concave shape as regards months. However, such phenomenon is less obvious for arrival airports. In fact, arrival airports such as BOS, SFO and LAX have a convex shape with respect to months, indicating more efficient en route operations for summer-time flights into those airports.

2.5 Conclusions

In this research, we have conducted macroscopic studies to explore sources of variations with respect to flight horizontal en route inefficiency qualitatively and quantitatively. We apply fixed effects regression techniques to a two-year flight-level performance dataset, which includes about six million flights from and into the US core 34 airports. We estimate the variance of inefficiency across different departure/arrival airports, seasons, and flight lengths.

Our estimation results suggest that flights from or to airports in the east coast corridor and south coast, especially in the New York metropolitan and Florida area, are generally more inefficient than the others. This is consistent with our expectation and the literature. Long-haul flights are more efficient, mainly because one component of excess distance – which results from flights using fixed entry and exit points that are not on the great circle route – is roughly independent of distance. Specifically, flights that are less than 200 nautical miles are 4% less efficient than flights longer than 1000 nautical miles. In addition, flights operated during the summer, when convective weather is more frequent, are less efficient than those in other seasons. The extended model, with emphasis on monthly variations within each airport, reveals that the departure operations are strongly impacted during summer, while the phenomenon is less obvious for arrivals. Moreover, the seasonal effects, though they differ in absolute magnitude, have similar patterns for geographically adjacent airports.

Finally, we point out several limitations and future work of this study. First of all, in all of our fixed effect models, the analysis of variance reports a low percentage of variance explained by those identified fixed variables. Although the estimates of included explanatory variables are mostly significant with reasonable signs, there are still a number of unobserved factors not captured by our models. In the literature, there are extensive studies on flight trajectory operations suggesting that wind and convection have strong impacts on strategic routing ([66], [57], [50], [49]), while en route traffic management initiatives such as miles-in-trail (MIT) affect tactical rerouting ([53], [56], [64]). These factors are therefore likely to lead to additional flight inefficiency. Secondly, all of our models focus on the macroscopic level of performance. However, we observe from Figure 2.3 that the variations for a specific airport pair within one month are quite pronounced. This suggests the need for a model that considers circumstances specific to each flight to explain the reason. One very important factor, as we have observed from individual flight track data, is that flight trajectories between a given airport pair typically follow one of a limited number of pre-specified routes, which have varying levels of inefficiency. Modeling the route to which a flight is assigned can therefore be an important step in the analysis of flight inefficiency. We will discuss this topic in the next chapter.



(a) ATL to EWR

(b) ATL to JFK

(c) MSP to LAX

(d) MSP to MIA

(e) LAS to SEA

(f) FLL to ORD

Figure 2.3: Horizontal Inefficiency Box Plots by OD pairs

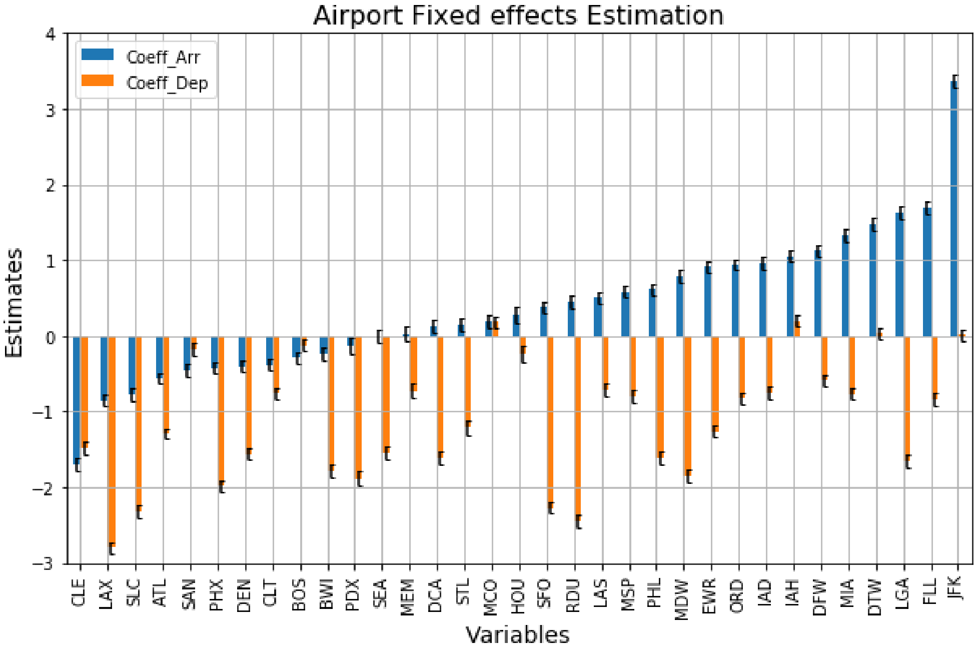


Figure 2.4: Model I-I: Airport Fixed Effects Estimates

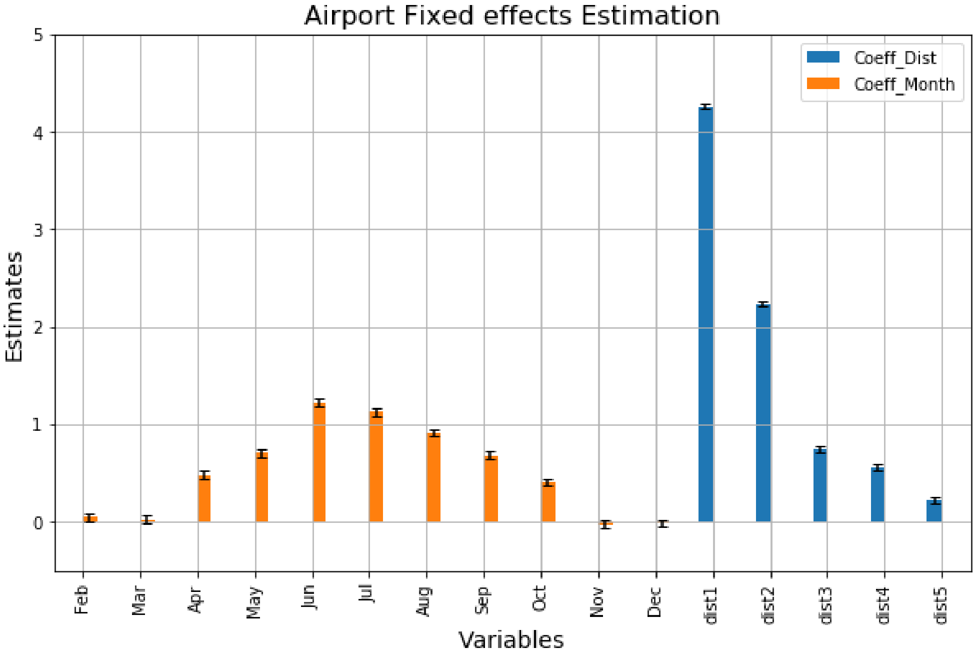


Figure 2.5: Model I-I: Distance and Monthly Fixed Effects Estimates

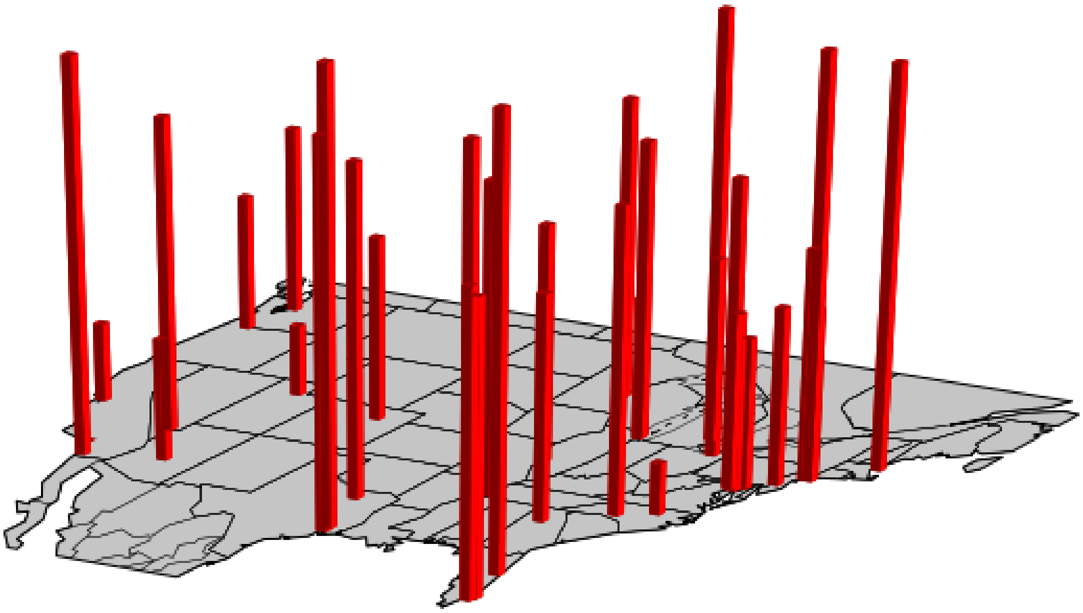


Figure 2.6: Model I-I: Departure Airport Fixed Effects Estimates

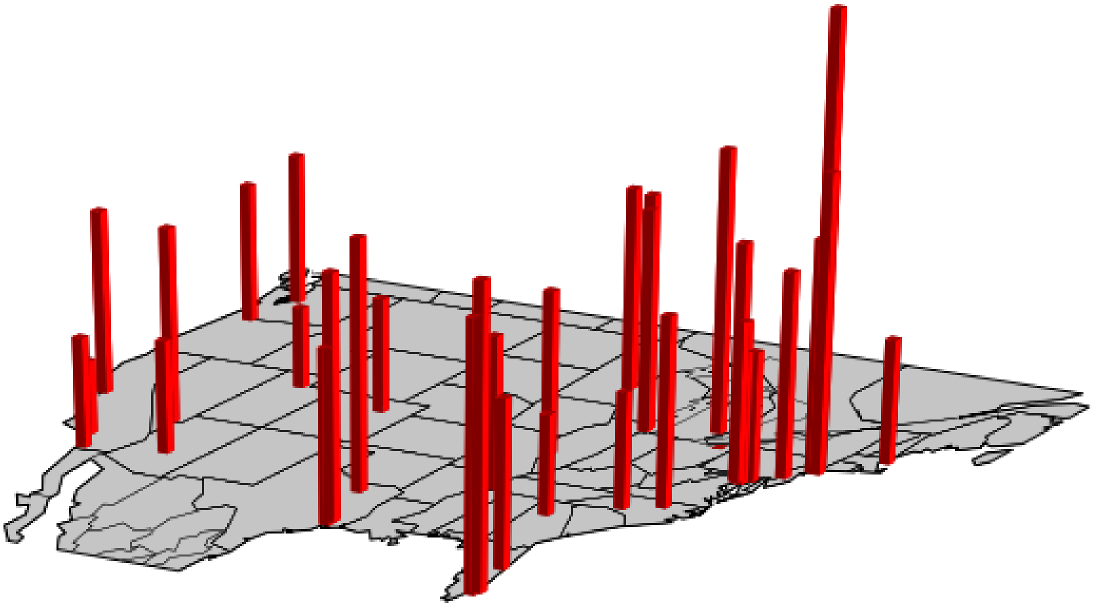


Figure 2.7: Model I-I: Arrival Airport Fixed Effects Estimates

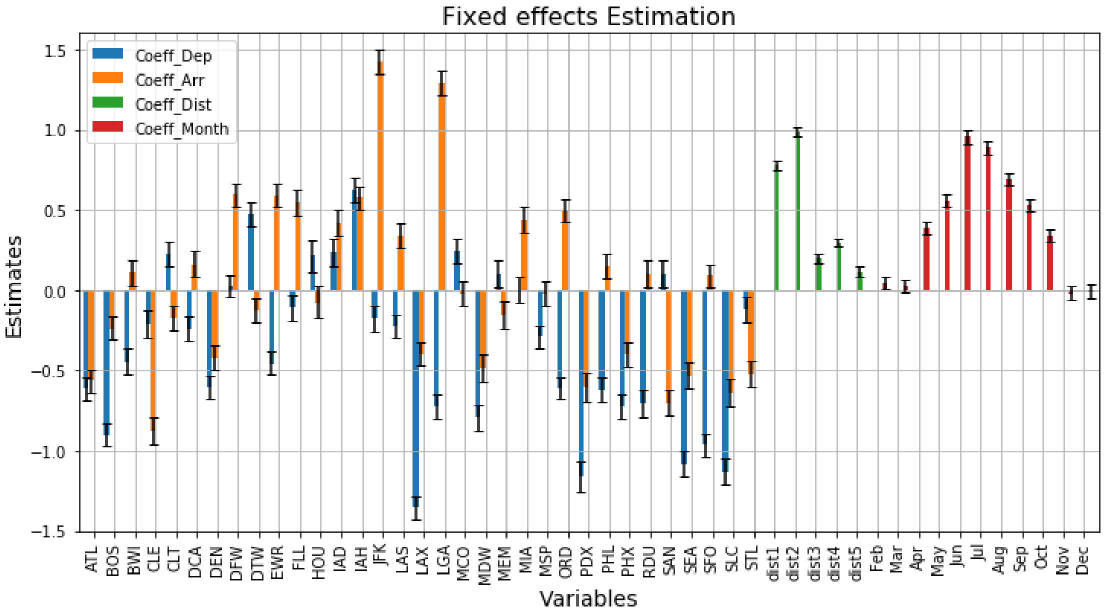


Figure 2.8: Model I-II: Fixed Effects Estimates

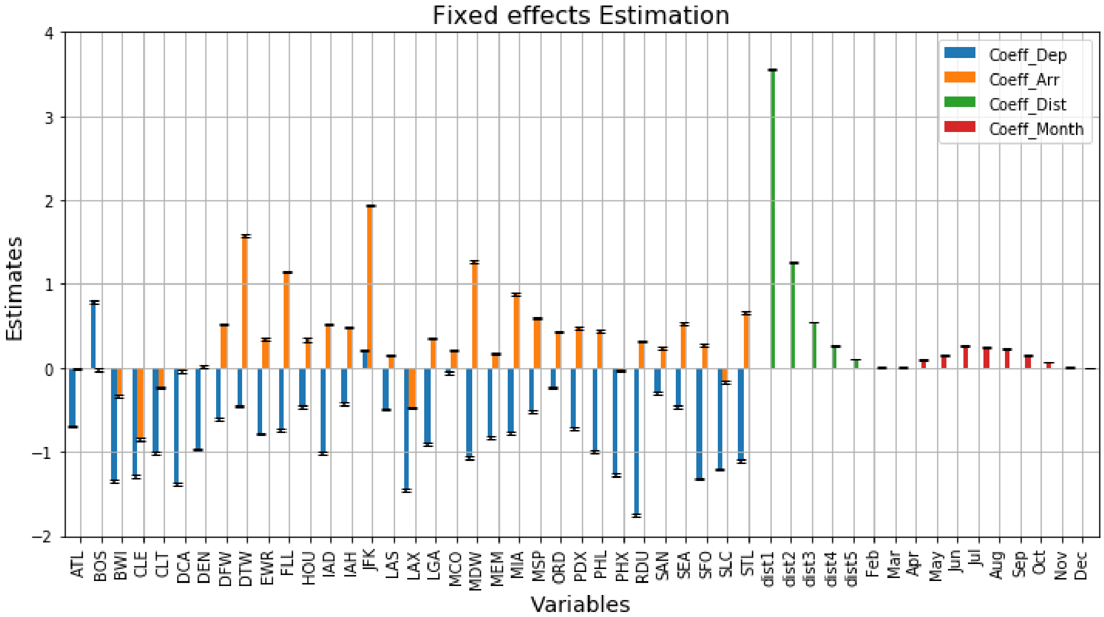


Figure 2.9: Model I-III: Fixed Effects Estimates

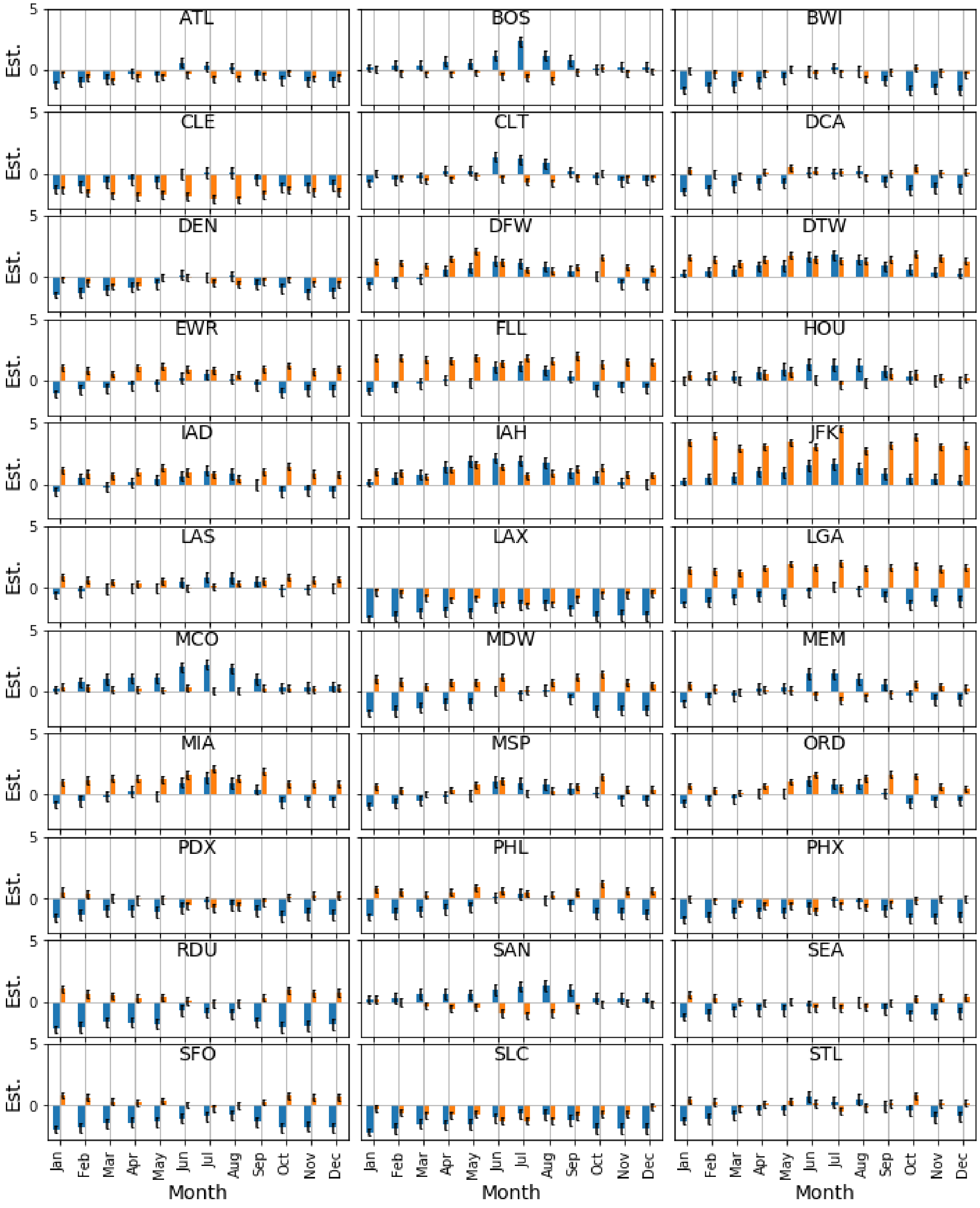


Figure 2.10: ATL to EWR

Chapter 3

Causal Analysis of En Route Flight Inefficiency

3.1 Introduction

Horizontal en route inefficiency, which evaluates actual flight trajectories against a benchmark trajectory, has received considerable attention in the literature. It is a fundamental measure in the European performance scheme for air navigation services [13]. In 2016, the most recent edition of the ICAO global air navigation plan (GANP) listed an en route trajectory metric as a recommended indicator for assessing aviation system block upgrades (ASBUs) [40]. Although the US does not employ targets for trajectory measures, they are still used in many applications. They are core to many fuel efficiency benefit assessments and the air traffic organization (ATO) records airborne holding as part of its reportable delay in OPSNET.

In this chapter, we use the same methodology and Equation 2.1 as described in Chapter 2 to calculate horizontal en route inefficiency (HIE), which is based on ground track distance in excess of the benchmark distance, as in the US-Europe Performance report [30]. Admittedly, there are many criticisms of using “achieved distance” as the benchmark distance for calculating en route inefficiency simply because the great circle distance (GCD) is not an “optimum” trajectory for most flight operations when considering meteorological conditions. For example, [66] examines the benefit of wind-optimal trajectories in transatlantic routes as measured by fuel efficiency. In many cases, a flight taking the shortest ground path would actually burn more fuel and sometimes take longer airborne time, thus-under performing in those standards of efficiency. However, given its simplicity and popularity among many performance assessment studies, we stick with Equation 2.1 as the measure of routing efficiency in this research.

In 2015, the average horizontal en route inefficiency for flights to and from the main 34 airports was 2.92% in Europe (See Appendix B for the 34 core EU airports) compared to 2.83% in the US (See Appendix A for the 34 core US airports). Europe experienced continuous improvement of en route efficiency between 2011 and 2014. However, this trend reversed in 2015. The US had an overall increase in flight inefficiency from 2013 to 2015, which can be linked to airports (LAX, SEA, DAL) with drastically increased air traffic. We have also observed a peak in inefficiency in 2014, which is caused by a fire at the Chicago air route traffic control center (ARTCC) in the fall of 2014.

Despite of great efforts spent on system level projects to define metrics, assess trends, and compare performance, work on linking routing inefficiency to causal reasons and determining if they can be mitigated through investment in technology, staffing or training has been limited. As a first attempt to analyze different sources of flight routing inefficiency, [63] used the discrepancy between the theoretically shortest lateral trajectory (great circle) and the actual flight trajectory as an inefficiency metric. This included departure inefficiency, en route inefficiency, and arrival inefficiency. In the study, the author identified 6 sources of inefficiency: a) departure procedures, b) restricted airspace, c) congestion, d) adverse weather, e) holding and vectoring in the terminal airspace, and f) arrival procedures. The author then used trajectory data from the US, Europe, and Africa to calculate the routing inefficiency on an "average day with no major weather". The work concluded that the standard routes and restricted airspace were the biggest contributors to the routing efficiency (27%). Using a similar strategy, the author estimated that the contribution of congestion to routing inefficiency in the US was approximately 13%; and the contribution of adverse weather was also about 13%. While this study has largely summarized key factors that might impact the flight routing efficiency, there are several limitations.

First, due to increasing air traffic demand, a growing number of traffic management initiatives (TMIs) have been deployed in recent years. Among various TMIs, airspace flow programs (AFP) and miles-in-trail (MIT) restrictions have received more attention in research due to their significant impacts on both strategic and tactical flight route planning. To be more specific, MIT ([27], [56]) is an iniative to manage air traffic into a manageable flow by enforcing a minimal spacing (number of miles) between aircraft over a fix, at an altitude, through a sector, or on a jet route. MITs are classified into three categories: departure MIT, en route MIT, and arrival MIT based on the locations where the MITs are actually implemented. Flights under MIT restriction(s) are therefore either slowing down or taking maneuvers to satisfy those spacing requirements. [53] conducted studies to match individual flight trajectories to historical MIT restrictions and investigated how those MITs affect flight performance. The authors found that 32% of MIT restrictions involved fewer than 5 flights. Even with those matched records, MITs did not appear to have a significant impact on airborne delay or en route spacing during the period of analysis (May 1 – 14, 2004). However, due to the age of the report and the increasing number of MITs implemented in more recent years, it is fair to say that very little is known about whether or to what degree MIT restrictions contribute to en route inefficiency in the present system. AFP ([27], [72]), on the other hand, identify flow constrained areas (FCAs) in the en route system and assign specific delays to flights that are filed into those areas. As a result, AFP mostly affect the flight planning process because flights are more likely to file a route that goes around those FCAs. [71] studied the airline decision-making process under the AFP using logit models on historical flight plan and AFP data. By modeling flight routing behavior in AFP – either route out of the AFP or stay in the AFP – the authors concluded that flights in AFP of higher duration are more likely to route out. This also implies that under AFP, flights tend to have higher en route inefficiency. Interested readers may also refer to [43] for more details on how AFP affect the flight rerouting process.

Second, while the authors emphasized the importance of convective weather in [63], they have somewhat ignored the effects of wind. As we have mentioned earlier, our inefficiency metric is purely a measure for ground track distance in excess of some theoretical shortest path. However, there is extensive work ([57], [33], [66]) suggesting that a flight almost always takes advantage of the wind, such that its route is "more wind-optimal", which is typically much longer than the great circle route. This effect is also most pronounced for transcontinental flights due to the jet streams.

Third, in [63], analysis was mostly based on macroscopic comparisons in different locations and times. However, in this study, we address more detailed questions such as "What is the mechanism of the en route inefficiency?", "Why would flights in similar conditions end up with drastically different inefficiencies?", and "If a factor affects the routing efficiency, then how and how much?". Answers to these questions require a more complex statistical modeling process. Specifically, the goals of this analysis are threefold.

1. Identify factors that affect route inefficiency based on existing literature and perceptions.

2. Quantify effects/contributions of those identified factors, which include convective weather, wind, SAA and traffic management initiatives, on en route efficiency.

3. Enable performance databases to assign factors causing trajectory inefficiency metrics similar to the manner in which factors are assigned to reportable delay in OPSNET [25] or airline reported delay in ASQP [74].

To achieve these goals, we have proposed two mechanisms that link the en route inefficiency to different causal factors, and developed a methodological framework to model these mechanisms. To summarize, the main contributions of this work are threefold.

1. Develop a trajectory clustering algorithm to identify nominal air traffic flows and construct route choice sets. Use the route choice set to build Logit-based models that model the flight route choice process.

2. Develop efficient tree-based matching algorithms that map different causal factors to 4D flight trajectories.

3. Estimate contributions of different factors to en route inefficiency through counterfactual analysis.

The remainder of this chapter is organized as follows. In Section 3.2, we introduce the data sources and review our previous work on macro variations of en route inefficiency. Section 3.3 describes our modeling philosophy and framework of mapping different causal factors to flight routes. Section 3.4 applies statistical models to quantify the impacts of different factors on flight en route inefficiency. Section 3.5 offers conclusions.

3.2 Preliminaries

In the section, we first introduce sources of datasets used in this work. We then present exploratory data analysis results from these datasets. Lastly, we propose two mechanisms that ascribe flight en route inefficiency to convective weather, wind, miles-in-trail restrictions, airspace flow programs, and special activity airspace.

3.2.1 Data Sources and Summary Statistics

In this project, we used seven datasets from different sources. The *flight event data*, which come from the FAA Traffic Flow Management System (TFMS) and from the Aviation System Performance Metrics (ASPM), contain flight-level records with variables including arrival and departure airports, aircraft type, D40 to A100 actual distance (A), D40 to A100 great circle distance (D), and D40 to A100 achieved distance (H). We obtained flight event data for years 2013 and 2014 and found an average inefficiency of approximately 3.4% across both years. This data set includes around 12 million records, 87% of which are domestic flights and less than 1% are either diverted flights or have missing records. We focused on flights arriving at and departing form the 34 main US airports listed in Appendix A. This list includes the 30 core US airports and an additional four high-traffic ones. The 34 main airports collectively handle around 3 million flights per year, which account for about 50% of the total number of flights in the US.

The *flight tracks dataset*, which also comes from TFMS, contains information about the position and movement of each aircraft throughout its flight. Some fields of interest include: latitude, longitude, altitude, ground speed, and time, with an interval of approximately one minute between two consecutive track points. In this research, we only obtained flight tracks between the eight airport pairs listed in Table 3.1 for the 2013 calendar year. In the preprocessing step to remove erroneous trajectories, we excluded tracks where spatial or temporal discontinuities were detected and ones that started or ended outside of the selected terminal areas. Figure 3.1 shows the historical flight tracks (projected on 2D space) from the George Bush international airport (IAH) to Boston Logan international airport (BOS) in 2013.

Our *convective weather data* contain two separate datasets. The first was derived from the quality controlled local climatological data (QCLCD), which were obtained from the national oceanic and atmospheric administration (NOAA). The dataset includes hourly summaries for convective weather conditions like thunderstorms, rain, hail, etc. at 1,600 ground weather

Table 3.1: Summary statistics for representative city pair inefficiency analysis

Airport Pair	Number of Records	Average Inefficiency
IAH to BOS	1617	4.37%
BOS to IAH	1742	2.58%
FLL to JFK	4011	3.59%
JFK to FLL	4043	3.06%
SFO to MCO	746	2.56%
LAX to CLE	875	1.69%
EWR to LAS	1726	2.07%
SFO to MSP	2644	1.11%

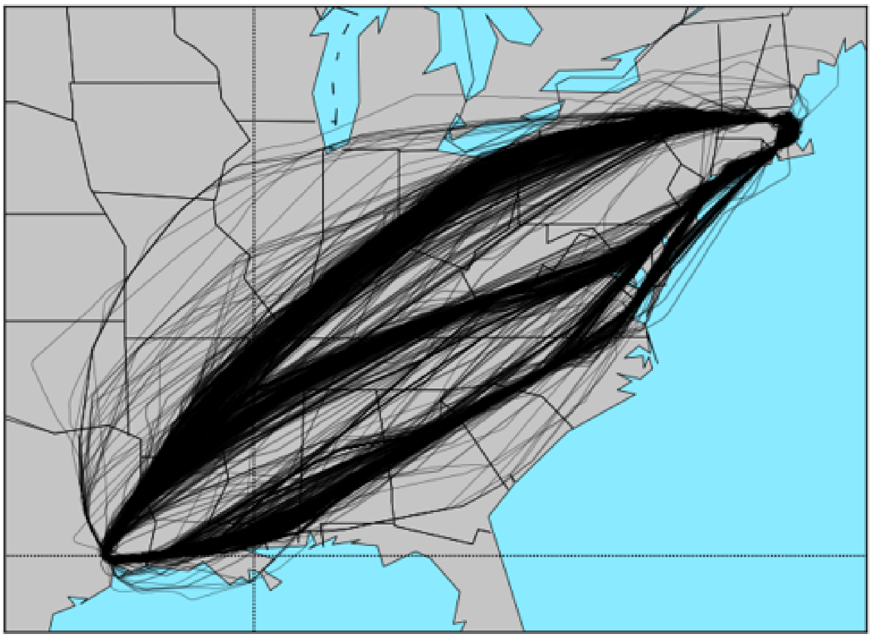


Figure 3.1: Flight tracks from IAH to BOS in 2013

stations in the US. Each record is a vector of binary variables indicating if there was a certain type of weather (e.g., thunderstorms, rain, etc.) occurring at a specific time and location. QCLCD were collected for the same time period as the flight track dataset. Figure 3.2 shows a snapshot of the QCLCD data from 18:00 to 21:00 Zulu on 08/08/2013. The green dots are the ground-based weather stations, while blue stars ("∗") and red crosses ("+") represent stations that reported rain and thunderstorm incidents, respectively. The second convective weather dataset comes from the national convective weather forecast (NCWF) system, which

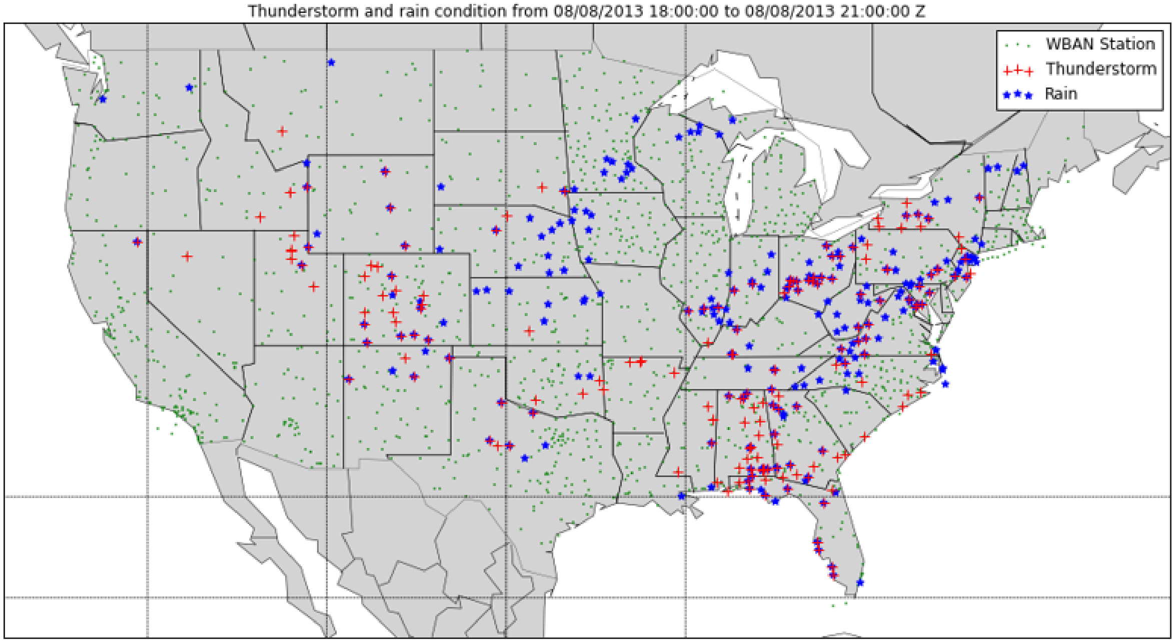

Thunderstorm and rain condition from 08/08/2013 18:00:00 to 08/08/2013 21:00:00 Z
WBAN Station
Thunderstorm
Rain

Figure 3.2: QCLCD snapshot data from 18:00 to 21:00 Zulu on 08/08/2013

contains the locations of convective weather polygons (coordinates of boundaries), the highest altitude of the storm, and the direction of movement at the time of recording. The dataset is typically updated every 5 minutes.

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. At each data production instance, the wind speed is given at a 3D grid with 2.5° × 2.5° latitude/ longitude resolution, and 17 pressure levels ranging from 10 mbar to 1000 mbar ([41], [9]). By collecting the wind data from the year 2013, we obtained 1460 raster files, and each file corresponds to 17 wind maps at different altitudes. Figure 3.3 shows a wind speed map at altitude 200 mbar at 18:00 Zulu on 02/04/2013. Each arrow is the wind direction computed based on the westerly and southerly wind speeds, and the shade indicates the speed of the wind.

The *miles-in-trail* (MIT) dataset was derived from the national traffic management log (NTML). It contains information about where, when, and why the MITs were implemented. For the MIT restriction data, we are interested in the fields including a) providing/requesting facility for the MIT, b) NAS element to enforce the MIT, c) start/end time of the restriction, d) altitude of the MIT, and e) the actual MIT spacing parameter (in miles). For this study, we excluded restrictions that either a) were cancelled before their initiation time, or b) had spacing parameters less than 0. We have also merged MITs that were extended or modified from their initial implementation. Lastly, we only included restrictions with known

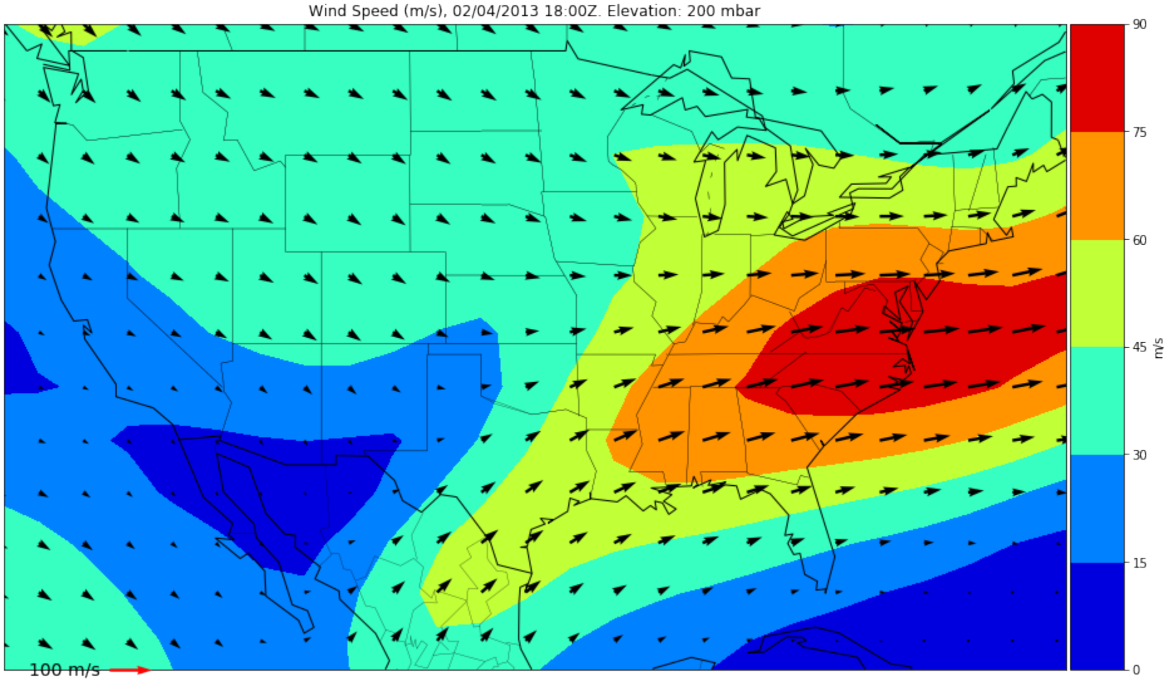


Figure 3.3: Wind Speed Map (m/s) at 200 mbar Derived From the NCAR Datasets at 18:00 Zulu on 02/04/2013

Table 3.2: Summary statistics for MIT restrictions

	Total	En Route
Cancelled Before Initiation Rate	2.80%	2.60%
Number Initiated	186,184	81,294
Modification Rate (Extended or Modified)	21.30%	19.00%
Average Duration	2.007 hours	1.861 hours
Average MIT Value	18.67 miles	20.52 miles
Total Number of MIT after cleaning	151,545	63,142
Percent lost by preprocessing	18.60%	22.33%

geometries of their facilities and NAS elements provided by the FAA. Figure 3.4 illustrates an actual MIT implemented at the NAS element ONL (as in red circle) from the Denver air route traffic control center (ARTCC ZDV, as in yellow polygon) to protect the Salt Lake City ARTCC (ZLC, as in green polygon), such that the flight in white curve will be subject to spacing at ONL. Table 3.2 provides summary statistics about MIT restrictions in 2013.

The *airspace flow programs* (AFP) data also come from the NTML and contain information about where, when and why the AFP were implemented, and if implemented, what

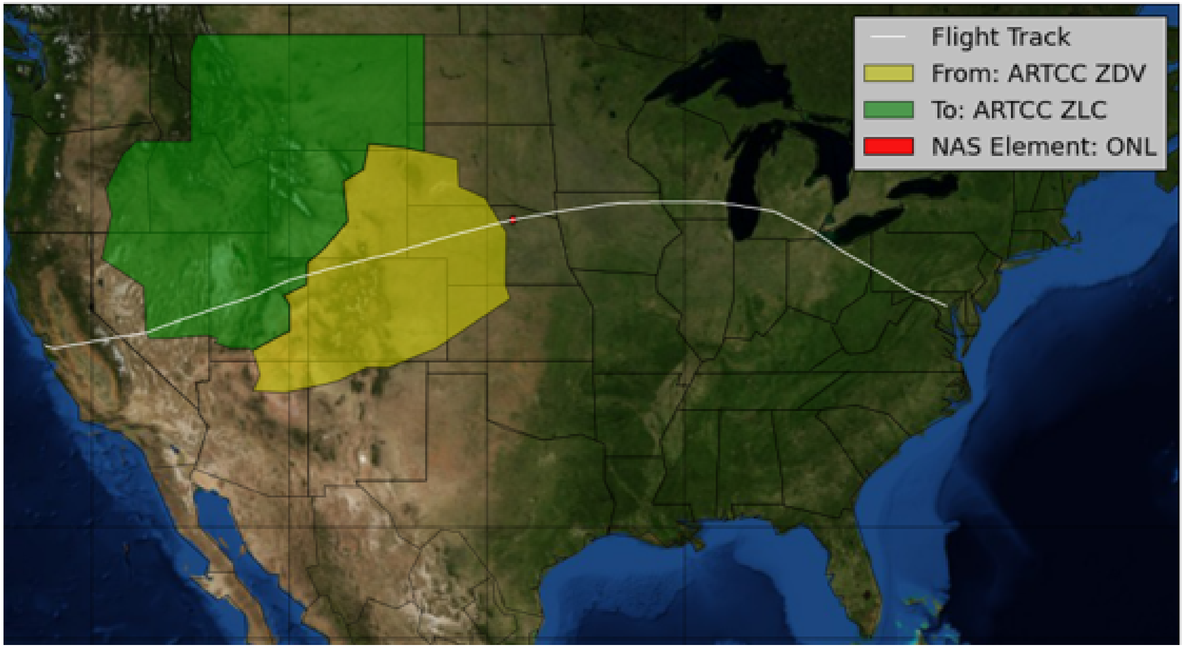


Figure 3.4: An Actual Miles-In-Trail Implemented at ONL From the Denver ARTCC (ZDV) to Protect the Salt Lake City ARTCC (ZLC)

the arrival rate and expected delays were. We have applied similar preprocessing steps as in the MIT dataset, and ended up with 118 AFPs in 2013. Figure 3.5 shows two sets of AFP respectively implemented on 04/18/2013 and 07/10/2013. The lines are flow constrained areas (FCAs), which control the arrival rates and delays for flights crossing them.

The *special activity airspace* (SAA) data come from the NASA Sherlock data warehouse. These data describe the start and end time for every active SAA in the airspace. Figure 3.6 presents the spatial distribution of the following six most popular SAAs in the airspace ([23]):

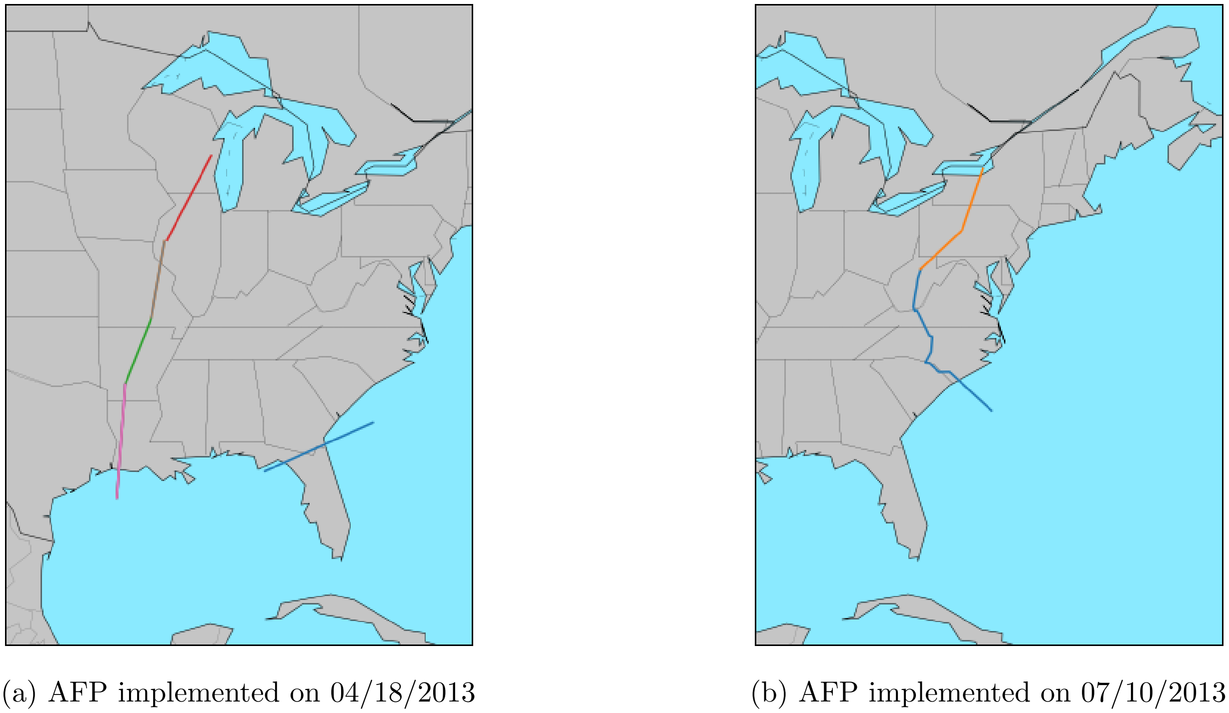
1. Alert area (AA): Areas that “may contain a high volume of pilot training or an unusual type of aerial activity”. Pilots “should be particularly alert” when flying in AAs.

2. Restricted area (RA): Areas with unusual hazards to aircraft such as “artillery firing, aerial gunnery, or guided missiles”. Flights within the area, “while not wholly prohibited, is subject to restrictions”.

3. Prohibited area (PA): Areas established for reasons “associated with the national welfare”, such that flights are prohibited to enter.

4. Military operations area (MOA)[1]: Areas established to separate military activities from

[1] Notice that in Figure 3.6, most MOAs are overlapped with the ATCAAs but have lower altitudes



(a) AFP implemented on 04/18/2013

(b) AFP implemented on 07/10/2013

Figure 3.5: Actual AFP data

instrument flight rules (IFR) traffic. IFR flights can enter the MOAs if separation can be provided by air traffic control; otherwise, they need to route out the MOAs. MOAs typically locate at altitudes lower than 18,000 ft.

5. Air traffic control assigned area (ATCAA): Areas assigned by air traffic control for the purpose of "providing air traffic segregation between specified activities conducted within the assigned airspace and other IFR air traffic". ATCAAs typically locate at altitudes from 18,000 to 60,000 ft.

6. Warning area (WA): Areas that are "three nautical miles outward from the coast of the US", which "warn nonparticipating pilots of the potential danger".

3.2.2 Motivations from Previous Work

In Chapter 2, we applied fixed effects regression techniques to a two-year flight-level performance dataset to quantify how departure/arrival airports and seasons impact en route inefficiency. Consistent with the literature, our estimation results suggest that long-haul flights are more efficient than short-haul flights. Additionally, we find that flights occurring

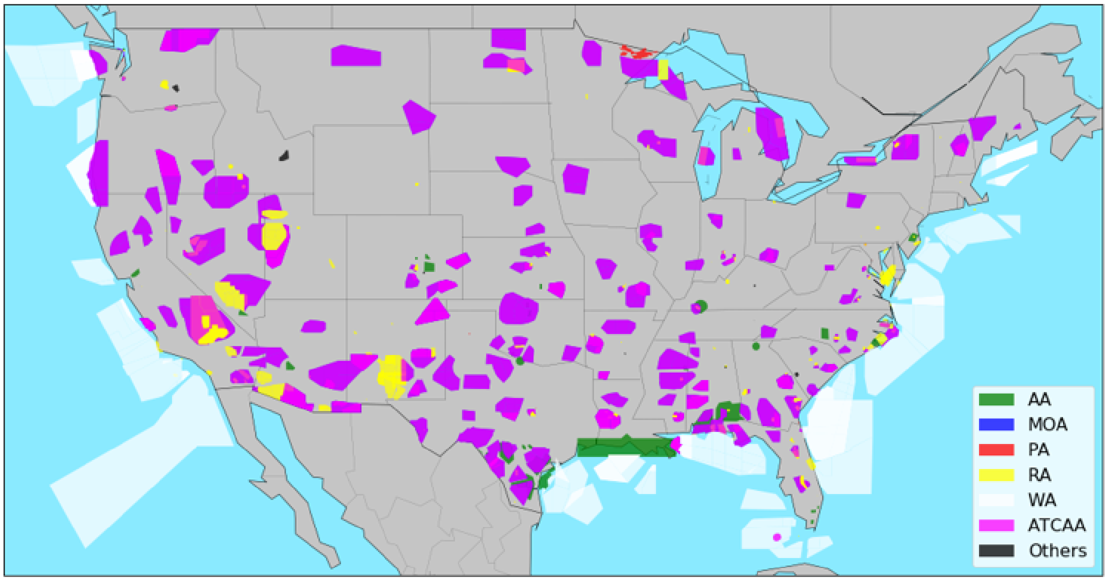


Figure 3.6: Spatial Distribution of Special Activity Airspace

in the summer season, when convective weather is more frequent, are less efficient than flights during other seasons.

A deficiency of this work, however, is that our models only explain a low percentage of the variance in flight inefficiency. We have also observed that variations for a specific airport pair, even within the same month, are quite pronounced, suggesting the need for models that consider circumstances specific to each flight. On the other hand, the historical flight track data reveal the fact that flights for given OD pairs are drastically different from each other. For example, Figure 3.1 demonstrates clear routing patterns, which consist of three dominant flows and some outlier flights having unusual trajectories. These flows are inherently different from each other with respect to their routing inefficiencies. This observation suggests that while variations in flight en route inefficiency are the result of weather, TMIs and SAAs, they are not directly attributed to these factors but arise from a two-step decision process – strategic route selection and tactical reroute. In other words, flight operators first choose from a route set with different base inefficiencies and utilities constituted of weather, wind, TMIs, and SAAs. Second, when flights are airborne, they may take tactical reroutes (e.g., vectoring, holding, and etc.) as responses to convective activities, TMIs and so on. These two mechanisms are depicted in respectively Figures 3.7 and 3.8. In the following sections, we will introduce our methodological framework to quantitatively analyze how these factors (i.e., weather, wind, TMIs, and SAAs) affect the en route inefficiency through the proposed two mechanisms.

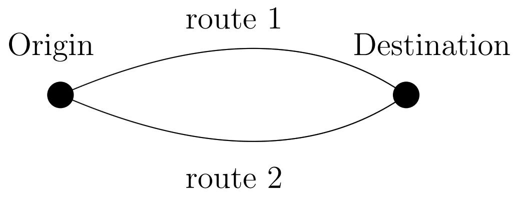


Figure 3.7: Route Choice Mechanism

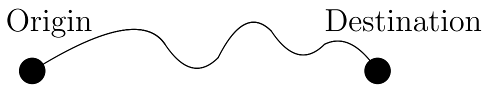


Figure 3.8: Reroute Mechanism

3.3 Nominal Routes, Attribute Identifications and Matching Algorithms

Our analysis is based on the concept of nominal route. Flight trajectories are unique, but most flights for a given origin-destination (OD) pair fall into one of several groups, or clusters. Trajectories within a cluster are quite similar to one another. Among the trajectories that form a given cluster, one can identify a specific trajectory that is the "center" of the cluster, which we term as the nominal route. Much of the variation in flight inefficiency can be attributed to variation in the inefficiency of the nominal routes. Moreover, one can use nominal routes as a basis for determining cluster attributes (e.g., exposure to convective weather) that help explain to which cluster a given flight belongs. As will be shown, this can provide an approach to the main problem addressed by this chapter: relating en route inefficiency to causal factors through both strategic flight planning and tactical rerouting. To achieve this goal, we pursue the following three objectives:

1. Design a trajectory clustering algorithm to classify flight tracks;
2. Identify the nominal route for each cluster;
3. Determine, for the departure time of a given flight, the relevant attributes arising from a collection of factors of each nominal route.

3.3.1 Trajectory Clustering

There is an extensive literature discussing the application of clustering algorithms to flight trajectories. To efficiently classify high-dimensional flight tracks into unique groups, Eckstein ([16]) proposed a three-step framework: step 1) linear interpolation, step 2) principal component analysis (PCA), and step 3) k-means [48] on the principal components. Enriquez and Kurez ([18]) improved the work of [16] by constructing a Gaussian kernel dissimilarity matrix and applying a variant of spectral clustering algorithm in the third step. However, such a method is computationally expensive. Other methods such as waypoint-based clustering ([32]) and partition-and-group framework for trajectory clustering ([46], [45]) can be applied in some circumstances requiring higher granularity (e.g., trajectory clustering in the terminal airspace). Interested readers may refer to [17] and [76] for further discussions on the topic of trajectory clustering.

For purposes of simplicity and efficiency, in this work, we will follow the framework of [16], but switch from a k-means algorithm to DBSCAN ([19]) in the final step for three reasons:

1. DBSCAN does not need us to predetermine the number of clusters;
2. DBSCAN allows trajectories to be classified as outliers (i.e., not belonging to any cluster);
3. DBSCAN can limit the variation within each cluster.

After the clusters are determined, we further calculate the nominal route for each cluster by solving a 1-median problem [51], and defining the set of nominal routes as a route choice set. Notice that each nominal route is a specific 4D trajectory that contains a sequence of latitudes, longitudes, altitudes, and timestamps. A brief summary of the algorithm is shown in Algorithm 1. Lines 3 – 6 describe the interpolation. Lines 7 is the PCA applied to the interpolated flight trajectories. Line 8 applies DBSCAN to the principal components, and lines 9 – 11 solve the 1-median problem to obtain nominal routes.

We have applied our trajectory clustering algorithm to the eight airport pairs listed in Table 3.1. The parameter n (number of points to be interpolated) is set to be 100 and k (number of principal components) is set to be 4 such that the variance explained by the first four principal components exceed 85%. Parameters (ε, p) in the DBSCAN algorithm are selected through experiments and are based on the resulting number of natural clusters, sizes of clusters, and inter-cluster similarities. The results are presented in Figures 3.9 to 3.16. In each figure, the left subfigure shows the actual horizontal flight tracks clustered into groups. Each right subfigure demonstrates a violin plot, with three caps from top to bottom: maximum, median and minimum, showing the range and probability density of flight en route inefficiency within each cluster. The x-axis of each violin plot also shows the percentage of total flights in each cluster. In each figure, different colors represent different clusters, and the color code is as follows: “k: black; r: red; g: green; m: magenta; c: cyan; b: clue”. The black trajectories are classified as outliers, and the white curves are the nominal routes associated with different clusters. From these figures, we first observe that for most airport pairs in our experiments, individual flights can be divided into 3 – 5 natural clusters (except for SFO to MSP). Members of each cluster are very similar to each other. The outlier cluster typically has low proportion of flights. Second, the distributions of en route inefficiency of natural clusters are typically unimodal with lower variance when compared with the outlier clusters. We have also summarized in Table 3.3 the average inefficiency statistics for each cluster and for each airport pair, in which the first row depicts the color code of natural clusters. Meanwhile, Table 3.4 reports the variance of inefficiency captured by the clusters.

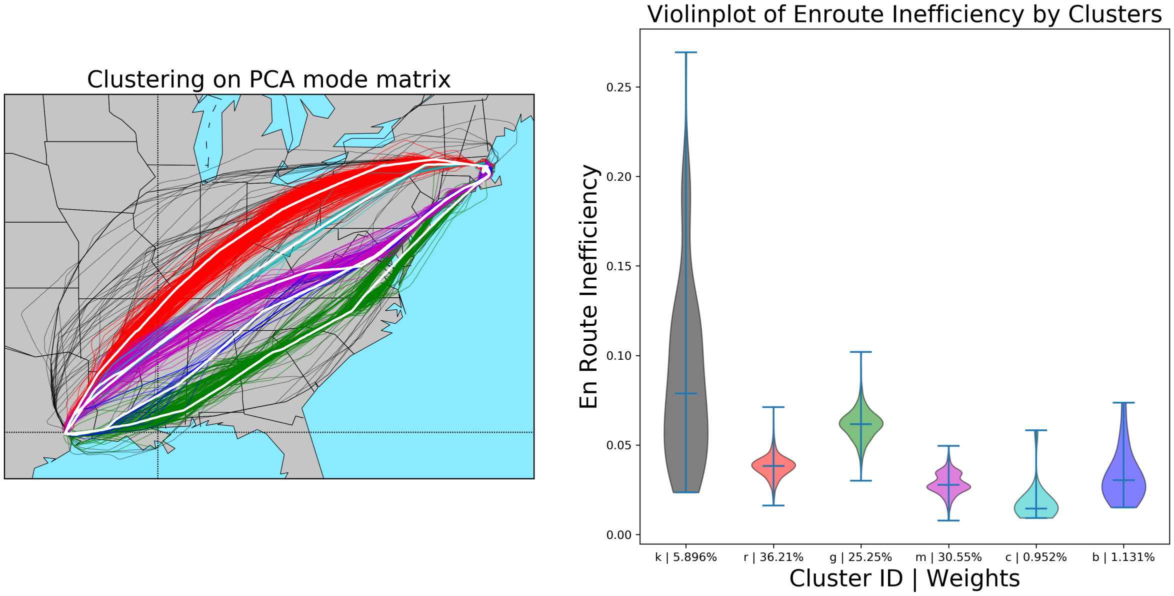


Figure 3.9: Trajectory Clustering for Flights from IAH to BOS

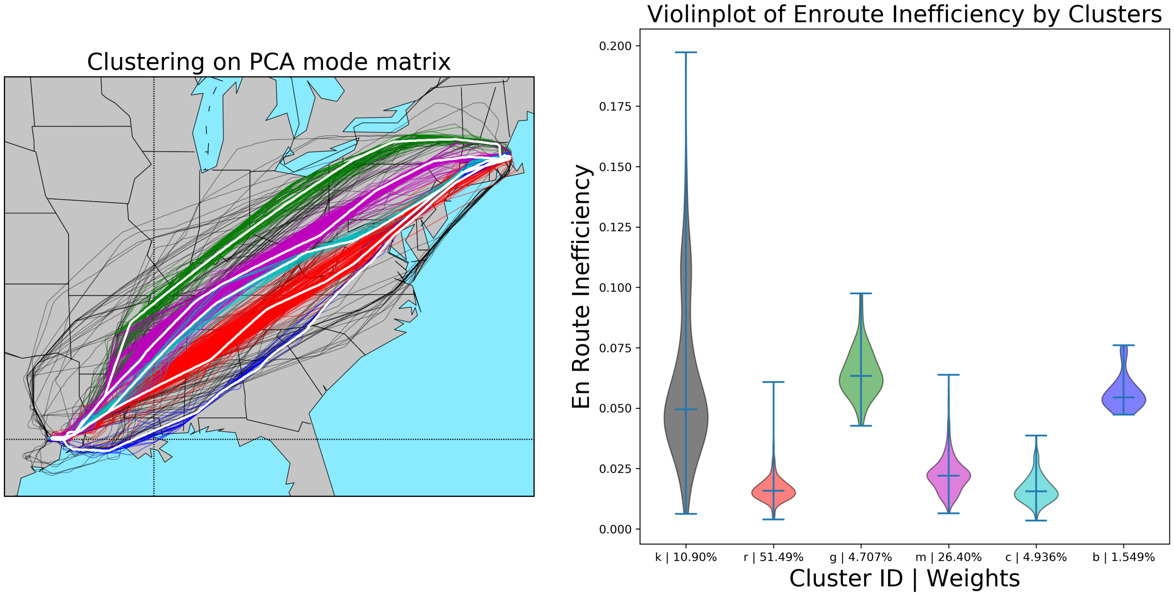


Figure 3.10: Trajectory Clustering for Flights from BOS to IAH

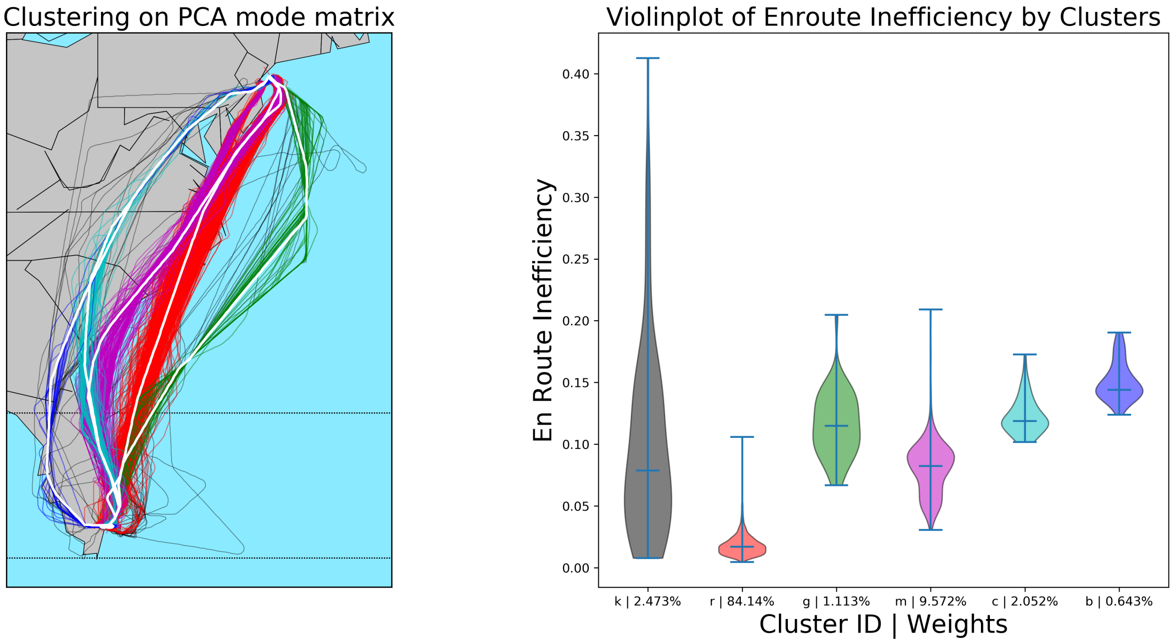


Figure 3.11: Trajectory Clustering for Flights from JFK to FLL

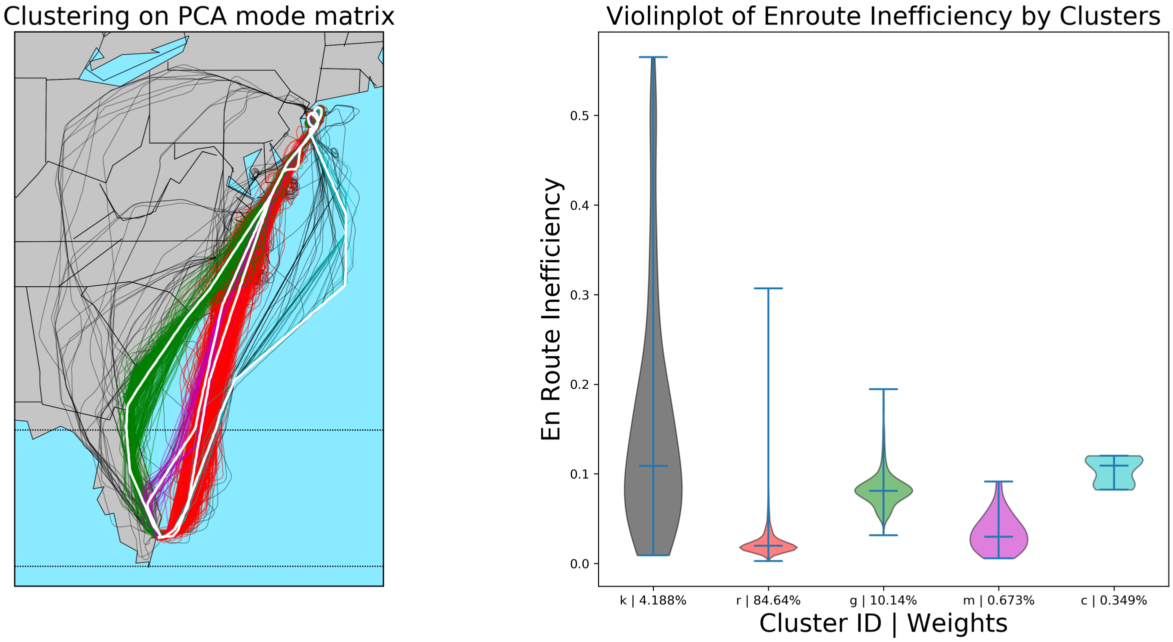


Figure 3.12: Trajectory Clustering for Flights from FLL to JFK

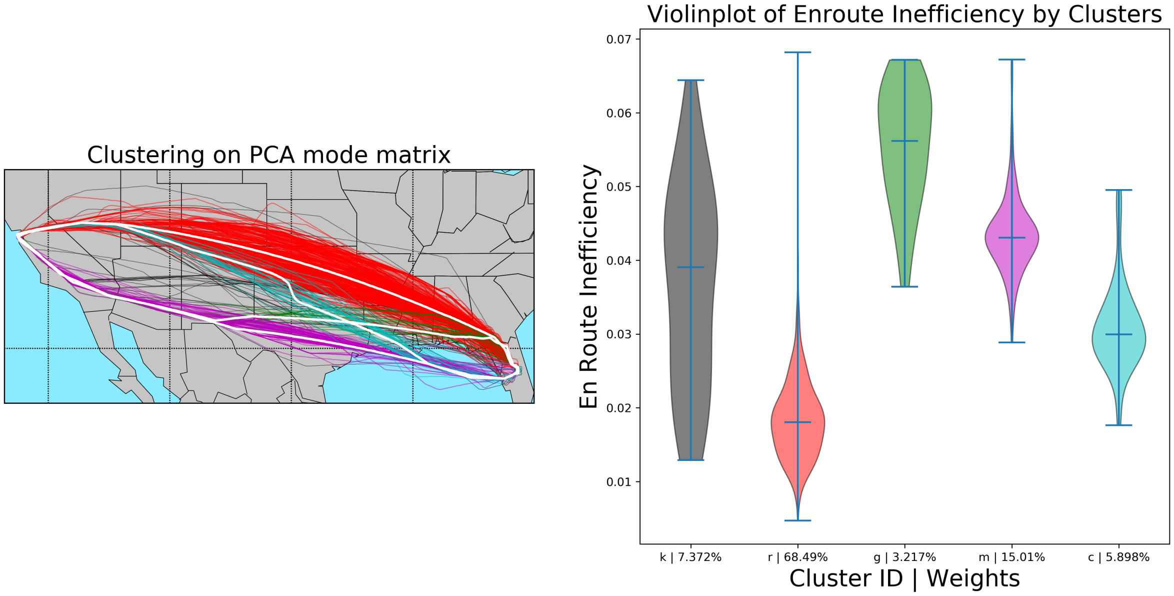


Figure 3.13: Trajectory Clustering for Flights from SFO to MCO

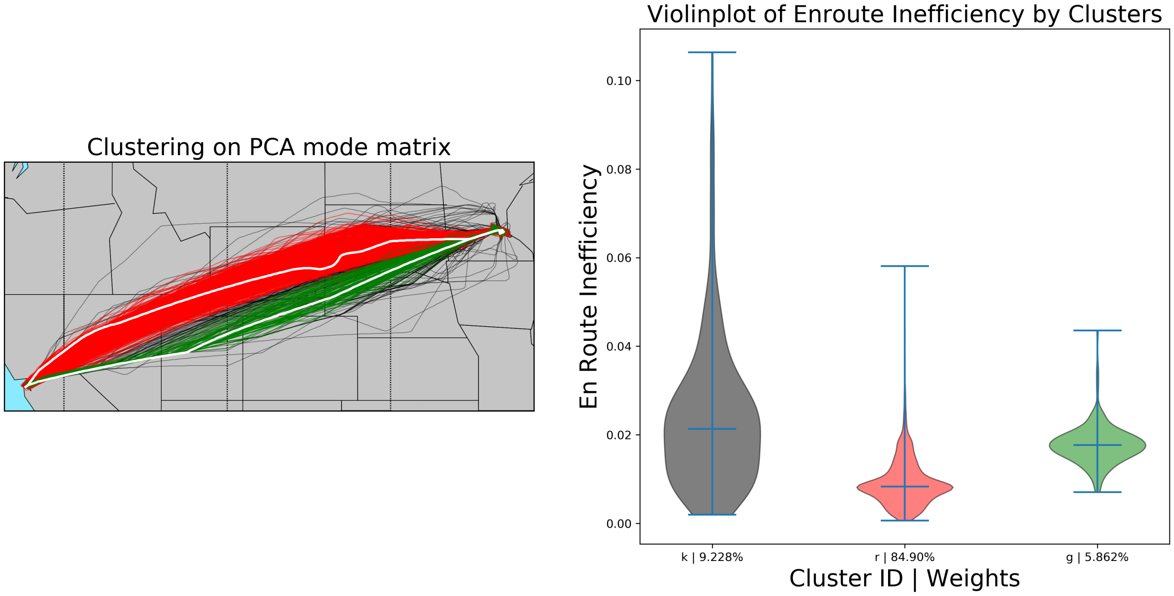


Figure 3.14: Trajectory Clustering for Flights from SFO to MSP

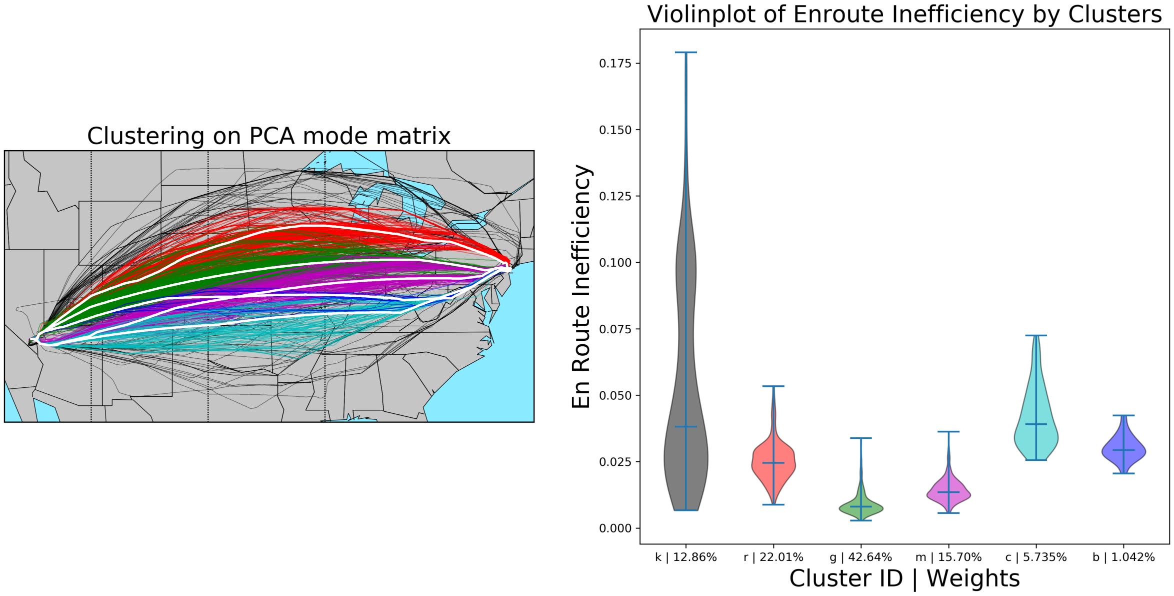


Figure 3.15: Trajectory Clustering for Flights from EWR to LAS

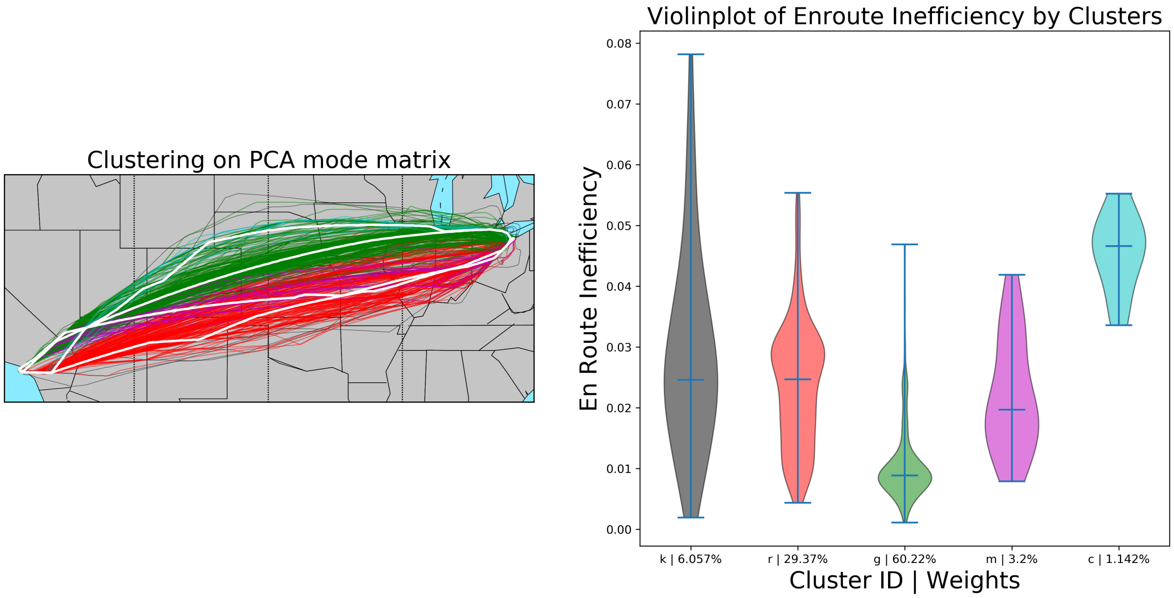


Figure 3.16: Trajectory Clustering for Flights from LAX to CLE

Algorithm 1 Trajectory Clustering (TC)

```
Inputs
    $\mathbb{P} = \big\{\mathbf{P}_i = [P_1, \dots, P_T], \forall i \in \{1, 2, \dots N\}\big\}, P_l = [x_l, y_l, z_l, t_l]$: 4D flight trajectory set.
Parameters
    $n$: Number of points to be interpolated;
    $k$: Number of principal components selected;
    $\varepsilon$: Distance threshold to be considered as in the same cluster in the DBSCAN algorithm;
    $p$: Minimal samples required to be a cluster in the DBSCAN algorithm.
Outputs
    Number of clusters, cluster assignment, and nominal routes $\{\mathbf{P}_j, j \in \{1, 2, \dots, m\}\}$.
procedure TC
    Initialize: $\mathbb{P}' = \Phi$
    for $i$ in $\{1, 2, \dots, N\}$ do
        Linearly interpolate trajectory $\mathbf{P}_i$ to $\mathbf{P}'_i$ that has $n$ points with equal distance.
        Append $\mathbf{P}'_i$ to $\mathbb{P}'_i$.
    end for
    Apply PCA on $\mathbb{P}'_i$ and obtain the first $n$ principal components $\mathbf{T}$.
    Apply DBSCAN$(\varepsilon, p)$ algorithm to $\mathbf{T}$ and obtain cluster set $\mathbb{M} = \big\{\Omega_i = \{\mathbf{P}_j, j \in \{1, \dots, m_j\}\}, \forall i \in \{1, 2, \dots, m\}\big\}$.
    for $j$ in $\{1, 2, \dots, m\}$ do
        Solve $\mathbf{P}_j = \arg\min \sum_i^{m_j} \|\mathbf{P}_i - \mathbf{P}\|$
    end for
end procedure
```

3.3.2 Mapping Framework

We start this section by introducing the general ideas behind the methodology that connects individual trajectories to the causal factors behind inefficiency. In Section 3.3.1, we have constructed a set of clusters and the corresponding cluster assignment for any given flight in our dataset. Therefore, for each flight, we can construct a scenario whereby the flight operator needs to ***choose*** a route from a choice set formed by natural clusters. Essentially, we are asking, for any flight and for each of the clusters, what would likely have happened to that flight, with respect to convective weather, wind, traffic management initiatives and special activity airspace, if it had flown that cluster with its original departure time. This is what we refer to as the first mechanism to associate flight inefficiency with the identified causal factors.

An immediate question posed by our proposed mechanism is, however, how do we arithmetically link the **choice set** with any of those causal factors? One naïve alternative would be for each flight, to use the full dataset of predictors covered by the flight time period.

Table 3.3: Average Inefficiency within Clusters

	K (outlier: black)	R (red)	G (green)	M (magenta)	C (cyan)	B (blue)
IAH to BOS	8.78%	3.80%	6.14%	2.84%	1.76%	3.26%
BOS to IAH	6.00%	1.65%	6.48%	2.28%	1.66%	5.63%
FLL to JFK	15.22%	2.41%	8.36%	3.42%	10.38%	
JFK to FLL	10.00%	1.85%	11.59%	8.05%	12.28%	15.09%
SFO to MCO	3.68%	1.86%	5.55%	4.35%	3.09%	
LAX to CLE	2.90%	2.39%	1.06%	2.21%	4.54%	
EWR to LAS	5.32%	2.54%	0.93%	1.44%	4.18%	3.01%
SFO to MSP	2.39%	0.92%	1.78%			

Table 3.4: Variance Explained by Clusters

	IAH BOS	BOS IAH	JFK FLL	FLL JFK	SFO MCO	LAX CLE	EWR LAS	SFO MSP
Variance Explained by Clusters	0.596	0.641	0.75	0.499	0.722	0.441	0.516	0.285

However, this runs the risk of overfitting and no unique estimates can be found for most regression models. For example, a normal flight from IAH to BOS takes around three hours of airborne time. Within the time period, the dimensions for convective weather alone would be approximately $2482 \times 3 \times 7 \approx 50000$, where 2,482 is the number of ground weather stations, 3 is the time instance (a 3-hour flight matches roughly 3-hour weather instances), and 7 is the number of weather phenomena. However, the total number of flights from IAH to BOS is 1617, which is much smaller than the dimensions of the naïve predictors. Therefore, we need to construct highly descriptive predictors that not only represent the conditions in the relevant airspace, but have controllable dimensions. To tackle this issue, we have identified a specific 4D trajectory with each cluster (lines 9 – 11 in Algorithm 1). This is treated as the "center" of the cluster, and we term it the *nominal route*. These routes are representative for the associated clusters so that we can use them as the bases for determining cluster predictors that help us understand to which cluster a given flight belongs. Specifically, the nominal routes in the choice set differ only in the route-specific characteristics, such as convective weather exposure, wind and TMIs, but shares the same flight-level attributes, such as departure time and airline. That being said, even though the nominal routes as originally

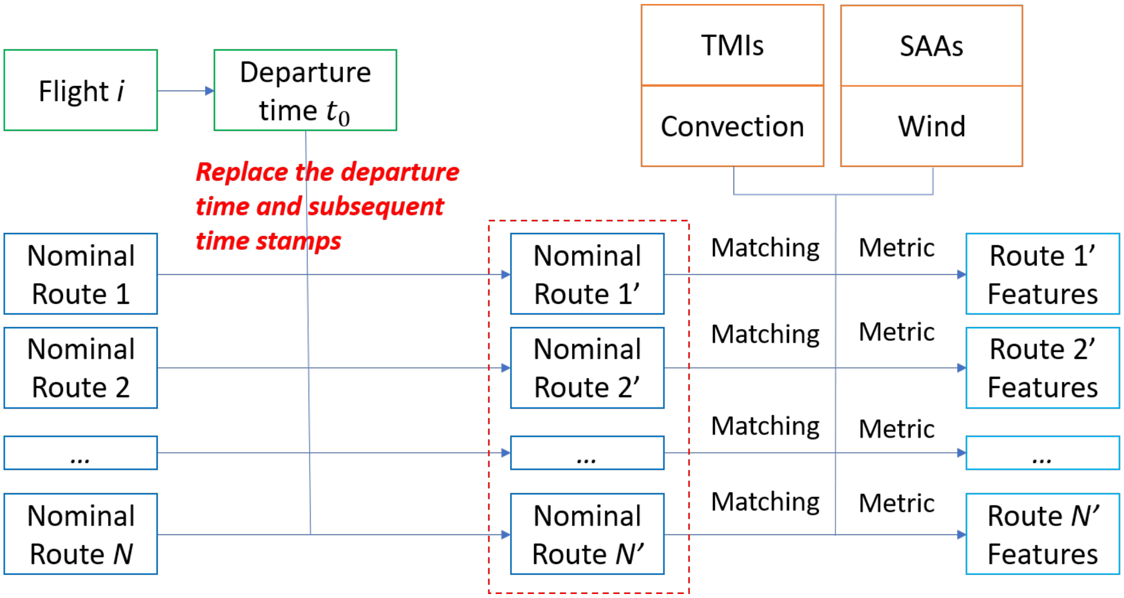


Figure 3.17: Illustration of the Mapping Framework

identified have their unique departure times and flight operators, we assume they have a common departure time and operator. These are specified by the flight whose cluster assignment we are modelling, and their route-specific characteristics are also based on the common departure time.

Figure 3.17 and Algorithm 2 summarize the general process and pseudo code of our mapping framework to relate the factors of interest with individual flights and the nominal routes. In the figure, we describe a scenario where flight i needs to chose from N nominal routes. Therefore, after extracting departure time t_0 of flight i, we can replace the timestamps of all N nominal routes as if they depart at t_0. This step corresponds to lines $2-9$ in Algorithm 2. After replacing timestamps, we further match those nominal routes with wind, weather, TMI and SAA datasets to construct route-specific characteristics. This corresponds to line 12 in Algorithm 2 and the details of this step will be introduced in the following sections.

3.3.3 Convective Weather Mapping Algorithm

Matching QCLCD Weather Data

The QCLCD convective weather records obtained from NOAA are recorded at individual ground-based weather stations. They are represented as time-stamped vectors of the form $[I_t, I_r, I_s, \ldots]$, where the binary indicator variables $\{I_i\}$ represent the presence or absence of different weather phenomena, including thunderstorms, rain, etc. For a given flight $\mathbf{P}_i$ and its associated nominal routes $\big\{\mathbf{P}_j, \forall j \in \{1, 2, \ldots m\}\big\}$, we compute the QCLCD weather

Algorithm 2 Mapping Framework (MF)

```
    Inputs
        $\mathbf{P}_i = [P_1, \dots, P_T], P_l = [x_l, y_l, z_l, t_l]$: Flight to be modeled.
        $\{\mathbf{P}_j = [P_1, P_2, \dots, P_{T_j}], \forall j \in \{1, 2, \dots m\}\}, P_l = [x_l, y_l, z_l, t_l]$: Set of nominal routes.
    Outputs
        Route characteristics of nominal routes associated with flight $\mathbf{P}_i$.
procedure MF
    Obtain $t_0$: Departure time of $\mathbf{P}_i$.
    Obtain $F_i$: Flight-level attributes of $\mathbf{P}_i$.
    for $j$ in $\{1, 2, \dots, m\}$ do
        Obtain $t_0^j$: Departure time of nominal route $\mathbf{P}_j$.
        Calculate: $\Delta t^j = t_0^j - t_0$.
        Set: $t_0^j := t_0$.    /* Replace departure time */
        for $l$ in $\{1, 2, \dots, T_j\}$ do
            Set: $t_l := t_l - \Delta t^j$.    /* Replace timestamp of nominal route */
        end for
    end for
    Match: use $\mathbf{P}_j$ with replaced timestamps to match with weather, wind, MIT, AFP,
and SAA.    /* See Sections 3.3.3 to 3.3.5 */
end procedure
```

exposure using Algorithm 3. From lines 4 to 7, we retrieve a subset from the original convective weather dataset for each track point of the nominal route that satisfies: a) the subset has the closest timestamp to the track point; and b) the subset contains weather information whose recording weather station is within a 150-nautical-mile radius around the track point. From lines 8 to 14, we calculate, for each track point on the nominal route, a *weighted* weather exposure. The weight is inversely proportional to the distance between the track point and the weather station. From line 16, we calculate the overall route-specific weather exposure by averaging all the weather exposures along the route. The average is used as the QCLCD weather metric for subsequent modeling.

Matching NCWF Weather Data

The NCWF weather records are preprocessed such that they are stored in a format similar to the QCLCD dataset. Therefore, the matching process largely agrees with Algorithm 3, except that we replace the input dataset from QCLCD with NCWF, and add an altitude range query right after line 6.

3.3.4 Wind Mapping Algorithm

The wind data from NCAR are stored as raster files. Each file corresponds to a single snapshot of wind vectors $\mathcal{V} = [u_{ijkt}, v_{ijkt}]$, where u_{ijkt} and v_{ijkt} are, respectively, the westerly and southerly wind speed vectors at the grid point $(lat, lon, alt) = (i, j, k)$ at time t. Similar

Algorithm 3 Mapping Convective Weather (MCW)

```
    Inputs
        $\mathbf{P}_i = [P_1, \ldots, P_T], P_l = [x_l, y_l, z_l, t_l]$: Flight to be modeled.
        $\{\mathbf{P}_j = [P_1, P_2, \ldots, P_{T_j}], \forall j \in \{1, 2, \ldots m\}\}, P_l = [x_l, y_l, z_l, t_l]$: Set of nominal routes.
        $\mathcal{D}_{wx}$: Convective weather database.
    Parameters
        $R$: Maximal query distance, default at 150 nm.
    Outputs
        $\{W_j\}$: Route-specific weather exposure of nominal routes associated with flight $\mathbf{P}_i$.
procedure MCW
    Run Algorithm 2 till line 11 to obtain nominal routes with replaced timestamps $\{\mathbf{P}'_j = [P_1, \ldots, P_{T_j}], \forall j \in \{1, 2, \ldots m\}\}$.
    for $j$ in $\{1, 2, \ldots, m\}$ do
        for $l$ in $\{1, 2, \ldots, T_j\}$ do
            Retrieve $\mathcal{D}_t$ by nearest neighbor searching $\mathcal{D}_{wx}$ with time $t_l$.
            /* This step uses k-d tree ([6]) based searching technique */
            Retrieve $\mathcal{D}_{t,p}$ by querying $\mathcal{D}_t$ within a distance range $R$ of $[x_l, y_l]$.
            /* This step uses k-d tree based range query technique */
            /* $(t, p)$ is tuple of time $t$ and 2D location instance $(x_p, y_p)$ */
            for $I_{t,p}$ in $\mathcal{D}_{t,p}$ do
                Calculate distance $d_{t,p} = \|(x_p - y_p) - (x_l, y_l)\|$
            end for
            for $I_{t,p}$ in $\mathcal{D}_{t,p}$ do
                Calculate weight $w_{t,p} = \frac{(\sum_p d_{t,p})/d_{t,p}}{\sum_p \left((\sum_p d_{t,p})/d_{t,p}\right)}$
            end for
            Calculate $W_l = \sum_p I_{t,p} \cdot w_{t,p}$: weather exposure at $t_l$.
        end for
        Calculate $W_j = \sum_l W_l / T_j$: weather exposure for nominal route $\mathbf{P}'_j$.
    end for
end procedure
```

to Algorithm 3, we construct the wind-related route-specific characteristics for the nominal routes using a tree-based searching technique. The details are described in Algorithm 4. To summarize, this algorithm first matches each track point on the nominal route with the spatially (latitude, longitude, altitude) and the temporally closest wind speed components (lines 4 – 7). Second, it projects the wind speeds (westerly and southerly) to the course of the aircraft flying at each timestamp and obtains either the headwind (specified as negative value) or tailwind (specified as positive value) speed (lines 8 – 10). Finally, the algorithm calculates the *wind distance* traveled along the nominal route by summing the products of tailwind/headwind speed and time (line 18), which measures the positive/negative contribution of the tailwind/headwind to the distance flown[2].

Algorithm 4 Mapping Wind (MW)

```
Inputs
    $\mathbf{P}_i = [P_1, \dots, P_T], P_l = [x_l, y_l, z_l, t_l]$: Flight to be modeled.
    $\{\mathbf{P}_j = [P_1, P_2, \dots, P_{T_j}], \forall j \in \{1, 2, \dots m\}\}, P_l = [x_l, y_l, z_l, t_l]$: Set of nominal routes.
    $\mathcal{D}_{wd}$: NCAR wind database.
Outputs
    $\{WD_j\}$: Route-specific wind distance of nominal routes associated with flight $\mathbf{P}_i$.
procedure MW
    Run Algorithm 2 till line 11 to obtain nominal routes with replaced timestamps $\{\mathbf{P}'_j = [P_1, \dots, P_{T_j}], \forall j \in \{1, 2, \dots m\}\}$.
    for $j$ in $\{1, 2, \dots, m\}$ do
        for $l$ in $\{1, 2, \dots, T_j\}$ do
            Retrieve $\mathcal{D}_t$ by nearest neighbor searching $\mathcal{D}_{wd}$ with time $t_l$.
            Retrieve $\mathcal{V} = [u_{ijkt}, v_{ijkt}]$ by nearest neighbor searching $\mathcal{D}_t$ with $[x_l, y_l, z_l]$.
            /* The above two steps use k-d tree based searching technique*/
            Calculate $\theta_l = \Theta(x_{l+1}, y_{l+1}, x_l, y_l)$: Course of the flight at $t_l$.
            /* $\Theta(\cdot)$ is standard course calculation formula */
            Calculate $v_l = u_{ijkt} \cdot \cos\theta_t + v_{ijkt} \cdot \sin\theta_t$: head/tail wind speed.
        end for
        Calculate $WD_j = \sum_{l=1}^{T_j} v_l \cdot (t_{l+1} - t_l)$: wind distance traveled along the route.
    end for
end procedure
```

3.3.5 Traffic Management Initiatives and Special Activity Airspace Mapping Algorithm

Miles-In-Trail Restrictions

The miles-in-trail (MIT) data from the NTML are assembled as individual data records. These have the structure $[st, et, f_{req}, f_{prov}, nas, z, v, \dots]$, where st and et are, respectively, the start and end time of the restriction; f_{req} and f_{prov} are the requesting and providing facilities; nas represents the NAS element where the MIT is enforced (it can be a jet route, TRACON or fix); z is the altitude restriction; and v is the actual mileage between subsequent aircraft imposed by the restriction.

For a given flight and its nominal routes, we compute the MIT-related impacts using the following steps.

1. **STEP 1:** Run Algorithm 2 to line 11 to obtain nominal routes with replaced timestamps.

[2] equivalent still air distance = ground distance - wind distance

2. **STEP 2:** Determine a set of candidate MIT restrictions that would have been within sufficient geographic proximity of that flight, and in its same overall direction of travel. This requires an exhaustive search of all MIT records. In a real implementation, however, one might pre-compute MIT membership in the airspace region, so as to take advantage of the results of lines 5 – 7 of Algorithm 3. In any case, each candidate MIT restriction considered must be assigned an influence area as follows:

    - If the NAS element is a fix, then the influence area is a circle of 0.25° latitude/longitude around the fix. This corresponds to about 15 nautical miles. A nominal route is considered to be a (matching) candidate for this MIT if the route intersects the circle in the same direction of travel for which the MIT is specified.
    - If the NAS element is a jet route, then we define a swath centered on the jet route with overall width 0.5°, which is about 30 nautical miles. A nominal route is considered to be a (matching) candidate for the MIT if the route intersects this swath for at least 1° offset of latitude/longitude, in the appropriate direction of travel.
    - If the NAS element is a center/TRACON, we use its existing polygonal boundary. The MIT is considered to be a (matching) candidate for a nominal route if the route intersects this polygon in the MIT restriction's direction of travel.

3. **STEP 3:** The list of candidate MIT restrictions is then filtered for timestamps. If the time of trajectory's intersection and the boundary of the MIT restriction is within the MIT's implementation interval $[st, et]$, then that restriction is retained as a candidate; otherwise it will be discarded.

4. **STEP 4:** The remaining list of candidate MIT restrictions is filtered for altitude. In our data, an MIT restriction is assigned to one of three possible altitude requirements: a) above a flight level z, b) below a flight level z, or c) at a flight level z. If a particular MIT involves cases a) or b), then we filter out that MIT if the nominal route's altitude at the time of intersection with the MIT influence area does not satisfy the requirement. In case c), we allow a 1000-foot buffer around the flight level z, and discard the MIT if the trajectory altitude does not fall within that altitude interval.

Notice that the filtering steps 2 – 4 can be done in any order, and it is wisest to do them in the most efficient order by starting with a strongest filter that eliminates the most MIT records, which is "step 3, step 4 and step 2" in our numerical experiment.

We consider that the flight in question, had it flown the route in question, would have been impacted by whatever MIT restrictions remain after steps 2 – 4. The magnitude of that impact is measured as the sum of the MIT stringency, which is defined as the product of MIT value (in miles) and duration (in hours), imposed by all the restrictions. This additive metric may be refined in subsequent research, as aircraft that are well spaced from one restriction may not need much additional spacing to comply with others, unless these involve additional traffic.

Airspace Flow Programs

The airspace flow programs (AFP) data from the NTML are recorded in a manner similar to the MIT dataset. Each AFP record is stored as a list $[st, et, \mu, f_{fca}, \ldots]$, where st and et are, respectively, the start and end time of the AFP; f_{fca} represents the flow constrained area (FCA) where the AFP is enforced (it can be either a polygon or a multiline); and μ is expected delay at the FCA. To match the AFP with a given flight and its nominal routes, we basically follow steps 1 to 3 of the procedure for the MIT. However, there is an exception at step 2, where we only consider the case where the FCA is either a polygon or line without the need to specify additional "influence area". We consider the impact of the AFP as the sum of the AFP expected delays imposed by all the matched AFPs.

Special Activity Airspace

The special activity airspace (SAA) data from the NASA Sherlock data warehouse also share similarities with the MIT dataset. Each SAA record contains the information $[st, et, f_{saa}, z_{\max}, z_{\min}, type, \ldots]$, where st and et are, respectively, the start and end time of the SAA; f_{saa} is the location of the SAA (typically a polygon); $z_{\max}$ and $z_{\min}$ specify the altitude range of the SAA; and *type* indicates the type of the SAA (e.g., military operations area). Thus, our matching process for the SAA is similar to the MIT matching process, except for the following modifications.

1. In **STEP 2**, we only need to consider the case where the SAA is a polygon.

2. In **STEP 2**, we create a swath centered on the nominal route with overall width 0.5° latitude/longitude and intersect it with the SAA polygon(s). This is because we have always ended up with no matches by matching the bare flight route with any SAAs due to the fact that flight routes often go around SAAs, even if the SAAs are not active. However, "no matches" does not imply "no impact" of SAA as flights need to extend their path by "going around SAAs".

3. After **STEP 4**, we also calculate the intersected area (in km^2) of flight route "swath" and SAA polygon(s), and use the sum of all the intersected areas (in km^2) as the metric to evaluate the impact of SAAs.

It should be noted that multiple thresholds, for example, the query distance R of convective weather and radius of NAS element r, have been chosen in the above mapping algorithm. While some thresholds, such as the radius of NAS element influence area, can be found in [53], we have performed sensitivity analysis on the others. For example, we have experimented with jet route width ranging from 0.25° to 0.75° of latitude/longitude in 0.25° increments. In general, the parameter values used here are the ones that yielded the best model performance in later modeling. However, the results obtained are not very sensitive across the range of parameters we have investigated.

We end Section 3.3 by summarizing the metrics of the identified causal factors for the subsequent statistical analysis.

1. Convective activities:
   - TS: Thunderstorm exposure (in percentage) to each nominal route. Thunderstorm is obtained from the NCWF dataset. Larger TS indicates more thunderstorm activities matched with the route.
   - W_x: Weather exposure (in percentage) to each nominal route. Weather includes four variables: rain, squall, ice, hail, and is obtained from the QCLCD dataset. Higher W_x suggests larger weather activities reported from the weather stations along the route.
2. Wind: wind distance traveled along the route (in 1000 nautical miles). Large positive wind distance indicates strong tailwinds, and large negative wind distance indicates strong headwinds.
3. MIT: sum of MIT stringency along the route (in 100 miles×hours). Larger MIT stringency suggests more severe MITs (could be either large MIT value, or MIT duration) matched with the route.
4. AFP: sum of AFP delays crossed by the route (in minutes). Larger AFP delays indicate longer expected delays assigned to the route.
5. SAA: sum of crossed SAA areas by the route (in 1000 km^2). Larger SAA areas suggest bigger overlap between the route "swath" and the SAAs.

3.4 Statistical Analysis

In this section, we first introduce the framework of how we statistically model the flight en route inefficiency through the proposed two mechanisms, and formulate the model specifications. Then we experiment with the data for the eight airport pairs listed in Table 3.1 and present the estimation results. Lastly, we conduct counterfactual analysis to estimate the contributions of different causal factors.

3.4.1 Modeling Philosophy

We establish two separate models to link causal factors to flight en route inefficiency, of which the first is a logit model that models the strategic route choice process, and the second is a linear model that models the tactical reroute.

Logit Model

To fit our use case within the random utility maximization framework ([4], [73]), here we have constructed a scenario whereby each flight i, prior to its departure, is a decision maker that needs to choose one nominal route $\mathbf{P}_j$ among m alternatives in the route choice set determined in Section 3.3.1. The "utility" U_{ij} that flight i obtains from choosing an alternative j is, though unknown to us, assumed to be a function of observed characteristics

($\mathbf{x}_{ij}$) of the alternative j and some attributes ($\mathbf{s}_i$) of the flight i, plus some unobserved random error ε_{ij} whose distribution needs to be further specified by us. In Equation 3.1, $V_{ij} = V(\mathbf{x}_{ij}, \mathbf{s}_i)$ is the assumed utility function, and the flight operator will choose alternative j that has the highest utility U_{ij}. The probability that corresponds to this utility is P_{ij}, as formulated below. $\mathbf{x}_{ij}$ and $\mathbf{s}_i$ will be further introduced in Section 3.4.2.

$$\begin{aligned} U_{ij} &= V_{ij} + \varepsilon_{ij} = V(\mathbf{x}_{ij}, \mathbf{s}_i) + \varepsilon_{ij} \\ P_{ij} &= Pr(U_{ij} > U_{ik}, \forall k \neq j) \end{aligned} \tag{3.1}$$

For discrete choice models, different error structures for ε_{ij} lead to different logit models. In this research, we start with the simplest, yet popular, *multinominal logit* (MNL) model by assuming that each ε_{ij} in Equation 3.1 is independently, identically distributed (IID) with Gumbel and type I extreme value distribution. The probability density function (PDF) is $f(\varepsilon_{ij}) = \exp(-\varepsilon_{ij}) \cdot \exp(-e^{\varepsilon_{ij}}), \forall j \in \{1, 2, \ldots, m\}$, which yields to a closed-form probability of flight i choosing alternative j shown below.

$$P_{ij} = \frac{e^{V_{ij}}}{\sum_{k=1}^{m} e^{V_{ik}}}$$

However, due to the independence from irrelevant alternatives (IIA) property, the MNL model suggests a proportional substitution across alternatives, which is not always the case. Specifically, we have observed some route alternatives from clustering results are geographically closer to each other. Therefore, to overcome this issue, we have experimented with *nested logit* (NL) model, in which m alternatives will be divided into K non-overlapping subsets, or nests. We denote these nests as $M_1, M_2, \ldots, M_K$. While the IIA property still holds for alternatives belonging to the same nest, it does not hold for alternatives from different nests. The error structure, $\varepsilon_i = [\varepsilon_{i1}, \varepsilon_{i2}, \ldots, \varepsilon_{im}]$, as a result, follows a joint distribution as in Equation 3.2, where the λ_k is a measure of independence among alternatives in nest M_k and needs to be estimated from the data. Notice that while $\varepsilon_{ij}, \forall j$ are not IID in general, they can be decomposed into two components ε_{ik} and ε_{ikj}, of which the later are IID in the same nest. This also explains why IIA property holds only within each nest. Lastly, we show in Equation 3.3 the probability of flight i choosing alternative j belonging to the nest M_k.

$$F(\varepsilon_i) = \exp\Bigg(- \sum_{k=1}^{K} \big(\sum_{j \in M_k} e^{-\varepsilon_{ij}/\lambda_k} \big)^{\lambda_k} \Bigg) \tag{3.2}$$

$$P_{ij} = \frac{\exp\left(\frac{V_{ij}}{\lambda_k}\right) \cdot \left[\sum_{l \in M_k} \exp\left(\frac{V_{il}}{\lambda_k}\right) \right]^{-1+\lambda_k}}{\sum_{q=1}^{K} \left[\sum_{l \in M_q} \exp\left(\frac{V_{il}}{\lambda_q}\right) \right]^{\lambda_q}} \tag{3.3}$$

Inefficiency Model

The inefficiency model predicts en route inefficiency, had we known which route a flight is going to fly and the measures of causal factors on *that route*. Therefore, we propose a linear regression, in which the dependent variable is the en route inefficiency, and the independent variables contain flight route assignment fixed effects, weather measurements, and etc., to quantitatively understand how different factors directly affect the flight en route inefficiency through tactical reroute. Notice that when we make predictions using the fitted linear model, the route assignment fixed effects will be the route choice probabilities predicted by the logit model. The detailed model specification is described in the next section.

3.4.2 Model Specification

Logit Model

In the logit model, the alternative set is formed by natural clusters (nominal routes), and a flight chooses a cluster to which it belongs. As mentioned in Section 3.4.1, the "utility" of flight i choosing route j can be written as the summation of an unobserved random error ε_{ij} and a deterministic utility function $V(\mathbf{x}_{ij}, \mathbf{s}_i)$, where $\mathbf{x}_{ij}$ is a vector of alternative-specific characteristics that differ from alternative to alternative for each flight; and $\mathbf{s}_i$ is a vector of flight-level attributes that only vary across flights. In this work, we have specified $\mathbf{x}_{ij}$ to include convective weather variables, wind distance, MIT stringency, AFP delays, and SAA influencing areas (see Section 3.3). We further specified $\mathbf{s}_i$ to contain five variables, including four departure time related variables (BH, Spr, Smr, Fall) in Table 3.5 and an alternative-specific constant (ASC).

Equation 3.4 shows the formula of our specifications for the logit model, in which the variable name code for $\mathbf{x}_{ij}$ is shown in Table 3.5. and coefficients $(\text{ASC}_j, \alpha_j, \zeta_j, \gamma_j, \theta_j, \beta_k)$ will be estimated using actual data. It should be noted that V_{im} only contains alternative-specific attributes to identify a proper logit model. Also notice that each of the alternative-specific attributes share the same coefficient (β's) across different alternatives. Based on the specification, we expect that the coefficients of the weather, MIT, AFP and SAA related variables should all have negative signs, since higher values of those covariates will always tend to degrade the attractiveness of a route. We also expect the sign of the wind distance coefficient to be positive since route with higher tailwind is in general favored.

$$\begin{aligned}
V_{i1} &= \mathrm{ASC}_1 + \alpha_1 \cdot \mathrm{BH}_i + \zeta_1 \cdot \mathrm{Spr}_i + \gamma_1 \cdot \mathrm{Smr}_i + \theta_1 \cdot \mathrm{Fall}_i \\
&+ \beta_{wd} \cdot \mathrm{WD}_{i1} + \beta_{mit} \cdot \mathrm{MIT}_{i1} + \beta_{afp} \cdot \mathrm{AFP}_{i1} + \beta_{saa} \cdot \mathrm{SAA}_{i1} \\
&+ \beta_{ts} \cdot \mathrm{TS}_{i1} + \beta_r \cdot \mathrm{R}_{i1} + \beta_q \cdot \mathrm{Q}_{i1} + \beta_{ice} \cdot \mathrm{Ice}_{i1} + \beta_h \cdot \mathrm{H}_{i1} \\
V_{i2} &= \mathrm{ASC}_2 + \alpha_2 \cdot \mathrm{BH}_i + \zeta_2 \cdot \mathrm{Spr}_i + \gamma_2 \cdot \mathrm{Smr}_i + \theta_2 \cdot \mathrm{Fall}_i \\
&+ \beta_{wd} \cdot \mathrm{WD}_{i2} + \beta_{mit} \cdot \mathrm{MIT}_{i2} + \beta_{afp} \cdot \mathrm{AFP}_{i2} + \beta_{saa} \cdot \mathrm{SAA}_{i2} \\
&+ \beta_{ts} \cdot \mathrm{TS}_{i2} + \beta_r \cdot \mathrm{R}_{i2} + \beta_q \cdot \mathrm{Q}_{i2} + \beta_{ice} \cdot \mathrm{Ice}_{i2} + \beta_h \cdot \mathrm{H}_{i2} \\
&\vdots \\
V_{im} &= \beta_{wd} \cdot \mathrm{WD}_{im} + \beta_{mit} \cdot \mathrm{MIT}_{im} + \beta_{afp} \cdot \mathrm{AFP}_{im} + \beta_{saa} \cdot \mathrm{SAA}_{im} \\
&+ \beta_{ts} \cdot \mathrm{TS}_{im} + \beta_r \cdot \mathrm{R}_{im} + \beta_q \cdot \mathrm{Q}_{im} + \beta_{ice} \cdot \mathrm{Ice}_{im} + \beta_h \cdot \mathrm{H}_{im}
\end{aligned} \tag{3.4}$$

Inefficiency Model

In the inefficiency model, the dependent variable is the flight en route inefficiency. There are five categories of independent variables in the model. The first category includes variables related to convective weather activities. Among the variables are thunderstorm (from the NCWF dataset), rain, squall, hail, and ice (from the QCLCD dataset). Since a flight may extend its route to avoid convection, we expect that those variables will have positive effects on the en route inefficiency. The second category of variables includes TMIs. Due to the fact that aircraft may be vectored or rerouted to comply with either the MIT restrictions or AFP, we expect that these variables will also have positive signs. The third category includes the SAAs. Since flight almost always go around SAAs, we expect this variable to have positive correlation with the inefficiency. The fourth set of variables relate to the departure time. We include both the departure seasons and departure busy hour indicator (8 am – 8 pm) as control variables to further capture the seasonal patterns and traffic effect, respectively. As indicated in [37], flights in the summer season tend to be more inefficient, and a higher volume of traffic in the airspace will also induce unnecessary maneuvers; thus, we expect that the estimate for summer season indicators should be systematically higher than the other seasons, and the estimate for the busy hour should also be positive. The last category pertains to the route structure. Figures 3.9 – 3.16 suggest that while the within-cluster en route inefficiency maintains a small variation, the cross-cluster inefficiencies tend to have significant difference. Therefore, we include the cluster memberships as the fixed effects in the model and use the "red" cluster as the baseline. We expect that the order of estimates for different clusters should match the order of the averages of within-cluster inefficiencies. The formula for the inefficiency model is shown in Equation 3.5, where the β's are the coefficients that need to be estimated. The explanatory variables and their notations are listed in Table 3.5.

Table 3.5: Description of Variables

Category	Notation	Variable description
Dependent variable	Y	En route inefficiency (in percentage)
Convective weather activities	TS	Thunderstorm exposure (in percentage)
	R	Rain exposure (in percentage)
	Q	Squall exposure (in percentage)
	Ice	Ice exposure (in percentage)
	H	Hail exposure (in percentage)
TMI	MIT	Sum of MIT stringency (in 100 miles × hours)
	AFP	Sum of AFP delays (in minutes)
SAA	SAA	Sum of overlapped area of special activity airspace and route swath (in km^2)
Departure time	BH	Busy hour indicator, equals 1 if local departure hour is between 8 am and 8 pm
	Spr	Equals 1 if departed between March to May
	Smr	Equals 1 if departed between June to August
	Fall	Equals 1 if departed between September to November
Route structure	C$_j$	$C_j = 1$ if flight chooses route j
Wind	WD	Wind distance (in 1000 nautical miles)

$$\begin{aligned} Y \quad &= \beta_0 + \beta_1 \cdot \mathrm{TS} + \beta_2 \cdot \mathrm{R} + \beta_3 \cdot \mathrm{Q} + \beta_4 \cdot \mathrm{Ice} + \beta_5 \cdot \mathrm{H} \\ &+ \beta_6 \cdot \mathrm{MIT} + \beta_7 \cdot \mathrm{AFP} + \beta_8 \cdot \mathrm{SAA} + \beta_9 \cdot \mathrm{BH} \\ &+ \beta_{10} \cdot \mathrm{Spr} + \beta_{11} \cdot \mathrm{Smr} + \beta_{12} \cdot \mathrm{Fall} \\ &+ \sum_{j=1}^{m-1} \beta_{c,j} \cdot \mathrm{C}_j \end{aligned} \tag{3.5}$$

3.4.3 Estimation Results

We have conducted comprehensive numerical studies on the eight airport pairs in Table 3.1 by experimenting with different nesting structures and variable selections. For the logit models, however, only airport pairs IAH ⟶ BOS and JFK ⟶ FLL exhibit significant nesting effects. We summarize the specific nesting structures in Table 3.6, in which three nests have been identified for alternatives in IAH ⟶ BOS, and four for JFK ⟶ FLL.

Also notice that for both pairs, those nests with only one alternative are “degenerate” nests whose nesting coefficients λ_k are fixed to 1 and therefore will not be estimated.

Table 3.6: Nesting Structures

	IAH $\longrightarrow$ BOS	JFK $\longrightarrow$ FLL
Nest 1	[Red, Cyan]	[Magenta, Blue]
Nest 2	[Magenta, Blue]	[Red]
Nest 3	[Green]	[Green]
Nest 4		[Cyan]

Note:Check Table 3.3 for the color code of alternatives

Table 3.8 and Table 3.9 report the complete estimation results for the inefficiency model. Notice that the subscripts of the variable names for route structure fixed effects are the color code for different clusters, for example, C_g is the fixed effect for the green cluster. Readers may refer to Table 3.3 for the complete color code. To save space, Table 3.7 only reports the estimated coefficients of those causal factors in the logit model; other coefficients, while most of which are significant, are not presented in this chapter. For all three tables, each column presents the estimates for one airport pair shown in the header, and the significance level is shown by the stars.

From Tables 3.8 and 3.9, the adjusted R^2's range from 0.543 to 0.858, except for pair SFO to MSP, which reports $R^2 = 0.255$. While we acknowledge that much of the variations in the SFO – MSP case might not be explained by the inefficiency model, we adopt the result as is and leave other possible modeling techniques in the future. Apart from the R^2's, we also observe that the vast majority of the coefficients in the inefficiency models are significant, and their signs generally match our expectations. First of all, the estimates confirm that thunderstorm activity along the chosen routes increases en route inefficiency for all eight airport pairs considered. However, among the four ground-based weather observations, only the rain exposure appears to have a minor positive impact on four pairs: JFK to FLL, SFO to MCO, EWR to LAS and SFO to MSP. Second, AFP delays only deem to have minor impact to flights from IAH to BOS in the tactical reroute, while MIT stringency and SAAs seem to affect the en route inefficiency more broadly across pairs. Third, the estimates for cluster membership fixed effects are all significant, and their order matches the statistics shown in Table 3.3, which confirms the effects of route selection. For instance, the estimate for CL_g in IAH to BOS case implies that if a flight decides to follow the path in cluster “g (green)”, then its inefficiency will be systematically 2.296% lower than flying the “red” cluster. Flights departing within the busy hours (8 AM to 8 PM local time) are all confirmed to be less efficient than departing at other hours. Finally, estimates for summer season fixed effects are somewhat surprising. While we expected their estimates to be consistently larger than other seasons, many of them are not statistically significant. This is probably because the effect of summertime convective weather has been explicitly accounted for in these models, and thus does not contribute to the summer season fixed effect.

As for the estimates presented in Table 3.7, the estimates in general agree with our expectations. Thunderstorm has significantly large negative impacts on route utility across all eight airport pairs, while rain reports significant negative estimates only for IAH to BOS, BOS to IAH, and EWR to LAS. This suggests that in the strategic planning phase, a flight will tend to fly a route that has less fewer convective activities. Second, the estimates for wind distance are significant with positive signs for all jet-stream-impacted airport pairs, which implies that flights tend to choose routes with favorable (tail) winds. However, for flights between FLL and JFK, possibly because of the comparably shorter distance and lower wind activities, wind is not an important factor on the route selection. Finally, MIT restrictions, AFP delays and SAAs only appear to have negative and significant estimates for a limited number of airport pairs, indicating that they are important in strategic route choice for some but not all OD pairs. We also report in Table 3.7 the prediction accuracy of the route choice models, which ranges from 55.99% (IAH to BOS) to 93.42% (SFO to MSP), and the pseudo $\bar{R}^2$, which ranges from 0.331 (IAH to BOS) to 0.722 (FLL to JFK). These statistics indicate relatively good fit of our specified logit models.

3.4.4 Contributions of Causal Factors

We further use the estimated logit and inefficiency models to quantify the contributions of wind, weather, TMIs and SAAs to flight en route inefficiency. Specifically, we are trying to see how the inefficiency changes, in percentage, if we "zero out" specific condition(s) (e.g., wind) in the airspace, when comparing to the actual airspace conditions observed in the data. The detailed procedure, known as counterfactual analysis, is described as follows.

1. **<u>STEP 1</u>**: Use the estimated logit models to calculate the probabilities of flight i choosing each route j in the alternative set, given the values of those causal factors $\mathbf{x}_{ij}$ in the original dataset. We denote this probability as $P^F_{ij} = \Pr(C_j|\mathbf{x}_{ij}), \forall j \in \{1, 2, \ldots m\}$.

2. **<u>STEP 2</u>**: For each route choice alternative j, substitute its associated route structure fixed effect in the inefficiency model with the calculated probability P^F_{ij} in STEP 1, and calculate the predicted en route inefficiency for flight i in the original dataset, denoted as $\hat{Y}^F_{ij}$.

3. **<u>STEP 3</u>**: Construct a counterfactual scenario in which we set the value(s) of any[3] investigated causal factor variable(s) (e.g., wind distance) to zero.

4. **<u>STEP 4</u>**: Estimate the route choice probability under the new scenario. We denote this probability as $P^{\tilde{F}}_{ij} = \Pr(C_j|\mathbf{x}_{ij} = [0, \ldots]), \forall j \in \{1, 2, \ldots m\}$.

[3] We can set whatever number of causal variables to 0, depending on whose variables contribution we want to investigate

Table 3.7: Estimation Results for the Logit Models

	Estimates (Est./Std.)							
Airport Pair	IAH BOS	BOS IAH	JFK FLL	FLL JFK	SFO MCO	LAX CLE	EWR LAS	SFO MSP
TS	-1.441*** (0.209)	-2.0136*** (0.437)	-1.405*** (0.248)	-0.802** (0.284)	-2.948*** (0.637)	-3.702*** (0.726)	-2.5163*** (0.445)	-5.593*** (1.760)
R	-0.0663*** (0.01)	-0.0618*** (0.016)	-	-	-	-	-0.1231*** (0.025)	-
WD	16.77*** (1.033)	21.616*** (1.438)	-	-	25.928*** (2.230)	21.789*** (2.084)	21.8581*** (1.322)	31.2304*** (3.086)
MIT	-	-	-	-	-0.246** (0.096)	-	-	-
AFP	-	-0.0145*** (0.003)	-0.0237*** (0.006)	-	-		-	-
SAA	-0.608** (0.282)	-	-	-	-	-0.0728* (0.039)	-	-
	Nesting Coefficient(s)							
μ_1	2.20*** (0.282)	-	2.708*** (0.556)	-	-	-	-	-
μ_2	2.63*** (0.442)	-	-	-	-	-	-	
Pseudo R^2	0.341	0.433	0.681	0.725	0.581	0.543	0.357	0.713
Pseudo $\bar{R}^2$	0.331	0.423	0.677	0.722	0.564	0.529	0.347	0.709
Accuracy	53.99%	63.79%	86.20%	88.34%	79.45%	76.76%	53.66%	93.42%

Note: $^{*}p<0.1$; $^{**}p<0.05$; $^{***}p<0.01$

Table 3.8: Estimation Results for the Inefficiency Model

	Estimates (Est./Std.)							
Airport Pair	IAH BOS	BOS IAH	JFK FLL	FLL JFK	SFO MCO	LAX CLE	EWR LAS	SFO MSP
BH	0.578***	0.09**	0.265***	0.795***	0.468***	0.264***	0.019	0.045*
	(0.106)	(0.038)	(0.044)	(0.06)	(0.151)	(0.056)	(0.04)	(0.026)
C_g	2.296***	4.795***	9.736***	5.711***	3.595***	-1.477***	-1.625***	0.809***
	(0.044)	(0.076)	(0.166)	(0.086)	(0.126)	(0.059)	(0.039)	(0.042)
C_m	-0.998***	0.637***	6.13***	0.894***	2.412***	-0.318**	-1.146***	
	(0.043)	(0.038)	(0.061)	(0.307)	(0.078)	(0.143)	(0.049)	
C_c	-2.247***	-0.035	10.259***	7.794***	1.115***	1.812***	1.554***	
	(0.174)	(0.076)	(0.125)	(0.426)	(0.098)	(0.234)	(0.069)	
C_b	-0.649***	3.838***	12.919***				0.366**	
	(0.162)	(0.128)	(0.218)				(0.144)	
Spr	0.028	0.105**	0.159***	0.123*	-0.088	0.158**	0.019	0.037
	(0.052)	(0.05)	(0.047)	(0.069)	(0.057)	(0.077)	(0.046)	(0.031)
Smr	0.152***	-0.048	0.019	0.1	-0.003	-0.059	0.025	0.106***
	(0.052)	(0.05)	(0.053)	(0.077)	(0.075)	(0.077)	(0.048)	(0.029)
Fall	0.016	0.027	-0.041	-0.039	-0.02	-0.287***	-0.031	0.05*
	(0.05)	(0.049)	(0.05)	(0.072)	(0.063)	(0.079)	(0.045)	(0.03)
Intercept	3.118***	1.512***	1.49***	1.649***	1.276***	2.29***	2.467***	0.595***
	(0.11)	(0.047)	(0.047)	(0.062)	(0.148)	(0.063)	(0.051)	(0.028)

Note:**p*$<$*0.1;* ***p*$<$*0.05;* ****p*$<$*0.01*

Table 3.9: Estimation Results for the Inefficiency Model (Continued)

	Estimates (Est./Std.)							
Airport Pair	IAH BOS	BOS IAH	JFK FLL	FLL JFK	SFO MCO	LAX CLE	EWR LAS	SFO MSP
TS	0.664***	0.831***	0.567***	1.036***	0.897***	1.598***	0.981***	0.83***
	(0.092)	(0.09)	(0.102)	(0.163)	(0.156)	(0.188)	(0.128)	(0.158)
R			0.007**		0.024***		0.009*	0.007*
			(0.003)		(0.008)		(0.005)	(0.004)
AFP	0.005***							
	(0.001)							
MIT	0.065***		0.114***	0.122***		0.196***		
	(0.016)		(0.017)	(0.016)		(0.061)		
SAA					0.011**			0.054***
					(0.004)			(0.003)
Observations	1580	1552	3943	3843	691	822	1504	2400
R^2	0.795	0.767	0.859	0.604	0.797	0.548	0.727	0.257
Adjusted R^2	0.794	0.766	0.858	0.604	0.794	0.543	0.725	0.255

Note: $^{*}p<0.1$; $^{**}p<0.05$; $^{***}p<0.01$

5. **STEP 5**: For each route choice alternative j, substitute its associated route structure fixed effect in the inefficiency model with the estimated probability $P_{ij}^{\tilde{F}}$ in STEP 4, and calculate the predicted en route inefficiency for flight i using the ***new data***, denoted as $\hat{Y}_{ij}^{\tilde{F}}$.

6. **STEP 6**: Calculate the percentage change in the the expected inefficiency using Equation 3.6. We term this percentage change as the ***contribution*** of (any) investigated causal factor(s).

$$\delta = \frac{\sum_i \left(\sum_{j=1}^{m} \hat{Y}_{ij}^{F} \cdot P_{ij}^{F} - \sum_{j=1}^{m} \hat{Y}_{ij}^{\tilde{F}} \cdot P_{ij}^{\tilde{F}} \right)}{\sum_i \sum_{j=1}^{m} \hat{Y}_{ij}^{F} \cdot P_{ij}^{F}} \tag{3.6}$$

The contributions for wind, convective weather (including thunderstorm and rain), TMIs and SAAs are summarized in Table 3.10, in which columns 3 to 7 show respectively the percentage change in the en route inefficiency in scenarios with and without the factor in the header row, and the second column shows the baseline average en route inefficiency (HIE) calculated based on the original dataset. Notice that a negative contribution statistics of a factor implies that the inefficiency increases in the case in which that factor is not present. From the table, the convective weather contributes up to 7% to the routing inefficiency, while wind contributes up to about 14%. TMIs do not deem to have significant contributions across all airport pairs, while SAAs contribute the most for the case SFO to MSP. The column "All" is the scenario where we set all factors to zero, and the numbers suggest that the contributions of different factors are approximately additive.

Although we are somewhat surprised by the "marginal" contributions presented in the table, especially for convective weather and wind due to their largely significant estimates in the statistical models, we find that for some airport pairs, their route-specific attributes are noticeably evenly distributed across alternatives. We show in Figure 3.18 a correlation matrix of three attributes – thunderstorm, rain and wind distance – across the five alternatives for the case BOS to IAH, in which each cell is the correlation between an attribute for a specific route and the others. For example, the cell at the first row and second column suggests that the thunderstorm exposure for the "red" and "green" routes has a Pearson correlation of 0.65. From this figure, we observe obvious positive correlations for each attribute between alternatives, especially for the wind variable. Therefore, to further understand what might lead to the "marginal" contribution estimates, we set up another counterfactual scenario in which only one alternative, say "red", has a "clear sky" while the others remain the same. Specifically, we follow the general procedure proposed in the beginning of this section, but modify step 3 such that we only set the value(s) of (any) investigated causal factor variable(s) of ***one*** specific alternative to 0, while keeping those variables unchanged for the other alternatives. Table 3.11 presents the full results. For example, in the case of IAH to BOS, if we clear the convective weather solely on the "magenta" alternative, then the overall inefficiency decreases by about 5%, while clearing the wind condition increases the inefficiency by about 8%. This is because when there is no wind on "magenta", flights will shift their route choice probabilities from "magenta" to other more inefficient routes, say "red" and "green". More dramatic changes are also observed for pairs such as BOS to IAH, in which inefficiency decreases by about 80% by clearing the wind condition of the "green" alternative. This is caused by the fact that flights from BOS to IAH are mostly in the headwind condition (negative wind distance), and therefore, clearing the wind condition on "green", which is the most inefficient route, attracts more flights to choose this alternative.

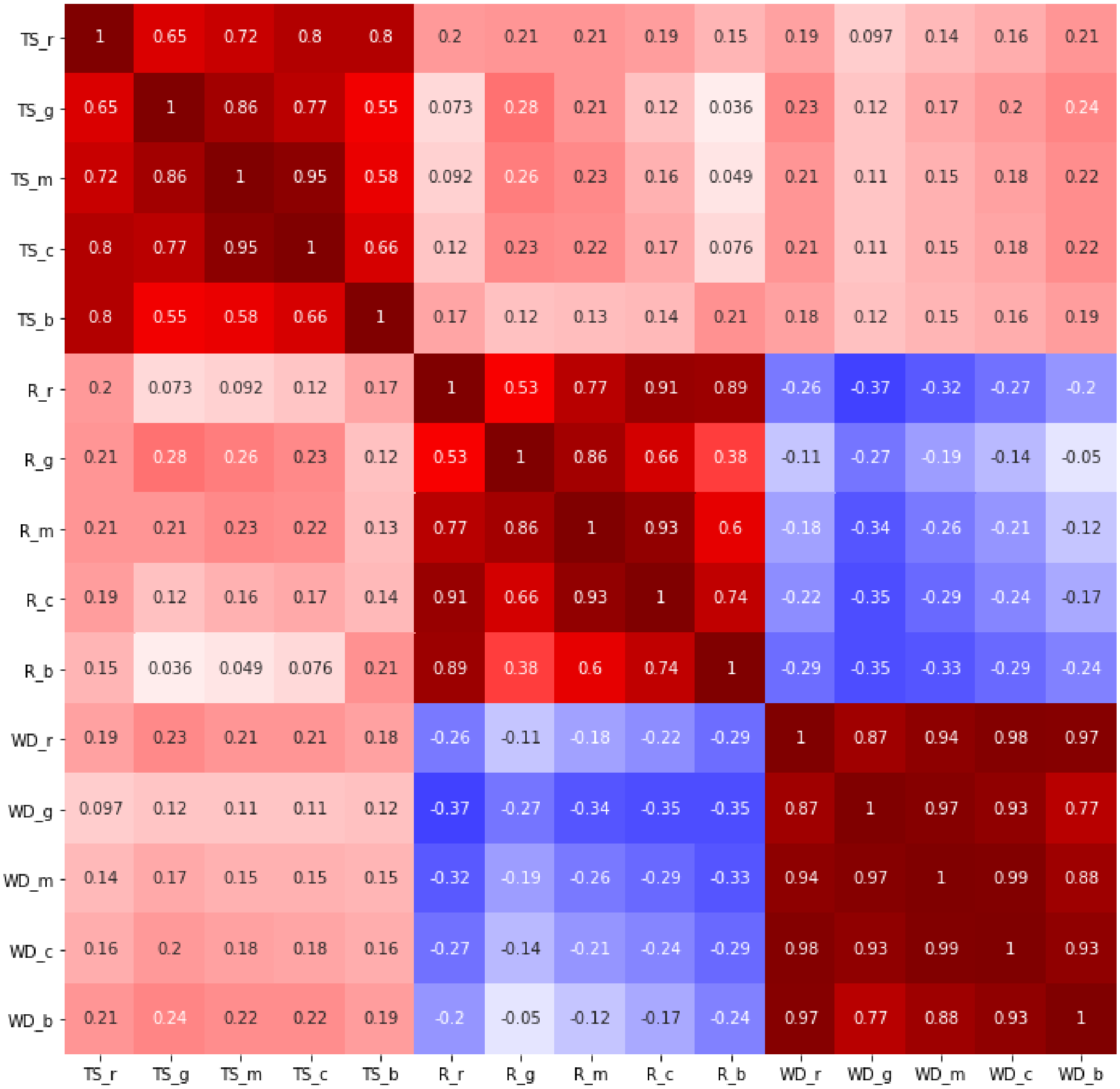
Table 3.10: Contribution of Causal Factors

	Baseline HIE	Convection	Wind	TMIs	SAAs	All
IAH to BOS	4.366%	0.799%	2.146%	0.962%	-4.893%	-1.226%
BOS to IAH	2.580%	3.844%	0.373%	-	-	3.973%
FLL to JFK	3.585%	1.393%	-	4.441%	-	5.786%
JFK to FLL	3.052%	1.991%	-	2.394%	-	4.222%
SFO to MCO	2.557%	6.476%	12.740%	0.461%	2.291%	22.058%
LAX to CLE	1.636%	7.104%	13.987%	2.426%	1.262%	24.283%
EWR to LAS	2.134%	5.230%	11.788%	-	-	17.416%
SFO to MSP	1.109%	2.989%	2.441%	-	21.935%	27.387%

3.5 Conclusions

In this research, we have demonstrated a practical means for fusing FAA trajectory performance databases with convective weather, wind, miles-in-trail (MIT), airspace flow programs (AFP), and special activity airspace (SAA) sources for the purpose of obtaining an improved estimate of flight trajectory efficiency, which presents a first step in ascribing en-route trajectory inefficiencies to causal factors. Two mechanisms have been proposed to relate those factors to inefficiency – strategic route choice and tactical reroute, through which we present a methodological framework, containing trajectory clustering, attribute matching, logit model and inefficiency model, to specifically tackle the problem.

In our framework, the inefficiency of a trajectory is defined as the horizontal distance in excess of the great circle distance, and is modeled as resulting from an assignment process through which the trajectory "chooses" among several trajectory clusters based on attributes of the nominal routes for each cluster, and a tactical process in which flight inefficiency for trajectories in a given cluster is related to convective weather, TMIs and etc. Using this approach, the contributions of wind, connective weather, MIT, AFP and SAA to flight inefficiency are estimated. Results vary across OD pairs, but in general if we systematically clear the values of these causal factors in the whole airspace, convective weather and wind make the greatest, yet not dramatic, contribution. However, we have also shown that this might be caused by the largely homogeneous distribution of different attributes across alternatives,



	TS_r	TS_g	TS_m	TS_c	TS_b	R_r	R_g	R_m	R_c	R_b	WD_r	WD_g	WD_m	WD_c	WD_b
TS_r	1	0.65	0.72	0.8	0.8	0.2	0.21	0.21	0.19	0.15	0.19	0.097	0.14	0.16	0.21
TS_g	0.65	1	0.86	0.77	0.55	0.073	0.28	0.21	0.12	0.036	0.23	0.12	0.17	0.2	0.24
TS_m	0.72	0.86	1	0.95	0.58	0.092	0.26	0.23	0.16	0.049	0.21	0.11	0.15	0.18	0.22
TS_c	0.8	0.77	0.95	1	0.66	0.12	0.23	0.22	0.17	0.076	0.21	0.11	0.15	0.18	0.22
TS_b	0.8	0.55	0.58	0.66	1	0.17	0.12	0.13	0.14	0.21	0.18	0.12	0.15	0.16	0.19
R_r	0.2	0.073	0.092	0.12	0.17	1	0.53	0.77	0.91	0.89	-0.26	-0.37	-0.32	-0.27	-0.2
R_g	0.21	0.28	0.26	0.23	0.12	0.53	1	0.86	0.66	0.38	-0.11	-0.27	-0.19	-0.14	-0.05
R_m	0.21	0.21	0.23	0.22	0.13	0.77	0.86	1	0.93	0.6	-0.18	-0.34	-0.26	-0.21	-0.12
R_c	0.19	0.12	0.16	0.17	0.14	0.91	0.66	0.93	1	0.74	-0.22	-0.35	-0.29	-0.24	-0.17
R_b	0.15	0.036	0.049	0.076	0.21	0.89	0.38	0.6	0.74	1	-0.29	-0.35	-0.33	-0.29	-0.24
WD_r	0.19	0.23	0.21	0.21	0.18	-0.26	-0.11	-0.18	-0.22	-0.29	1	0.87	0.94	0.98	0.97
WD_g	0.097	0.12	0.11	0.11	0.12	-0.37	-0.27	-0.34	-0.35	-0.35	0.87	1	0.97	0.93	0.77
WD_m	0.14	0.17	0.15	0.15	0.15	-0.32	-0.19	-0.26	-0.29	-0.33	0.94	0.97	1	0.99	0.88
WD_c	0.16	0.2	0.18	0.18	0.16	-0.27	-0.14	-0.21	-0.24	-0.29	0.98	0.93	0.99	1	0.93
WD_b	0.21	0.24	0.22	0.22	0.19	-0.2	-0.05	-0.12	-0.17	-0.24	0.97	0.77	0.88	0.93	1

Figure 3.18: Pearson Correlation of Attributes Across Alternatives

and further, if we "clear the sky" solely for an alternative route, flight en route inefficiency may receive substantial improvement/deterioration. This, on the other hand, should be specifically emphasized for future advise of TMI implementations in the NAS system.

In addition to its scientific contribution, this work also has a number of practical applications. First, it helps target interventions to reduce flight inefficiency, by identifying which causes of inefficiency are the most important for different OD pairs. Equally important, it estimates the extent to which flight inefficiency, as defined in this chapter, is an illusion arising from the choice of routes with favorable winds but longer ground distances. Finally, it reveals the degree to which inefficiency is not the result of any identifiable cause. This inefficiency "dark matter" is presumably attributable to features of the NAS route structure, which may also be targets for intervention.

Table 3.11: Percentage Change In HIE by Zeroing Out Factor(s) of One Alternative

		Convection	Wind	TMIs	SAAs	All
IAH to BOS	Red	1.45%	1.58%	0.17%	0.00%	2.08%
	Green	-5.00%	12.34%	0.19%	-6.14%	6.70%
	Magenta	4.65%	-7.63%	0.58%	1.20%	-4.31%
	Cyan	1.97%	-0.40%	0.01%	0.00%	0.06%
	Blue	0.48%	-0.05%	0.04%	0.03%	0.20%
BOS to IAH	Red	6.69%	15.42%	-	-	19.87%
	Green	-4.04%	-80.55%	-	-	-90.48%
	Magenta	1.25%	-4.63%	-	-	-2.44%
	Cyan	1.32%	11.67%	-	-	14.52%
	Blue	-1.86%	-47.13%	-	-	-55.21%
JFK to FLL	Red	5.84%	-	3.01%	-	8.56%
	Green	-0.44%	-	0.01%	-	-0.43%
	Magenta	-2.31%	-	-0.64%	-	-3.14%
	Cyan	-1.91%	-	-0.39%	-	-2.65%
	Blue	-2.34%	-	-0.71%	-	-3.76%
FLL to JFK	Red	2.66%	-	4.42%	-	7.01%
	Green	-1.34%	-	-0.81%	-	-2.10%
	Magenta	0.07%	-	0.08%	-	0.19%
	Cyan	-0.04%	-	0.02%	-	-0.04%
SFO to MCO	Red	7.54%	-50.61%	0.90%	0.76%	-45.76%
	Green	-0.70%	3.50%	-0.14%	0.06%	3.36%
	Magenta	-0.69%	9.95%	-0.50%	1.02%	9.73%
	Cyan	0.17%	0.97%	-0.09%	0.19%	1.11%
LAX to CLE	Red	-0.40%	22.19%	0.30%	-1.19%	21.96%
	Green	8.84%	-40.32%	2.17%	2.85%	-28.94%
	Magenta	0.46%	1.06%	0.15%	-0.41%	1.14%
	Cyan	-0.89%	2.21%	0.18%	0.11%	2.13%
EWR to LAS	Red	-3.94%	-40.00%	-	-	-39.25%
	Green	12.75%	37.91%	-	-	44.68%
	Magenta	4.46%	11.30%	-	-	16.95%
	Cyan	-5.98%	-95.49%	-	-	-101.83%
	Blue	-0.52%	-34.57%	-	-	-37.23%
SFO to MSP	Red	3.25%	-29.47%	-	20.35%	-14.09%
	Green	-0.39%	4.53%	-	1.60%	4.72%

Chapter 4

Predicting Aircraft Trajectories: A Deep Generative Convolutional Recurrent Neural Networks Approach

Nomenclature

Scalars

x Longitude

y Latitude

z Altitude

t Time

$\dot{x}$ First order derivative of longitude

$\dot{y}$ First order derivative of latitude

Vectors

$P_t = [x, y, z, t]$ 4-dimensional track points defined by longitude, latitude, altitude, and time

$X_t = [x, y, z, \dot{x}, \dot{y}]$ 5-dimensional aircraft state at time t defined by longitude, latitude, altitude, longitude speed, and latitude speed

$\tilde{X}_t = [x, y]$ 2-dimensional aircraft state at time t defined by longitude and altitude

F_t^G Feature cube grid at time t

F_t Feature cube spanned on the feature cube grid at time t

Matrices

$\mathbf{P} = [P_1, P_2, \ldots, P_T]$ Sequence of 4-dimensional track points

$\mathbf{X} = [X_1, X_2, \ldots, X_T]$ Sequence of 5-dimensional aircraft states

$\tilde{\mathbf{X}} = [\tilde{X}_1, \tilde{X}_2, \ldots, \tilde{X}_T]$ Sequence of 2-dimensional aircraft states

$\mathbf{F}^G = [F_1^G, F_2^G, \ldots, F_T^G]$ Sequence of feature cube grids

$\mathbf{F} = [F_1, F_2, \ldots, F_T]$ Sequence of feature cubes

4.1 Introduction

With the growing 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. Based on this system, we have developed route choice models in Chapter 3 that enable us to analyze air traffic flow and predict to which nominal route a flight would fly, given convective weather, wind, and traffic management initiatives. In Europe, EUROCONTROL adopted the PREDICT system ([22]) to forecast the pre-tactical traffic load. These systems depend on aircraft trajectory prediction tools. However, due to the computational complexity, most of today's prediction systems are based on deterministic flight trajectory prediction processes, using, for example, filed flight plans or historical routes ([54], [28]). 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). 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. Furthermore, in the next generation of the NAS system (NextGen), the concept of trajectory based operations (TBO) has been proposed to be a cornerstone ([26]). TBO requires the development of tools to plan and predict accurate aircraft trajectories in four dimensions – lateral, vertical and time. In recent years, efforts have been spent on developing systems focusing on optimizing and controlling aircraft 4D trajectories ([5], [59], [60]), however, there are far fewer studies on predicting 4D trajectories. Therefore, in this chapter, we propose a methodological framework to predict, in a real-time manner, actual 4D trajectories, which incorporates different sources of uncertainties including weather, wind, and management actions.

In our framework, the future trajectory of an aircraft is modeled as a sequence of 4D coordinates that are correlated with its realized trajectory, last filed flight plan, which is a sequence of 2D waypoints, and weather conditions in the vicinity. Therefore, we can formulate our task as a "sequence to sequence learning" problem, in which the input sequence is the flight plan and the output is the actual flight trajectory. To solve this sequential learning problem, an encoder-decoder recurrent neural network structure has been employed, where the encoder learns from the flight plan and the decoder integrates the weather information and recursively ”translates” the embedded flight plan information into a full 4D trajectory. We further assume the 4D trajectory coordinates follow conditional Gaussian mixture, whose parameters are learned by the decoder. To model the weather effect, we first propose an efficient tree-based matching algorithm that can work both in batch mode and recursive mode to 4D match aircraft coordinates with nearby convective weather, wind speeds and air temperature. Then we integrate convolutional layers into the decoder network pipeline to extract representations from the high-dimension weather features. In summary, this chapter has the following contributions.

- To the authors' knowledge, this chapter is the first research using an encoder-decoder recurrent neural network structure to predict 4D aircraft trajectories. Once trained, our model can also be easily adapted into systems that require real-time predictions with accurate prediction intervals.

- Our proposed approach is a generative model that incorporates multiple factors that influence trajectories, including convection, wind, and temperature. These features are constructed by an efficient tree-based matching algorithm, which is highly generalizable and can be used in many spatiotemporal matching applications.

- We propose both training and inference pipelines, in which the inference pipeline employs multiple filtering and searching algorithms to produce the best predictions.

The rest of the chapter is organized as follows. In section 4.2, we review related works in trajectory prediction and deep neural networks. Section 4.3 introduces related concepts and summarizes our data sources and preprocessing procedure. Section 4.4 presents our matching (feature engineering) algorithms, and section 4.5 describes our methodological framework and model architecture. In section 4.6, we present the results of a case study. Section 4.7 offers conclusions and suggestions for future research.

4.2 Related Work

Common approaches to aircraft trajectory prediction can be summarized into two categories: deterministic and probabilistic. The former approach ([5], [2], [8]) usually applies a specific aerodynamic model to estimate the state of an aircraft and then propagate the estimated states into the future (e.g., using Kalman filter). Although it accounts for specific aircraft parameters and kinematic equations, this approach, without considering any uncertainties such as future winds and pilot control actions, can either predict only a specific phase of a flight or suffers from degraded prediction accuracy. Moreover, this approach yields a "point estimate" of the future trajectory rather than a prediction internal. In contrast, the probabilistic approach ([14], [10], [3]) understates the aerodynamics but relies on statistical models to learn how aircraft fly from one point to another from historical trajectory datasets. De Leege et al. ([14]) trained a generalized linear model (GLM) to use wind and aircraft initial state to predict aircraft trajectories in the arrival airport's terminal area. However, instead of predicting a 4D trajectory, the authors only predict the time of arrival to a fixed set of reference points (a.k.a., waypoints). Choi and Hebert ([10]) proposed a Markov model to predict future motion of a moving object based on its past movement. In their work, the trajectory of an object is broken down into a sequence of short segments, which are assumed to be generated from latent states. Therefore, their method predicts future trajectory segments of an object instead of actual coordinates. Ayhan and Samet ([3]) extended the work of Choi and Hebert and applied the method to predict actual 4D coordinates (track points) of an aircraft trajectory. In the chapter, the 4D coordinate of the aircraft is defined as the hidden state, and the observed weather information that is closest to the aircraft coordinate is a realization of the hidden state. By training a hidden Markov model (HMM) on a historical trajectory and weather dataset, the authors obtained the parameters of the HMM model, which were further used to predict trajectories given the observed weather sequence. However, their model has three main drawbacks. First of all, the hidden state is fixed, indicating that every predicted track point can only be one of the historical track points. Second, for each track point, the authors only consider the weather condition that is closest to it, however, in aircraft routing problem, pilots usually consider a much larger region. Lastly, the prediction is highly dependent on the quality of the clustering results of the observed weather sequence, which is in practice difficult to control.

To a large extent, aircraft trajectories have clear temporal and spatial patterns, demonstrating high predictability. Thus, with the right tools, these patterns can be employed in predictive models. Recurrent neural networks (RNNs) have become the state of the art for sequence modeling. Furthermore, long short-term memory (LSTM) ([39]) is one of the most popular variations of RNNs with proven ability and stability in solving tasks in various domains, such as speech recognition ([36], [11], [12]), neural translation ([69]), and image captioning ([42], [75], [15]). The most relevant work is that of work of Lin et al. ([47]) and Alahi et al. ([1]). Lin et al. ([47]) proposed a deep generative model, which integrated an input-output HMM (IO-HMM) and a LSTM network, to model urban mobility. In the model, the daily activities of travelers are sequences of locations they visit through a day and are generated by the IO-HMM from cellular data. Then spatiotemporal patterns of the activities are learned by a LSTM network. Alahi et al. ([1]) proposed a "Social LSTM" model to predict human trajectories in crowded spaces. In the space, multiple individuals are observed, and each individual's trajectory is modeled by a LSTM network. To share information among individuals, the authors use the pooling techniques to connect those LSTM networks. Comparing to simple models such as Gaussian process and linear model, the authors approach has far better prediction errors on two open source datasets.

Inspired by Sutskever et al. ([69]), Lin et al. ([47]) and Alahi et al. ([1]), we develop an encoder-decoder LSTM-based generative model to predict aircraft 4D trajectories. Our approach differs from the past efforts in the following aspects.

- It is a probabilistic generative model that both learns the spatiotemporal trajectory patterns and can generate a full 4D trajectory from the realized trajectory with any length of information.
- It provides prediction intervals.
- it gives the possibility of predicting different routes that are not observed in the training set.
- It uses high-dimension regional features generated from weather variables.

4.3 Preliminaries

In this Section, we first introduce some key concepts and notations in our methodological framework. Then we formally formulate our problems. Lastly we introduce the datasets and our detailed preprocessing steps.

4.3.1 Definitions and Notations

Definition 4.3.1. 4D Trajectory $\mathbf{P}$ A sequence of 4-dimensional track points defined by longitude x, latitude y, altitude z, and time t.

Definition 4.3.2. Pre-departure Last Filed Flight Plan $\tilde{\mathbf{X}}$ A sequence of 2-dimensional way points described by longitude x and latitude y.

Definition 4.3.3. Aircraft State $\mathbf{X}$ A sequence of a tuple of aircraft 3-dimensional positions (x, y, z) and lateral speeds (longitude speed $\dot{x}$ and latitude speed $\dot{y}$)

Definition 4.3.4. Georeferencing System $\mathbf{G}$ A grid of the fixed spatial points in the 3-dimensional space.

Definition 4.3.5. 4D matching algorithm An algorithm to match a 4-dimensional trajectory point P_t with raw features of interest by some pre-defined distance measures.

Definition 4.3.6. Feature Cube F_t A multi-dimensional array of matched features at time t around a 4-dimensional track point P_t.

4.3.2 Problem Formulation

In this work, we seek to learn a *generative model* that predicts the actual 4D aircraft trajectories. For each flight, we observe its last filed flight plan prior to departure $\tilde{\mathbf{X}}$, which is a sequence of 2-dimensional waypoints that guides the actual flight path but is not always strictly followed. Given the aircraft state X_t at time t, we observe the weather conditions such as wind speed, air temperature and convective weather in the vicinity of the aircraft, and 4D match it with the flight location to create the feature cube F_t. Then we use the F_t to predict its state for the next time instance $t+1$. Thus, given flight plan, weather datasets, and some initial states of the aircraft, we can predict a whole sequence of the flight trajectory to the last timestamp by recursively predicting the next state and 4D matching the state with weather information. Mathematically, the goal of this research is to find some mapping $\mathcal{F}$, such that:

1. During training process, $\big(\mathbf{X}, \tilde{\mathbf{X}}, \mathbf{F}\big) \xrightarrow{\mathcal{F}} \mathbf{X}$ learns the best spatiotemporal patterns of the 4D trajectory.

2. During inference process, $\big(\mathbf{X}[:t], \tilde{\mathbf{X}}, \mathbf{F}[:t]\big) \xrightarrow{\mathcal{F}} \mathbf{X}[(t+1):]$ can generate the rest of unobserved trajectory.

This task can be viewed as a combination of a translation and a sequence generation problem. Therefore, as a state-of-the-art algorithm to tackle similar problem, the encoder-decoder neural network structure can be employed ([69]) to approximate $\mathcal{F}$, where an encoder neural network takes an input sequence that corresponds to the last filed flight plan, and a decoder neural network recurrently predicts an output sequence corresponding to an actual flight trajectory.

4.3.3 Data Sources and Preprocessing

In this work, we used four datasets from different sources. The *flight tracks dataset*, which comes from FAA Traffic Flow Management System (TFMS), contains 4D positions of each aircraft throughout its flight – latitude, longitude, altitude, and time, whose resolutions are respectively about 1 minute, 1 minute, 100 feet, and 1 minute. For this study, we limit our scope for flights from George Bush International Airport (IAH) to Logan International Airport (BOS) in 2013, since over 95% of flights are operated by the United Airlines. During the preprocessing, we first excluded flights where spatial or temporal discontinuities were detected, and ones that started or ended outside of the selected terminal areas (0.5-degree latitude/longitude boxes). We then down sampled flight by eliminating one out of every two track points to reduce computational complexity. Lastly, we derived the course, and latitude and longitude speeds by assuming each flight has a constant ground velocity between two consecutive points. The final dataset includes 1,679 flights with an average sequence length of 94.

The *flight plan dataset*, which also comes from FAA TFMS, contains the last filed 2D flight plan coordinates (latitude and longitude) for each flight. While the lengths of flight plans vary from 9 waypoints to 144 waypoints, we can reduce their dimensions significantly by identifying characteristic points. We implemented a variant of the approximate trajectory partitioning algorithm proposed by [46], in which we introduced a parameter $\alpha \in [1, 2]$ to control the length of the output sequence. The final dataset includes 118 unique flight plans with maximal length of 16.

The *atmospheric datasets*, which contain the wind speed dataset and air temperature dataset, come from the North American Mesoscale (NAM) Forecast system. It produces high resolution atmospheric information four times a day – at respectively 00:00, 06:00, 12:00, and 18:00 UTC – and each production cycle provides forecast at hours 0, 1, 2, 3 and the last hour (6) of the cycle. The datasets use Lambert conformal conic (LCC) projection and the original horizontal resolution is 614 by 428, in which the longitude ranges from 152.88°W to 49.42°W, and latitude ranges from 12.19°N to 57.33°N. However, since our flight tracks only lie in the continental US, we cropped the original georeferencing system so that the new georeferencing is a box area with four corners: (130°W, 22°N), (130°W, 52°N), (64°W, 22°N), (64°W, 52°N). The new horizontal resolution is about 413 by 336, which is roughly half of the original. Apart from the horizontal projection, the datasets also include 39 isobaric pressure altitude layers ranging from 50 millibar (about 68,000 ft) to 1,000 millibar (about mean sea level), and we denote L^{atm}_{alt} the list of all 39 unique pressure

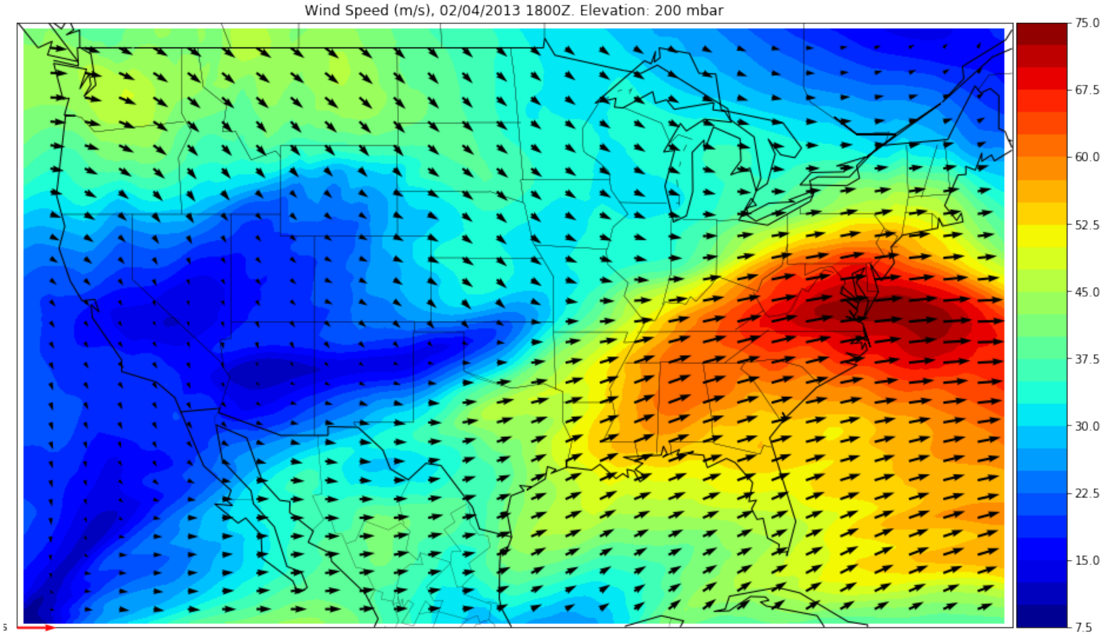


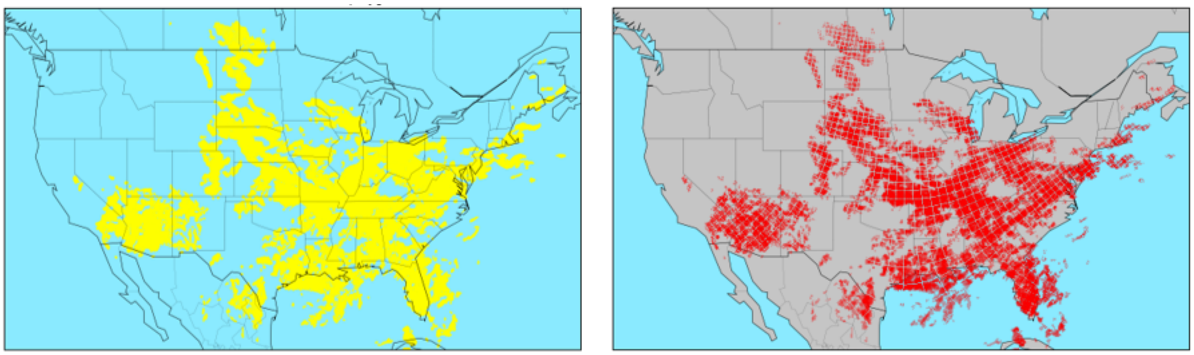
Figure 4.1: Wind Speed Map at 200 mbar Derived From the Atmospheric Datasets

altitudes. Therefore, at each forecast time instance, the final datasets have three separate data arrays – westerly wind speed, southerly wind speed, and air temperature – each of which has dimension 39 × 413 × 336. Figure 4.1 shows a wind speed map at 200 mbar at 18:00 Zulu on 02/04/2013, in which each arrow is the wind direction computed based on the westerly and southerly wind speeds, and the color indicates the speed of the wind. Readers can imagine that at each forecast dataset, we have 39 layers of wind speed map, and each corresponds to one dedicated altitude level. Lastly, we also want to point out that the NAM datasets are very similar to the NCAR dataset introduced in Chapter 3, Section 3.2, however NAM has far better resolution and more information that are required for our individual flight track prediction task.

The *convective weather dataset* comes from the National Convective Weather Forecast (NCWF) system. In the dataset, every record contains the locations of convective weather polygons (coordinates of boundaries and highest altitude of the storm) and the direction of movement at the time of recording. The dataset was typically updated every 5 minutes. We preprocessed the dataset by the following steps.

1. Unique storms' altitudes with resolution 1,000 ft, which yield to a list (in 1,000 ft) $L_{alt}^{wx} = [0, 14, 20, 24, 29, 35, 39, 45, 50, 54, 60, 65, 69]$.

2. Merge storm polygons within the same altitude group and hourly group (i.e., overlay all storm polygons at the same altitude level within an hour).



(a) One Day from the Original NCWF data

(b) Processed NCWF data

Figure 4.2: Comparison of NCWF datasets

3. At each time instance (hour of day), create a binary array with dimension 13×138934, where 13 is the number of altitude levels and 138,934 is the resolution of the horizontal georeferencing system $\mathbf{G}$ obtained from processing NAM datasets.

4. At each time instance, overlay storm polygons at the same altitude level to the georeferencing system $\mathbf{G}$. If a grid point from $\mathbf{G}$ is covered by a storm polygon, then the corresponding element in the data array has value 1, otherwise 0.

A comparison of the raw and processed NCWF datasets are illustrated by an example in Figure 4.2. The left subfigure shows one-day of convective weather activities projected on the 2D space, in which the yellow polygons describe the locations of convective regions. The right subfigure, however, shows the discretized convective information from the left, in which each red point indicates a single location where convection was reported.

Lastly, we obtain our horizontal georeferencing system $\mathbf{G}$ from the *atmospheric datasets*. Namely, we cropped the horizontal region to a rectangular space cornered by (130°W, 22°N), (130°W, 52°N), and (64°W, 22°N), and used all the grids from the reduced *atmospheric datasets* inside the region. The final $\mathbf{G}$ is shown by the blue raster in Figure 4.3, in which the red raster is the original georeferencing system.

4.4 Feature Engineering

In this work, we predict aircraft trajectories using three types of information. The first is the flight's last filed flight plan prior to departure. While this flight plan is clearly an important indicator of what the flight trajectory will be, it is by no means determinative. It is common for aircraft to deviate from their flight plans rather than "fly as filed". Convective weather, winds, clearances to fly direct, and vectors to resolve conflicts can all cause deviations. Figure 4.4, in which all flights shown by the blue trajectories filed the same flight plan shown by

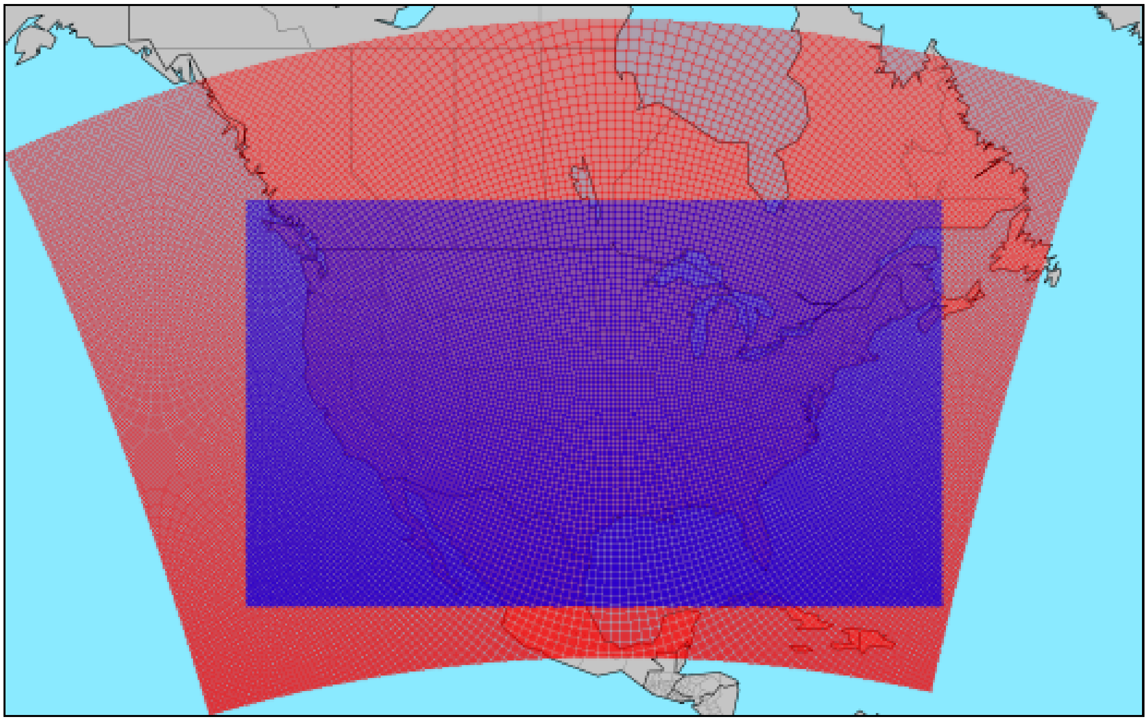


Figure 4.3: Georeferencing System **G**

the red curve reveals the extent to which flights deviate from their last pre-departure flight plans.

The second source of information used in our model pertains to convective weather. Strong updrafts and downdrafts are evident within a convection area, which cause significant and unfavorable turbulence. Therefore, aircraft almost always avoid those areas either strategically by choosing a route predicted to be clear of convective weather or tactically by maneuvering around convective weather cells. The last category of information pertains to atmospheric conditions, specifically air temperature and wind speed. The conditions can affect trajectories in at least two ways. First, flights prefer routes with strong tailwind since it saves both fuel and time. Third, high temperatures are associated with more turbulence, which pilots may seek to avoid. Finally, pilots prefer flying through cold and dense air since aircraft engines can produce more power.

To predict actual flight trajectories, it is crucial for us to convert the underlying datasets into features that provide the basis for the subsequent modelling. In this section, we first introduce the concept of *feature cube*, which is a multi-dimensional data array that contains the weather-related information surrounding a track point. Then we summarize a batch mode and a recursive mode approach to efficiently match the flight track with data cubes during the training and inference process, respectively.

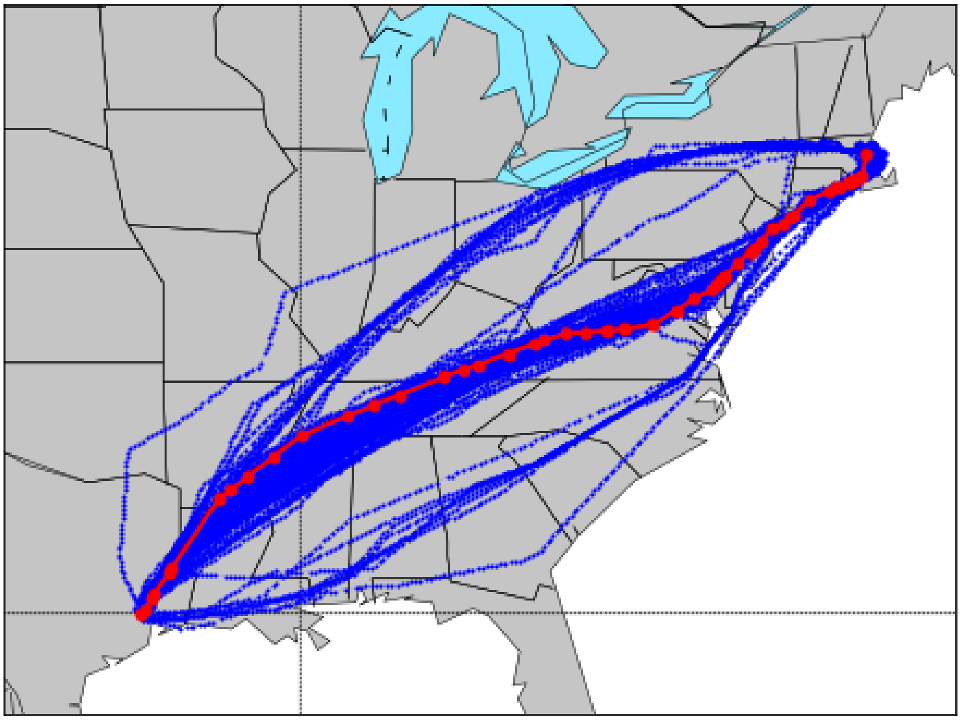


Figure 4.4: Actual Flight Trajectories vs. Their Flight Plan

4.4.1 Feature Cube Referencing System

To match flight trajectories with the relevant raw datasets, we generate a 4D referencing system that is related to a given track point. For each track point at time t, we first construct a grid square F_t^G surrounding the point. The width and height of the grid square are respectively dx and dy degrees in latitude/longitude, and the resolution of the square is n_x by n_y points. Due to the fact that only the weather conditions in front of an aircraft matters, we center *one side*, instead of the centroid, of the grid square at the track point. Then we rotate the grid square by the course of the previous track point. Figure 4.5 demonstrates an illustrative feature cube grid for one track point, in which the black dot and the arrow are respectively the location and course of the aircraft, and the grid square ($n_x = 20, n_y = 20, dx = 2°, dy = 2°$) is spanned by its grid point. Second, we specify the altitude of the grid square. For the atmospheric datasets (wind speeds and air temperature), we use the closest pressure altitude (e.g., 200 millibar) to the track point's altitude; for the convective weather dataset, we use the closest altitude in L_{alt}^{wx} to the track point's altitude. Lastly, we set the timestamp for the grid square by directly using the corresponding track point's timestamp.

The details of the above process are described in Algorithm 5[1]. Notice that each flight is

[1] We can run this algorithm in a batch mode by entering a collection of 4D flight trajectories $\mathbf{P}$; we can

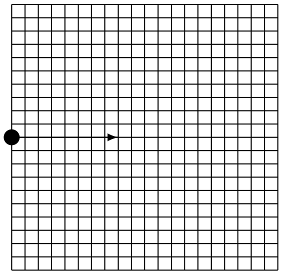


Figure 4.5: Illustration of Feature Cube Creations

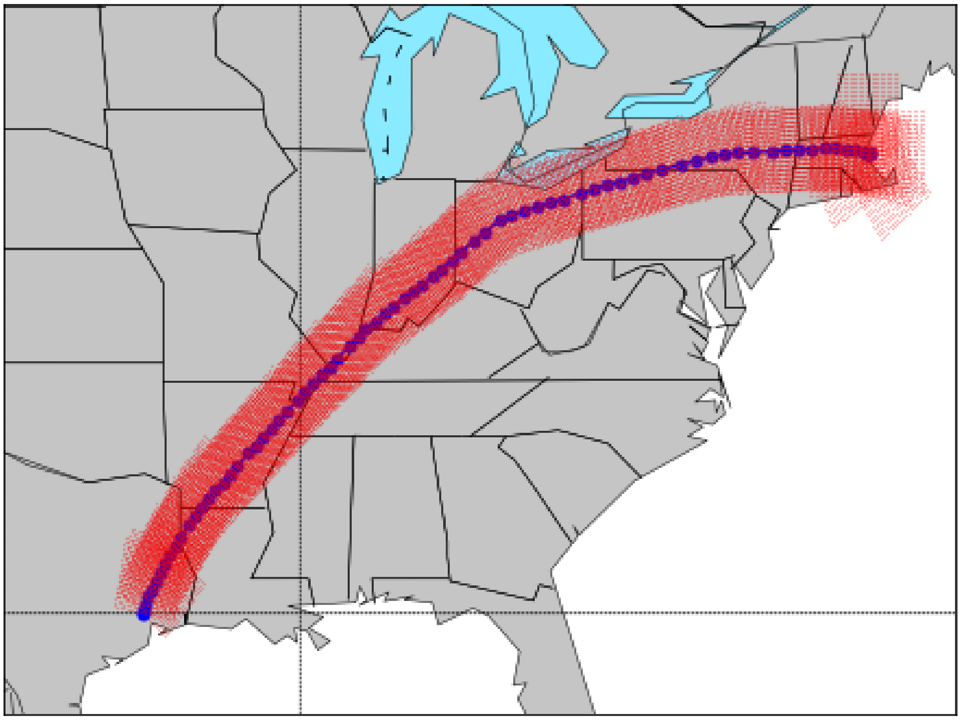


Figure 4.6: Feature Cube Grid Path Projected on the 2D Space

a sequence of track points, and therefore, has a trajectory of the referencing grids, which we term as feature cube grid path. An example of the feature cube grid path is illustrated in Figure 4.6, where the blue series is one of the actual trajectories and the red lattice shows the grid path around the trajectory.

also run this algorithm in a recursive mode by specifying **P** to be one single track point P_t

Algorithm 5 Feature Cube Grid Generator (FCGG)

```
Inputs
    $\mathbf{P} = [P_1, \ldots, P_T], P_t = [x, y, z, t]$: 4D flight trajectory;
    $L_{alt}^{atm}$: Atmospheric datasets unique altitude list;
    $L_{alt}^{wx}$: Convective weather dataset unique altitude list.
Parameters
    $n_x, n_y, dx, dy$: Size of the feature cube, default at respectively 20, 20, 2°, 2°.
Outputs
    $\mathbf{F}^G = [F_1^G, \ldots, F_T^G]$: Sequence of (2D) feature cube grid for $\mathbf{P}$
    $L_{atm}^G = [z_1^{atm}, \ldots, z_t^{atm}]$: Sequence of feature cube atmospheric altitude for $\mathbf{P}$
    $L_{wx}^G = [z_1^{wx}, \ldots, z_t^{wx}]$: Sequence of feature cube convection altitude for $\mathbf{P}$
procedure FCGG
    Initialize:
    Feature cube grid $F_t^G = \{g_{ij}\}, \forall t \in \{1, 2, \ldots, T\}, \forall i \in \{1, 2, \ldots, n_x \cdot n_y\}, \forall j \in \{1, 2\}$,
    $g_{p \cdot n_x + q, 0} = q \cdot dx/(n_x - 1)$,
    $g_{p \cdot n_x + q, 1} = p \cdot dy/(n_y - 1)$, $\forall 0 \leq q \leq n_x - 1, \forall 0 \leq p \leq n_y - 1$

    Initialize: empty $L_{atm}^G, L_{wx}^G$
    for $t$ in $\{1, 2, \ldots, T\}$ do
        1. Create rotation matrix $R_t = \begin{bmatrix} \cos(\theta_t) & -\sin(\theta_t) \\ \sin(\theta_t) & \cos(\theta_t) \end{bmatrix}$.
        2. Calculate rotated feature grid $F_t^G = F_t^G \cdot R_t + P_t'$, where $P_t' = [x, y]_t$.
        3. Calculate $z_t^{atm} = \arg\min_\xi \left\{\xi - z_t : \xi \in L_{alt}^{atm}\right\}$. Append $z_t^{atm}$ to $L_{atm}^G$.
        4. Calculate $z_t^{wx} = \arg\min_\xi \left\{\xi - z_t : \xi \in L_{alt}^{wx}\right\}$. Append $z_t^{wx}$ to $L_{wx}^G$.
    end for
end procedure
```

4.4.2 Feature Cube Matching

To construct highly descriptive feature space, we match the proposed feature cube grid for every track point with the weather-related datasets mentioned above. Both the grids and datasets have 4-dimensional spatiotemporal structures – latitude, longitude, altitude, and time, each grid therefore needs to query into the weather datasets and use the value of the closest 4D reference point as its feature value, which is generally computational extensive. In this study, we proposed a tree-based matching algorithm that efficiently matches flight trajectories with high-dimension datasets in a 4D manner. For conciseness, we only describe an example of matching with convective weather dataset, and the full algorithm is presented in Algorithm 6.

To accomplish our goal, two sets of k-d trees ([6]) – spatial tree and temporal tree – are firstly constructed (Algorithm 6: lines 3 – 6). To be more specific, the spatial tree TR_S is based on the horizontal georeferencing system $\mathbf{G}$, where each data entry is a tuple of grid's latitude and longitude. Since both the atmospheric datasets and convective weather dataset share the same $\mathbf{G}$, TR_S is static. Also notice that TR_S only has two dimensions and does not include the altitude information. We do this because there is a very limited number of altitude levels from both the convective weather and atmospheric datasets, and therefore we can perform quick altitude matching by simply using grouping and indexing (as in the SQL techniques). The temporal tree TR_T is a one-dimensional k-d tree that constructed based on the elapsed time (e.g., seconds) from a pre-specified baseline time (e.g., 01/01/2013 00:00 Zulu) for each convective weather data entry.

To perform efficient 4D matching, we first generate feature cube grids $\mathbf{F}^G$ for all flight tracks in a batch mode using Algorithm 5. Second, we batch query into the spatial tree TR_S using the 2D feature cube grids, which returns the indices of the closest grid points from the georeferencing system $\mathbf{G}$ to the feature cube grids (Algorithm 6: line 9). Third, we further batch query into the temporal tree TR_T using the timestamps of the feature cube grids, which aligns the original 4D trajectory, with a maximal time distance bound, and collect the closest time indices from the weather dataset to the feature cube grids (Algorithm 6: lines 10 – 11). Fourth, we group the feature cube grid by their altitudes. However, since storms usually propagate fast, we should not only consider the convection for the current altitude level, but also convection below and above. Therefore, within each group, we also specify an altitude buffer, and subset all convective weather datasets that are within the altitude range (Algorithm 6: lines 13 – 15). Lastly, we use the calculated three set of indices – altitude indices, 2D spatial indices, and temporal indices – to collect the matched convective weather, which is an array with dimension (k, n_x, n_y), where k is the number of altitude levels within the altitude buffer, for each feature cube grid. Lastly, for each matched feature cube grid, we overlay all k layers of matched weather arrays so that the resulting array has dimension $(1, n_x, n_y)$, in which each element is 1 if there was convection at any layers of the corresponding grid, and 0 the otherwise (Algorithm 6: line 14).

Notice that the above description only presents our matching procedure for the convective weather dataset. For atmospheric datasets, however, the process is very similar except that we don't specify altitude buffers (Algorithm 6: line 17). Therefore, the final matched feature cube F_t for each feature cube grid has 4 layers – each with dimension $(1, n_x, n_y)$ – where the first layer is a binary array indicating whether there was convective weather, the second layer is a numeric array representing air temperature, the last two layers are also numeric arrays indicating westerly and southerly wind speed, respectively.

4.5 Module Design

Aircraft that "fly as filed" frequently deviate from their pre-departure last filed flight plans for the various reasons discussed above. To predict such "deviations", or equivalently the actual flight trajectories, we formulate the problem as a combination of *translation* and *sequence generation problem*, in which we recurrently use weather information to "translate" a flight plan to an actual flight trajectory. Specifically, our model integrates three modules:

Algorithm 6 Feature Cube Matching (FCM)

```
    Inputs
        $\mathbf{G} = \{(x, y)_i\}, \forall i \in \{1, 2, \dots, N\}$: 2D georeferencing system;
        $\mathbf{F}^G, L^G_{atm}, L^G_{wx}$: Outputs from Algorithm 5;
        $\mathcal{D}_{atm}(I_t, I_x, I_y, I_z) = \{D^{atm}_t\}, \forall t \in \Omega_{atm} = \{1, 2, \dots, M\}$: Atmospheric datasets indexed by time $I_t$, and 3D locations $I_x, I_y$, and $I_z$, where $D^{atm}_t$ contains wind speeds and air temperature information (as matrices) at time $t$;
        $\mathcal{D}_{wx}(I_t, I_x, I_y, I_z) = \{D^{wx}_t\}, \forall t \in \Omega_{wx} = \{1, 2, \dots, K\}$: Convective weather datasets indexed by time $I_t$, and 3D locations $I_x, I_y$, and $I_z$, where $D^{wx}_t$ contains convective weather information (as binary matrices) at time $t$.
    Parameters
        $A = 20000$ ft: Altitude buffer.
        $B = 1$ hour: Maximal time bound.
    Outputs
        $\mathbf{F} = [F_1, \dots, F_T]$: Sequence of feature cubes.
procedure FCM
    Initialize: empty array $\mathbf{F}[1 \dots T][1 \dots 4][1 \dots n_x][1 \dots n_y]$
    Construct spatiotemporal $k$-d trees:
     $TR_S = kdtree(\mathbf{G})$
     $TR^{atm}_T = kdtree(\Omega_{atm})$
     $TR^{atm}_T = kdtree(\Omega_{wx})$

    Batch query:
     $I'_x, I'_y = TR_S.query(\mathbf{F}^G)$
     $I^{atm'}_t = TR^{atm}_T$.query($\mathbf{F}^G$, max bound$= B$)
     $I^{wx'}_t = TR^{wx}_T$.query($\mathbf{F}^G$, max bound$= B$)

    for $I'_z$ in $L^G_{wx}$.groupby($z$) do
        $\mathbf{F}[I^{wx'}_t][1][I'_x][I'_y] = 1 - \prod \left\{1 - \mathcal{D}_{wx}\Big(I^{wx'}_t, I'_x, I'_y, (I'_z - A) : (I'_z + A)\Big)\right\}$
    end for
    for $I'_z$ in $L^G_{atm}$.groupby($z$) do
        $\mathbf{F}[I^{atm'}_t][2 \dots 4][I'_x][I'_y] = \mathcal{D}_{atm}\Big(I^{atm'}_t, I'_x, I'_y, I'_z\Big)$
    end for
end procedure
```

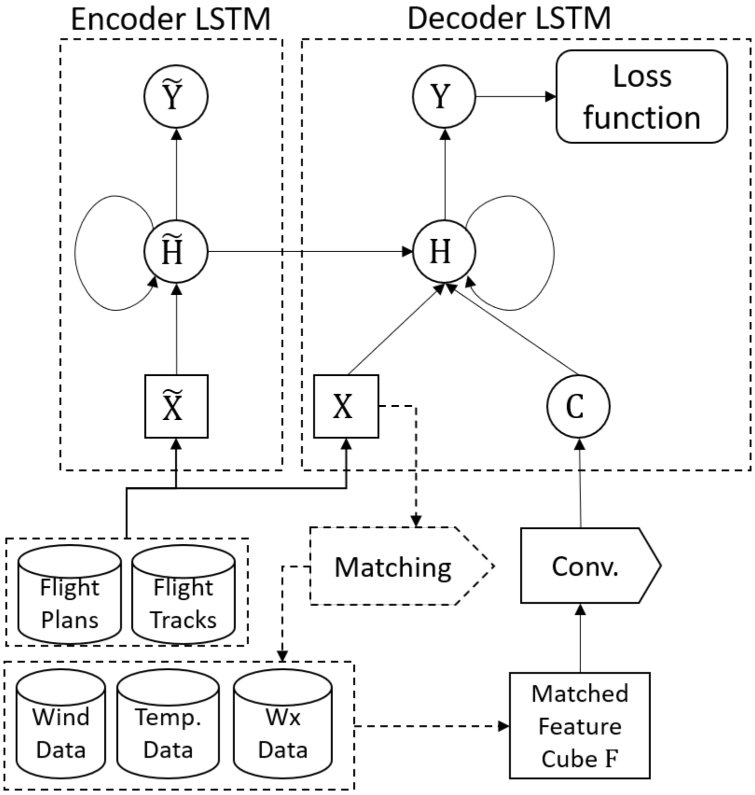


Figure 4.7: Training Framework

1. an encoder LSTM to embed flight plans into a fixed-size feature vector;
2. a decoder LSTM that maps the fixed-size feature vector to the target flight trajectory sequence;
3. a set of convolutional layers that are integrated into the decoder network to condense high-dimension weather-related feature cubes into fixed-size feature representations.

Figure 4.7 shows the general structure of the framework (unrolled LSTM). In the figure, the five cylinders represent raw datasets, from which we extracted the flight plan state $\tilde{\mathbf{X}}$, aircraft state $\mathbf{X}$, and feature cubes $\mathbf{F}$. The "Conv." module utilizes convolutional layers to map the feature cubes $\mathbf{F}$ to some fixed-size feature representation $\mathbf{C}$. The first box shows the encoder LSTM, which maps the flight plan to some hidden states $\tilde{\mathbf{H}}$, while the second box shows the decoder LSTM that maps the aircraft state $\mathbf{X}$, encoded $\tilde{\mathbf{H}}$, and $\mathbf{C}$ to a parameter set $\mathbf{Y}$ that later constitutes the loss function. The detailed network architecture will be explained the following subsections.

4.5.1 Network Architecture

Convolutional Layers

Convolutional neural network (CNN) has achieved great success in image recognition ([44], [65], [70], [38]) and object detection ([35], [34], [62]). As is also well known, deep CNN is not only capable of classification tasks, but also learning feature representations from high-dimensional input signals. Using the feature engineering techniques introduced in Section 4.4, each actual track point P_t will be matched with a high-dimension feature cube F_t. The feature cube can be treated as a multi-channel "image" whose width and height are decided by the size of the relevant regions around the aircraft location (i.e., n_x, n_y), and number of channels is the number of weather-related features considered. In this work, the feature cube has four channels – westerly and southerly wind speeds, air temperature, and convective weather. To extract feature representation from the feature cube, we employ multiple convolutional layers. Specifically, at each timestamp, our convolutional layers use small filters to map a feature cube into a fixed-size feature vector and *directly* feed to the decoder LSTM network (i.e., this is an end-to-end neural network). During the training process, therefore, the loss will be back propagated to all weights of the convolutional layers. If we denote $\mathcal{F}_{\mathcal{C}} : \mathbf{F} \longrightarrow \mathbf{C}$ the convolutional layers with weight matrix $\mathbf{W}_c$, then we can describe the above mapping in Equation 4.1.

$$C_t = \mathcal{F}_{\mathcal{C}}(F_t, \mathbf{W}_c) \tag{4.1}$$

Encoder-decoder LSTM

Long Short-Term Memory (LSTM) networks have been shown to be an effective tool to learn representations from sequential data with long temporal dependencies. Inspired by the success of [69] to solve neural translation problems, we develop an LSTM-based encoder-decoder architecture to map flight plans to actual flight trajectories. Let $\mathcal{F}_{\mathcal{E}} : \tilde{\mathbf{X}} \longrightarrow \tilde{\mathbf{H}}$ denote the encoder LSTM network that maps the 2D flight plan sequence $\tilde{\mathbf{X}}$ (i.e., 2D aircraft state in the nomenclature) to fixed-length hidden states $\tilde{\mathbf{H}}$, let $\tilde{\mathbf{W}}$ further denote the weight matrix of the encoder LSTM $\mathcal{F}_{\mathcal{E}}$, then we have Equation 4.2 to describe the transitions between $\tilde{\mathbf{X}}$ and $\tilde{\mathbf{H}}$, in which $\tilde{X}_t$ and $\tilde{H}_t$ are respectively the elements of $\tilde{\mathbf{H}}$ and $\tilde{\mathbf{X}}$. Notice that since only the hidden state enters the decoder network, we have discarded the output $\tilde{Y}_t$ in later discussions.

$$(\tilde{Y}_t, \tilde{H}_t) = \mathcal{F}_{\mathcal{E}}(\tilde{H}_{t-1}, \tilde{X}_t, \tilde{\mathbf{W}}), \forall t \in \{1, 2, \ldots, \tilde{T}\} \tag{4.2}$$

The decoder network LSTM, denoted as $\mathcal{F}_{\mathcal{D}} : (\mathbf{X}, \tilde{H}_{\tilde{T}}, \mathbf{C}) \longrightarrow Y$, maps aircraft state, encoded flight plan hidden state and feature representation (see **Convolutional layers**) to a parameter set Y that best describes the spatiotemporal patterns of the actual flight trajectories. Specifically, we assume that the aircraft state X_t at each timestamp t follows a Gaussian mixture with K Gaussian components and is parameterized by a vector $Y_t = \{\phi_t^i, \mu_t^i, \Sigma_t^i\}, \forall i \in \{1, 2, \ldots, K\}$, where ϕ_t^i, $\mu_t^i = [\mu_x^i, \mu_y^i, \mu_z^i, \mu_{\dot{x}}^i, \mu_{\dot{y}}^i]_t$, and Σ_t^i are respectively the weight, mode, and covariance matrix of the i^{th} Gaussian component. To further simplify the model, we have also assumed the covariance matrix for each Gaussian component Σ_t^i is block diagonal, with $\Sigma_t^{(i,1)}$ a full 3×3 covariance matrix for latitude, longitude, and altitude (x, y, z), and $\Sigma_t^{(i,2)}$ a full 2×2 covariance matrix for latitude speed and longitude speed $(\dot{x}, \dot{y})$. Lastly, we let the parameter set $\mathbf{Y} = \{Y_1, \ldots, Y_T\}$ learned by the decoder LSTM network. Therefore, if we let $\mathbf{W}_d$ denote the weight matrix of $\mathcal{F}_\mathcal{D}$, then Equation 4.3 and Equation 4.4 describe the above functional relations.

$$\begin{aligned} X_t &\sim \sum_{i=1}^{K} \phi_t^i \cdot \mathcal{N}(\mu_t^i, \Sigma_t^i) \\ \Sigma_t^i &= \begin{bmatrix} \Sigma_t^{(i,1)} & 0 \\ 0 & \Sigma_t^{(i,2)} \end{bmatrix} \end{aligned} \tag{4.3}$$

$$\begin{aligned} (Y_t, H_t) = \mathcal{F}_\mathcal{D}(H_{t-1}, X_t, C_t, \mathbf{W} | \tilde{H}_{\tilde{T}}), \forall t \in \{1, 2, \ldots, T\} \\ Y_t = \{\phi_t^i, \mu_t^i, \Sigma_t^i\}, \forall i \in \{1, 2, \ldots, K\} \end{aligned} \tag{4.4}$$

While the above two equations can in theory achieve our goal, it almost always suffers from the numerical stability issues in practice, and namely, exploding weight parameters ϕ and indefinite covariance matrix Σ. Therefore, we have conducted two further steps. First, to ensure the positive definiteness of the covariance matrix, we let the decoder LSTM outputs two lower triangular matrices $L_t^{(i,1)}$ and $L_t^{(i,2)}$ that further construct the covariance matrices $\Sigma_t^{i,1}, \Sigma_t^{i,1}$ using Cholesky decomposition. Therefore, we can rewrite Equation 4.4 to Equation 4.5. Second, we have employed an additional softmax layer to the output that corresponds to the weight parameter of the Gaussian mixture, as stated in Equation 4.6. Lastly, to make equal number of operations, we further add a nonlinear mapping to the mode parameters as in Equation 4.7, where f can be any differentiable nonlinear functions (e.g., exponential linear unit activation). Notice that the above three operations involve matrix multiplications whose weights, which are not explicitly expressed by Equations 4.6 and 4.7, are also part of $\mathbf{W}$ and thus need to be learned/optimized.

$$\begin{aligned} (Y_t, H_t) = \mathcal{F}_\mathcal{D}(H_{t-1}, X_t, C_t, \mathbf{W} | \tilde{H}_{\tilde{T}}), \forall t \in \{1, 2, \ldots, T\} \\ Y_t = \{\phi_t^i, \mu_t^i, L_t^{(i,1)}, L_t^{(i,2)}\}, \forall i \in \{1, 2, \ldots, K\} \\ \Sigma_t^{(i,1)} = L_t^{(i,1)} \cdot L_t^{(i,1)\top} \\ \Sigma_t^{(i,2)} = L_t^{(i,2)} \cdot L_t^{(i,2)\top} \end{aligned} \tag{4.5}$$

$$\phi_t^i = \frac{e^{\phi_t^i}}{\sum_{j=1}^{K} \phi_t^j}, \forall i \in \{1, 2, \ldots, K\} \tag{4.6}$$

$$\mu_t^i = f(\mu_t^i), \forall i \in \{1, 2, \ldots, K\} \tag{4.7}$$

Lastly, we have employed the negative log likelihood to be the loss function of the proposed neural network, and we can therefore obtain the weights $\mathbf{W}_c$, $\mathbf{W}_d$, and $\tilde{\mathbf{W}}$ by optimizing the loss through gradient-based method. The full loss function is formulated by Equation 4.8, where $Pr(\cdot)$ is the probability density function of Gaussian distribution, and X_t^j is aircraft state at time t for the j^{th} aircraft.

$$\mathcal{L}(\mathbf{W}_d, \mathbf{W}_c, \tilde{\mathbf{W}}) = -\sum_{j=1}^{N}\sum_{t=1}^{T} \log\left[\sum_{i=1}^{K} \phi_t^i \cdot Pr(X_t^j|\mu_t^i, \Sigma_t^i)\right] \tag{4.8}$$

4.5.2 Inference Process

During the inference time, we are trying to answer what the rest of trajectory will be, if we have known an aircraft's last filed flight plan, first T' states ($1 \leq T' \leq T$), and the corresponding first T' weather-related feature cubes. One simple solution is to directly feed the flight plan $\tilde{X}_t, \forall t = \{1, 2, \ldots, \tilde{T}\}$, first T' states and feature cubes $X_t, F_t, \forall t = \{1, 2, \ldots, T'\}$ to the pre-trained model, and obtain the set of Gaussian mixture parameters $Y_t = [\phi_t^i, \mu_t^i, \Sigma_t^i], \forall i = \{1, 2, \ldots, K\}, \forall t = \{1, 2, \ldots, T'\}$. Then we use the Gaussian mixture to sample one aircraft state, say $\hat{X}_{T'+1}$, and use Algorithms 5 and 6 (in the recursive mode) to match the generated point $\hat{X}_{T'+1}$ with weather data and obtain $F_{T'+1}$. By recursively repeating the process until the last timestamp T, we can generate a whole flight trajectory after time T'. Although simple as it is, the predicted sequence tends to have very large variance, especially towards the end of the sequence. One may argue that this can be tackled by sampling multiple points, however, in practice, this often ends up with "zigzagged" predicted track points due to the symmetry property of the Gaussian distribution. Therefore, instead of using sampling techniques in most literature (e.g., [47], [1]), we have abandoned the idea of using any sampling process from the learned Gaussian mixture, but directly use the inferred parameter set (e.g., the mode vectors μ^i and covariance matrices Σ^i from $\mathbf{Y}$) as the predicted aircraft state $\hat{\mathbf{X}}$. These, however, creates two more problems: (a) the number of predicted states at each timestamp grows exponentially (with base K); (b) the predicted states still suffer from large variance – one falsely predicted point will induce uncontrollable variance for any later predictions in the sequence. Therefore, we have employed three main techniques to further tackle these issues: adaptive Kalman filter (AKF), beam search (BS), and Rauch-Tung-Striebel (RTS) smoother ([61]).

We first present the AKF in Algorithm 7, which contains two steps: Kalman update and gating. In the first step, the predicted state $\hat{X}_{t-1}$ and its covariance matrix $\hat{\Sigma}_{t-1}$ is used to produce a estimates for the next timestamp using basic kinematic equations (Algorithm 7: line 3, Predict process). Then the estimates are combined with the predicted measurement Y_t from the pre-trained encoder-decoder LSTM model to further produce residual estimate $\mathbf{R}_t$ and best predictions $\hat{X}_t, \hat{\Sigma}_t$ (Algorithm 7: line 3, Update process). In the second step, we compare the magnitude of the residual measurement $\mathbf{R}_t$ with some pre-defined thresholds e_1, e_2 to decide whether the predicted measurement Y_t is an outlier (Algorithm 7: line 5), or maneuver (Algorithm 7: line 6), which require reiterating step 1 using a larger initial process noise covariance matrix $\mathbf{Q}$, or others (Algorithm 7: line 7).

Algorithm 7 Adaptive Kalman Filter with Gating (AKF)

```
    Inputs
        $\hat{X}_{t-1}, \hat{\Sigma}_{t-1}$: Predicted aircraft state and covariance matrix at time $t-1$;
        $Y_t = [\mu_t, \Sigma_t]$: Measurements at time $t$, i.e., output from decoder LSTM at time $t$.
    Parameters    /* Default values provided in Section 4.5.3 */
        $\mathbf{A}$: Aircraft dynamic matrix;
        $\mathbf{Q}$: Initial process noise covariance matrix;
        $\mathbf{H}$: Measurement matrix;
        $e_1, e_2$: Outlier and maneuver gatings;
        $q_s$: Maneuver scaling factor.
    Outputs
        $\hat{X}_t, \hat{\Sigma}_t, M_t$: Kalman filter estimates (i.e., predictions) $\hat{\mathbf{X}}_t, \hat{\Sigma}_t$ at time $t$ with gating indicator $M_t$, $M_t = 1$ if outlier and 0 the otherwise.
procedure AKF
    Kalman Update:
     Predict:
                    $\underline{X_t} = \mathbf{A} \cdot \hat{X}_{t-1}$
                    $\underline{\Sigma_t} = \mathbf{A} \cdot \hat{\Sigma}_{t-1} \cdot \mathbf{A}^\top + \mathbf{Q}$
     Update:
                    $\mathbf{S}_t = \mathbf{H} \cdot \underline{\Sigma_t} \cdot \mathbf{H}^\top + \Sigma_t$
                    $\mathbf{K}_t = \underline{\Sigma_t} \cdot \mathbf{H}^\top \cdot \mathbf{S}_t^{-1}$
                    $\mathbf{R}_t = \mathbf{H} \cdot (\mu_t - \underline{X_t})$
                    $\hat{X}_t = \underline{X_t} + \mathbf{K}_t \cdot \mathbf{R}_t$
                    $\hat{\Sigma}_t = (\mathbf{I} - \mathbf{K}_t \cdot \mathbf{H}) \cdot \underline{\Sigma_t}$
    Gating:
     If $\|\mathbf{R}_t\| > e_1$: $M_t = 1$, return $\hat{X}_t, \hat{\Sigma}_t, M_t$
     Else If $\|\mathbf{R}_t\| > e_2$: $M_t = 0, \mathbf{Q} = q_s \cdot \mathbf{Q}$, go to Kalman Update
     Else: $M_t = 0$, return $\hat{X}_t, \hat{\Sigma}_t, M_t$.
end procedure
```

Second, we show in Figure 4.8 and Algorithm 8 the details of our proposed inference procedure. In the figure, the cylinders are raw datasets from which we extracted flight plans $\tilde{\mathbf{X}}$, aircraft state $\mathbf{X}$, and feature cube $\mathbf{F}$. The left-most box contains the pre-trained encoder LSTM, and from left to right we show the recursive feed-forward pass and feature matching using the pre-trained decoder LSTM (and convolutional layers). Notice that in each of these box, we have integrated two modules "AKF" and "BS" to reduce variance and increase the predictability. The top part of the figure shows the final step – RTS smoother – of our sampling process that further improves the variance and smoothness of the predictions.

To be more specific, at time T', which is the last timestamp of the provided flight trajectory, we first perform a feed forward pass using the pre-trained encoder-decoder LSTM and collect the hidden states of the last timestamp H_t (Algorithm 8, lines 3 – 9). Then we use it to recursively feed forward the decoder LSTM (with CNN layers) to obtain the parameter set of the Gaussian mixture $Y_t = [\phi_t^i, \mu_t^i, \Sigma_t^i], \forall i = \{1, 2, \ldots, K\}$ (Algorithm 8, lines 13 – 15). Third, for each Gaussian component, we use the adapted Kalman filter (see Algorithm 7 for details) to obtain the best estimates from previous estimates and LSTM models. We further use Equation 4.9 to calculate a cumulative log likelihood for each Gaussian component, where function $l(M_t^i = 1)$ gives a penalty if μ_t^i is considered as an outlier (Algorithm 8, lines 17 – 18), and π_1 and π_2 assign weights to respectively the log of Gaussian component weight ϕ_t^i and log of Gaussian probability density function $Pr(\cdot)$. Typically we set π_1 to be larger because selecting the correct Gaussian component in the Gaussian mixture appears to be much more important. Also notice that from lines 13 to 20 of Algorithm 8, each predicted aircraft state from previous timestamp will produce K more estimates (due to Gaussian mixture), therefore, to avoid exponential growth of the predicted sequences, we keep tracing back each estimate its parent state and cumulating its log likelihood throughout the whole sequence. This operation, which is also known as beam search, eventually allows us to limit the sampling size at each timestamp and only keep the best top N_{BS} sequences by sorting the cumulated "sequential" log likelihood (Algorithm 8, lines 18 – 21). Forth, we use Algorithms 5 and 6 to match the best N_{BS} aircraft state estimates with weather features and repeat the above steps recursively until the last timestamp $T + T'$. Lastly, we use Rauch-Tung-Striebel smoother to further improve the prediction (Algorithm 8, line 24).

$$\mathcal{L}_t^i = \mathcal{L}_t^J + \pi_1 \cdot \log \phi_t^i + \pi_2 \cdot \log Pr(\mu_t^i | \mu_t^i, \Sigma_t^i) + l(M_t^i), \pi_1 + \pi_2 = 1 \tag{4.9}$$

4.5.3 Implementation Details

Feature Engineering

In our feature engineering process, the horizontal size of each feature cube grid is chosen as 2° × 2° latitude-longitude with resolution 20 × 20, which approximately covers a 120-nautical-mile by 120-nautical-mile square region in front of the aircraft. As a result, the dimension of each feature cube is 4 × 20 × 20, where the first dimension is constituted by convective weather, westerly and southerly wind speeds, and air temperature. All the

Algorithm 8 Trajectory Generator (TG)

```
    **Inputs**
        $\mathcal{F}_{\mathcal{C}}, \mathcal{F}_{\mathcal{E}}, \mathcal{F}_{\mathcal{D}}$: Pre-trained encoder-decoder LSTM neural networks;
        $\tilde{\mathbf{X}} = [\tilde{X}_1, \dots, \tilde{X}_{\tilde{T}}]$: Last filed flight plan sequence;
        $\mathbf{X} = [X_1, \dots, X_{T'}]$: Initial actual flight state sequence up to time $T'$;
        $\mathbf{F} = [F_1, \dots, F_{T'}]$: Initial matched feature cubes corresponding to $\mathbf{X}$.
    **Parameters**    /* Default values provided in Section 4.5.3 */
        $\pi_1, \pi_2$: Log likelihood weights;
        $N_{BS}$: Size of the beam search;
        $T$: Length of the full predicted trajectory;
        $l(M_t)$: Penalty function for outlier samples.
    **Outputs**
        $\mathbb{S} = \{\hat{\mathbf{X}}_j = [\hat{X}^j_{T'+1}, \dots, \hat{X}^j_T], 1 \le j \le N_{BS}\}$: The set of $N_{BS}$ best predicted sequences.
procedure TG
    Initialize:$\mathbb{S} = \{\hat{\mathbf{X}}_j[1 \dots T], \forall j \in \{1, \dots, N_{BS}\}\}$
    Feed Forward Pass:
    for $t := 1$ to $\tilde{T}$ do
        $(\tilde{H}_t, \dots) = \mathcal{F}_{\mathcal{E}}(\tilde{H}_{t-1}, \tilde{X}_t, \tilde{\mathbf{W}})$    /* feed forward encoder LSTM */
    end for
    for $t := 1$ to $T'$ do
        $C_t = \mathcal{F}_{\mathcal{C}}(F_t, \mathbf{W}_c)$    /* feed forward CNN layers */
        $(H_t, \dots) = \mathcal{F}_{\mathcal{D}}(H_{t-1}, X_t, C_t, \mathbf{W} | \tilde{H}_{\tilde{T}})$    /* feed forward decoder LSTM */
    end for
    Inference: set $\hat{X}^j_{T'} = X_{T'}, \hat{\Sigma}^j_{T'} = \mathbf{0}, \forall j \in \{1, \dots, N_{BS}\}$
    for $t := T' + 1$ to $T + T'$ do
        for $j := 1$ to $N_{BS}$ do
            $C_t = \mathcal{F}_{\mathcal{C}}(F_t, \mathbf{W}_c)$
            $(\phi^i_t, \mu^i_t, \Sigma^i_t, \dots) = \mathcal{F}_{\mathcal{D}}(H_{t-1}, \hat{X}^j_t, C_t, \mathbf{W} | \tilde{H}_{\tilde{T}}), i \in \{1, \dots, K\}$
            for $i := 1$ to $K$  do    /* Loop over Gaussian components */
                $(\hat{X}^{i,j}_t, \hat{\Sigma}^{i,j}_t, M^{i,j}_t)$ =AKF$(\hat{X}^j_{t-1}, \hat{\Sigma}^j_{t-1}, \mu^i_t, \Sigma^i_t)$    /* Adaptive Kalman filter */
                $\mathcal{L}^{i,j}_t = \mathcal{L}^j_t + \pi_1 \cdot \log \phi^i_t + \pi_2 \cdot \log Pr(\hat{X}^{i,j}_t | \mu^i_t, \Sigma^i_t) + l(M^{i,j}_t)$
            end for
        end for
        Sort $\mathcal{L}^{i,j}_t$ and append $(\hat{X}^{i,j}_t, \hat{\Sigma}^{i,j}_t)$ associated with the top $N_{BS}$ $\mathcal{L}^{i,j}_t$ to $\mathbb{S}$.
        $(F^G_t, \dots)$ =FCGG$(\hat{X}^{i,j}_t)$, $F_t$ =FCM$(\hat{X}^{i,j}_t, F^G_t \dots)$    /* Feature cube matching */
    end for
    $\mathbb{S} = \{\text{RTS}(\hat{\mathbf{X}}_j), 1 \le j \le N_{BS}\}$
end procedure
```

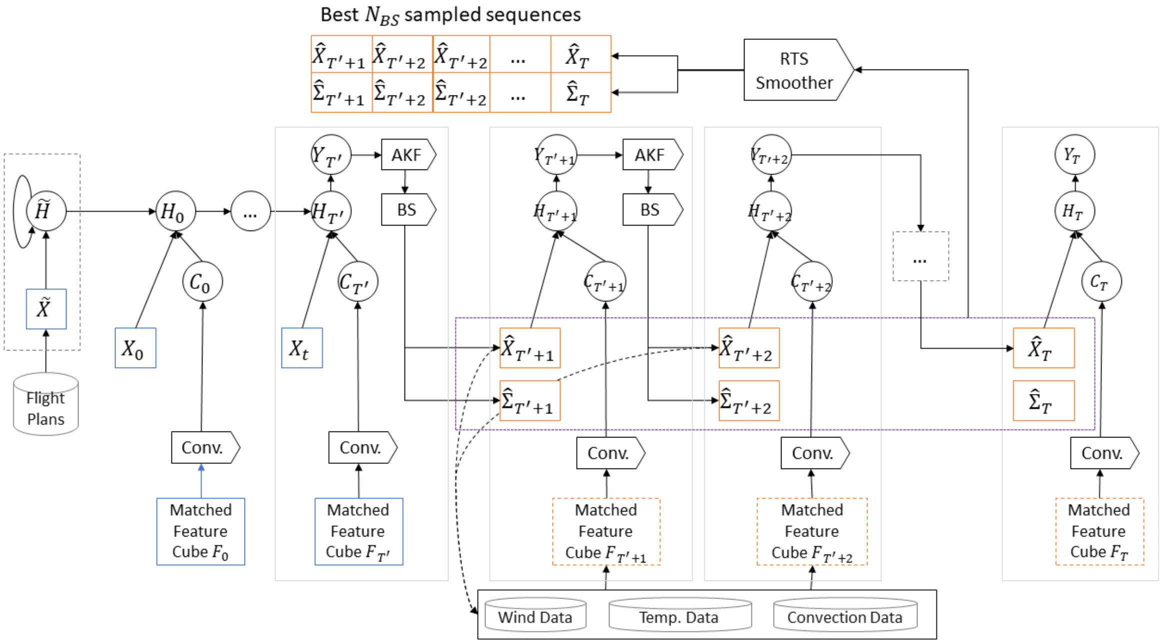


Figure 4.8: Inference Framework

features except the convective weather are normalized to zero mean and unit variance before feeding into the neural networks.

To potentially transfer our model to a variety of airport pairs, we have also subtracted the 3D coordinates (latitude x, longitude y, altitude z) of the origin airport from both the last filed flight plans $\tilde{\mathbf{X}}$ and actual flight tracks $\mathbf{X}$, such that our model learns, instead of the absolute coordinates of aircraft locations, the deviations away from the origin airport. Similar to the feature engineering, we normalize those outcomes to zero mean and unit variance.

Training

For the encoder network, we first use an embedding layer with dimension 32 for the flight plan coordinates before feeding them to the LSTM. Our encoder LSTM has two layers, and each layer has a fixed-size hidden state with dimension 128. For the decoder network, we have adopted two-layer LSTM with 128-dimensional hidden state, which takes the last hidden state from the encoder LSTM as its initial state. We have also chosen $K = 3$ Gaussian mixture components to model the probabilistic distributions of actual flight tracks. Therefore, at each timestamp, the size of the decoder LSTM output parameter set $\mathbf{Y}$ is 45 (3 components, each with 15 parameters = 1 weight + 5 modes + 9 elements from the Cholesky decomposed lower triangular matrices).

The convolutional layers in the decoder network learn representations from the weather feature cubes. Four layers – three convolutional and one fully-connected – are proposed and

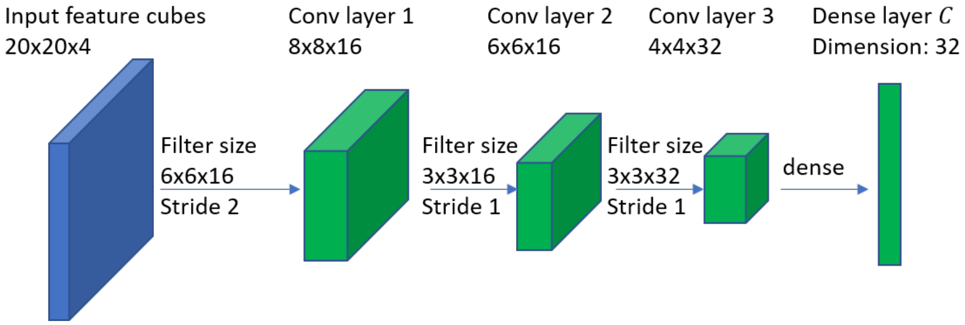


Figure 4.9: Structure of the Convolutional Layers

the overall structure is summarized in Figure 4.9. The first layer inputs the feature cube with 16 filters of size $6 \times 6 \times 4$ and stride 2. The second layer uses 16 filters of a smaller size $3 \times 3 \times 16$ and stride 1, and the third layer uses 32 filters of $3 \times 3 \times 16$ and stride 1. The fully connected layer has 32 neurons. Notice that pooling and padding operations are not employed since locational information of the weather significantly affects the aircraft state in our problem domain.

Feature representations $\mathbf{C}$ from the convolutional layers and aircraft state variables $\mathbf{X}$ are fed into an embedding layer with 64 neurons before entering the decoder LSTM. The outputs of the LSTM will then be mapped to the estimates of parameters of Gaussian mixture using a dense layer. In our architecture, we choose the exponential linear unit (ELU) for all activation functions.

Inference

In calculating the cumulative log likelihood, we assign a higher weight to the probability (ϕ_t^i) of Gaussian component, specifically, $\pi_1 = 0.8, \pi_2 = 0.2$.

In the adaptive Kalman filter, we have assumed a simple linear dynamic (Equation 4.10) where at each timestamp the aircraft moves at a constant horizontal speed and zero vertical speed ($\ddot{x} = \ddot{y} = \dot{z} = 0$). The changes in horizontal speed and altitude are captured by the measurement (predictions from pre-trained neural networks) and are treated as the errors in the linear dynamic system. The process error $\mathbf{Q}$ is assumed to be a diagonal matrix as in Equation 4.11, and the measurement matrix is set to be an identity matrix $\mathbf{H} = \mathbf{I}_{5\times 5}$. We choose two gating thresholds $e_1 = 0.8, e_2 = 0.3$ to signify whether the measurement is an outlier, or a maneuver, or neither. Namely, if the sum of absolute errors of latitude and longitude is larger than e_1, then the measurement is an outlier and a negative value $l(M = 1) = -9$ will be added to the trajectory's cumulative log likelihood (Equation 4.9). If the error is greater than e_2, then the measurement is considered as a maneuver and we increase the process error Q by a factor of $q_s = 10$ and repeat the Kalman filter updating process. Lastly, to ensure the numerical stability, we only use the diagonal of the matrix $\mathbf{S}_t$ in calculating the inverse of the updated covariance matrix.

In the beam search, we pick the sampling size $N_{BS} = K^2 = 9$ during the sampling procedure, and only output one trajectory with the highest log likelihood in the final evaluation.

$$\underline{X}_t = \mathbf{A} \cdot \hat{X}_{t-1}; \quad \mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & 0 & \Delta t \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}; \quad \Delta t = 120s \tag{4.10}$$

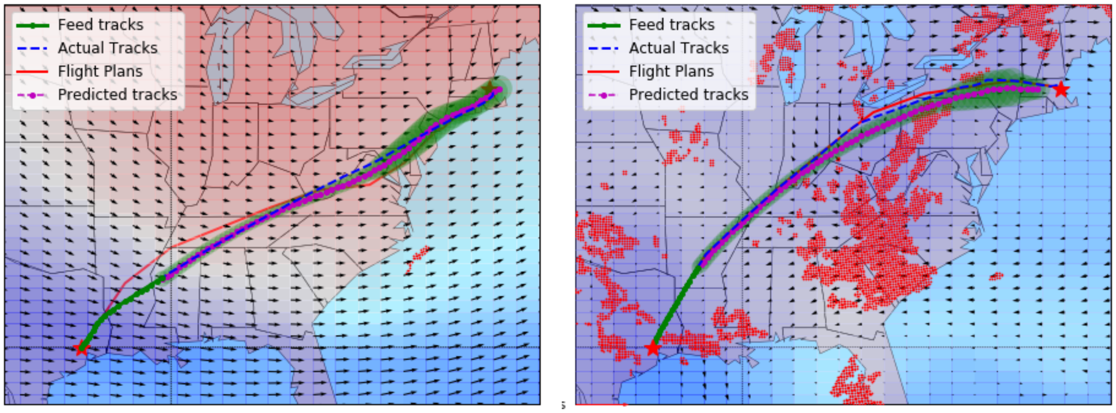
$$\mathbf{Q} = \begin{bmatrix} 10^{-3} & & & & \\ & 10^{-3} & & & \\ & & 1 & & \\ & & & 10^{-6} & \\ & & & & 10^{-6} \end{bmatrix} \tag{4.11}$$

4.6 Experimental Results

We applied our method on a historical flight trajectory dataset from IAH to BOS in the year 2013. The preprocessed dataset contains 1679 flights and is split into two sets, with 80% in the training set and the rest in the evaluation set. In the inference process, we use the first 20 actual track points and their corresponding feature cubes for every flight on the evaluation set as the observed sequence and predict the rest of flight tracks using Algorithm 8. Figure 4.10 illustrates two examples of our sampled trajectories, where the red curve is the last filed flight plan, the green curve represents the first 20 observed flight tracks, the magenta curve is the predicted flight tracks, and the blue dashed curve is the actual flight tracks (ground truth). In the figure, the background color indicates the average air temperature, with warmer color (towards red) as higher temperature. The arrows represent wind directions and speeds, and the red polygons are the convective weather regions. The green band indicates the path with three standard deviations (obtained from the covariance matrix at each timestamp) around predicted flight tracks. In the figure, the left subplot shows the case with slight convective activities, while the right subplot with substantial convection. Both predicted trajectories agree largely with the actual flown tracks (ground truth), with small deviations in the middle parts of the trajectories. However, those deviations are mostly covered by the prediction intervals. We have also noticed that towards the end of the predictions, the confidence intervals are larger. This might be caused by the fact that aircraft in general has more complex approaching patterns, typically by maneuvers in the terminal airspace.

We report four errors listed below to evaluate the performance of the proposed model. Notice that the point-wise vertical error is signed error while the other three errors are unsigned.

1. Point-wise horizontal error (PHE). The distance (in nautical mile) between every predicted point's 2D coordinate (latitude and longitude) and the ground truth.



(a) Prediction Example 1

(b) Prediction Example 2

Figure 4.10: Examples of our Predictions

2. Point-wise vertical error (PVE). The vertical distance (in feet) between every predicted point's altitude and the ground truth.

3. Trajectory-wise horizontal error (THE). The average point-wise horizontal error (in nautical mile) along each trajectory.

4. Trajectory-wise vertical error (TVE). The average point-wise vertical error (in feet) along each trajectory.

Figure 4.11 shows the histograms of the above errors, and Table 4.1 presents the average absolute values of the four errors, denoted respectively as MAPHE, MAPVE, MATHE, MATVE. The distribution of the point-wise horizontal errors is largely skewed towards the left, with an average of 49.39 nautical miles. The point-wise vertical errors, which are signed value, are mostly centered in the range from -5000 ft to 5000 ft. The trajectory-wise horizontal and vertical errors, while both skewed to the left, have similar average absolute values with point-wise errors. We also point out that we do observe large prediction errors for flights that are have very unusual departure procedures (outlier flights), which requires further research to explore possible solutions. Lastly, we report the horizontal error propagation results using Figure 4.12, in which the x axis represents the elapsed time after the last track point fed into the pre-trained model, and the y axis shows the average horizontal error at each of the associated elapsed time instance for all testing flights except for 10 outliers with unusual departure procedures. We have also shown the standard deviation using the error bars. In the figure, we observe a concave shape of the error, indicating clear error propagation through the middle of the predicted trajectories, and convergence towards the arrival phase of the prediction. The standard deviation, similarly, also demonstrates propagation towards the middle of the prediction.

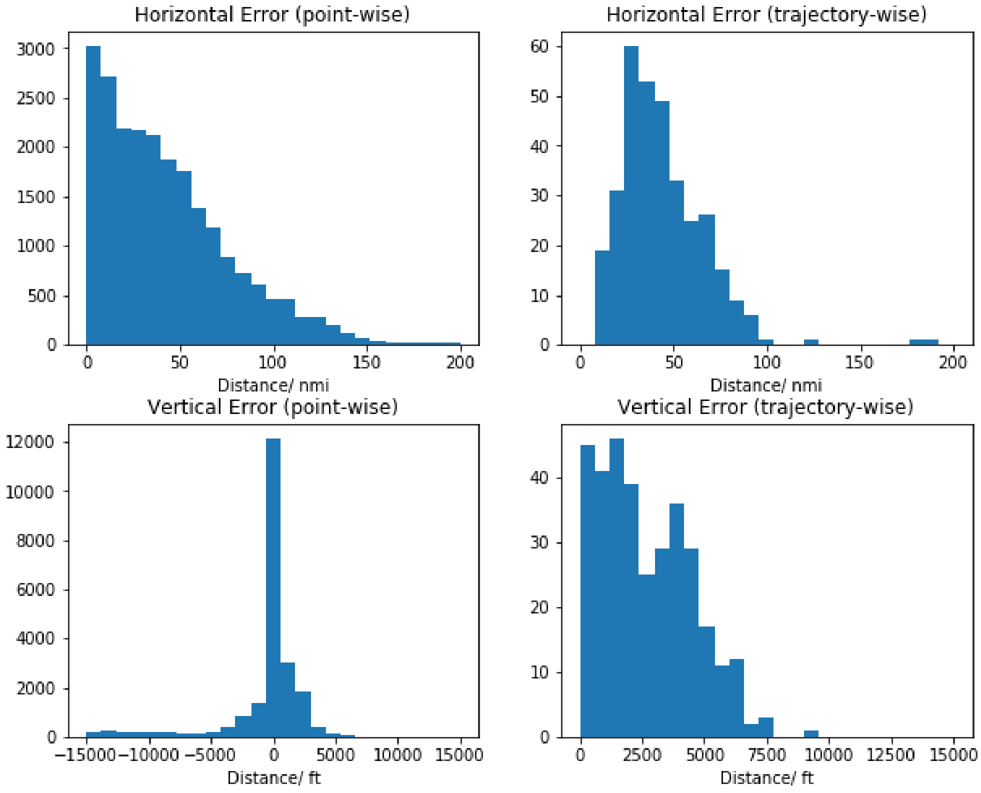


Figure 4.11: Histogram of Prediction Errors

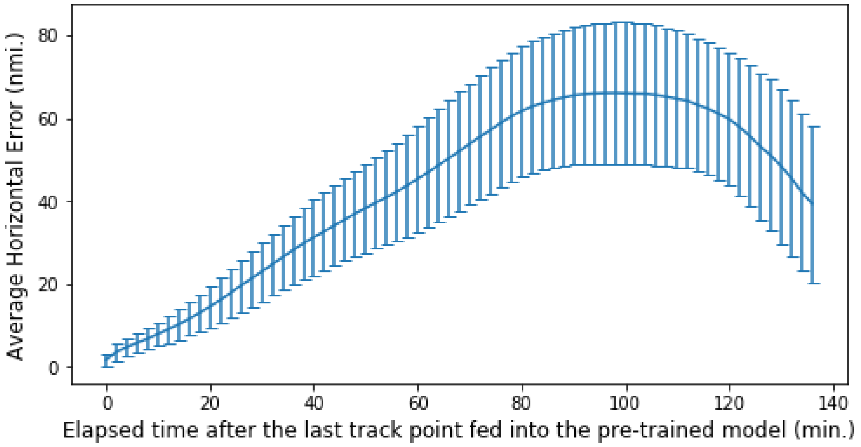


Figure 4.12: Error Propagation

Table 4.1: Average of the Four Absolute Errors

	MAPHE (nmi.)	MAPVE (ft.)	MATHE (nmi.)	MATVE (ft.)
Value	49.39	2669.35	49.44	2690.72

We end this section by visualizing the convolutional layers of our neural networks in Figures 4.13 and 4.14. Figure 4.13a shows 32 arbitrarily chosen input feature cubes, with each column a unique feature cube. The first row of the figure represents the convective weather layer, in which the red polygons identify the location of convection; the second row shows the air temperature layer; and the last two rows show respectively the westerly and southerly wind speeds. In the last three layers, warmer color (towards red) indicates larger numerical value. We feed those random inputs into our convolutional layers of the neural networks and obtain the outputs. Figures 4.13b, 4.13c, and 4.14 show respectively the outputs from the first (width 16), second (width 16), and third (width 32) convolutional layers, in which each column represents the corresponding feed-forward output from Figure 4.13a, and each row shows the results from different filters in the convolutional layers. We first notice that convolutional layer 1 identifies the locations of convective weather, especially filters 1, 2, 7 and 8. Filters 11, 12 and 16 reflect the two wind speed components, and other filters capture the nonlinear relations among the four channels associated with the input feature cubes. Figures 4.13c and 4.14 show the outputs from convolutional layer 2 and layer 3, which largely agree with the first layer but further abstracts the feature space with more nonlinearities.

4.7 Conclusions

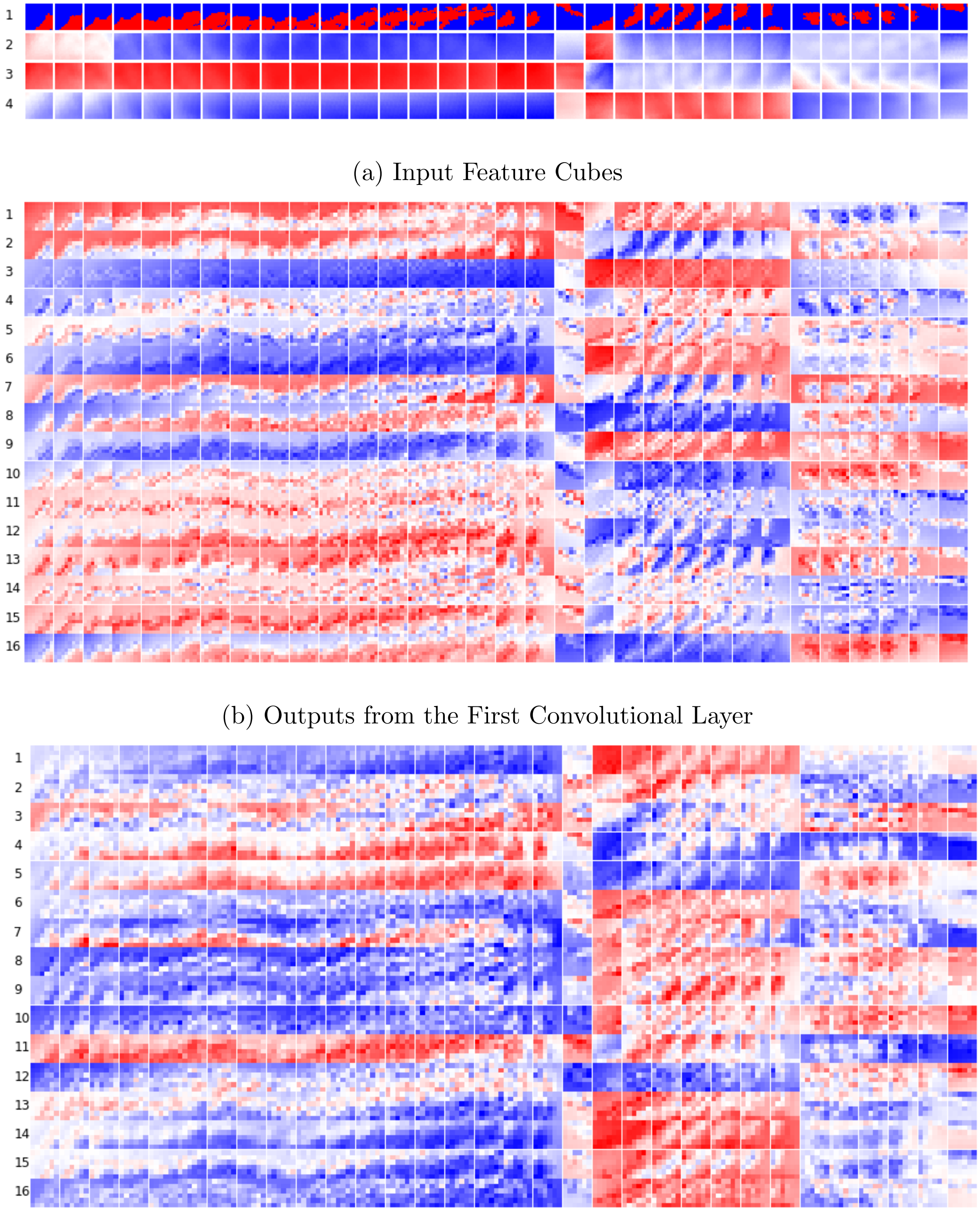
In this research, we propose a novel approach to predict, in a real-time manner, the actual aircraft 4D trajectories, using high-dimension meteorological features and last filed flight plans. To tackle this “sequence to sequence” prediction problem, our approach is composed of a matching algorithm, a deep generative model, a training framework, and an inference framework.

The matching algorithm, which is based on the k-d tree, efficiently 4D matches actual flight tracks with high-fidelity weather-related datasets – convective weather, air temperature, and wind speeds – in the vicinity of the aircraft. The constructed feature space for each track point is based on a novel concept of feature cube grid, which is a spatiotemporal 4D grid surrounding the track point. By querying such grid into the weather datasets, our matching algorithm constructs “image-like” feature cubes, whose width and height are the size of the interested regions around aircraft locations, and number of channels are the number of weather-related features.

In our modeling framework, we model the actual flight states – latitude, longitude, altitude, latitude speed, and longitude speed – as conditional Gaussian mixture with parameters to be learned from our proposed end-to-end deep generative model, which consists of a multi-layer encoder LSTM, a multi-layer decoder LSTM, and a set of convolutional layers. The encoder network inputs the last filed flight plan sequence and produces a fixed-size hidden state variable that is later fed into the decoder network. The convolutional layers learn a lower-dimensional feature representations from the matched feature cubes. The convolutional layers are integrated into the decoder network so that the loss will be back propagated through their weights. The decoder network takes the encoder's output as its initial hidden state and predicts the parameter set of Gaussian mixture based on each timestamp's flight state and feature representation. In the training process, the loss function is the total negative log likelihood.

In the inference process, we first feed the flight plan, observed flight states and their corresponding feature cubes into the trained model and obtain the Gaussian mixture parameters for the first predicted flight state. Thereafter, we recursively apply an adaptive Kalman filter with gating, which improves the predictability by reducing variance, and our matching algorithm, which produces the feature cubes for the best estimated states, and feed the outputs to the decoder LSTM to predict the rest of the flight trajectory. A beam search algorithm is implemented to reduce computational complexity in the recursion, and a RTS smoother is used to further improve the best predicted trajectory.

We apply our model on the datasets for flights from IAH to BOS in the year 2013. 1342 flights are used for training and 337 flights are used for evaluation. By visualizing our convolutional layers, we observe that the learned filters successfully locate the convective weather and generalize the weather-related features well. We use four metrics to measure the prediction error. Both the point-wise and trajectory-wise average absolute horizontal errors are around 50 nautical miles, while the average absolute vertical errors are around 2800 feet. We also observe large prediction errors for (outlier) flights with unusual departure procedures, which will be explored in our future research. Other future works include extending our matching algorithm to more features such as air traffic management initiatives (miles-in-trail, airspace flow program, etc.) and neighboring aircraft.



(a) Input Feature Cubes

(b) Outputs from the First Convolutional Layer

(c) Outputs from the Second Convolutional Layer

Figure 4.13: CNN Feature Representation

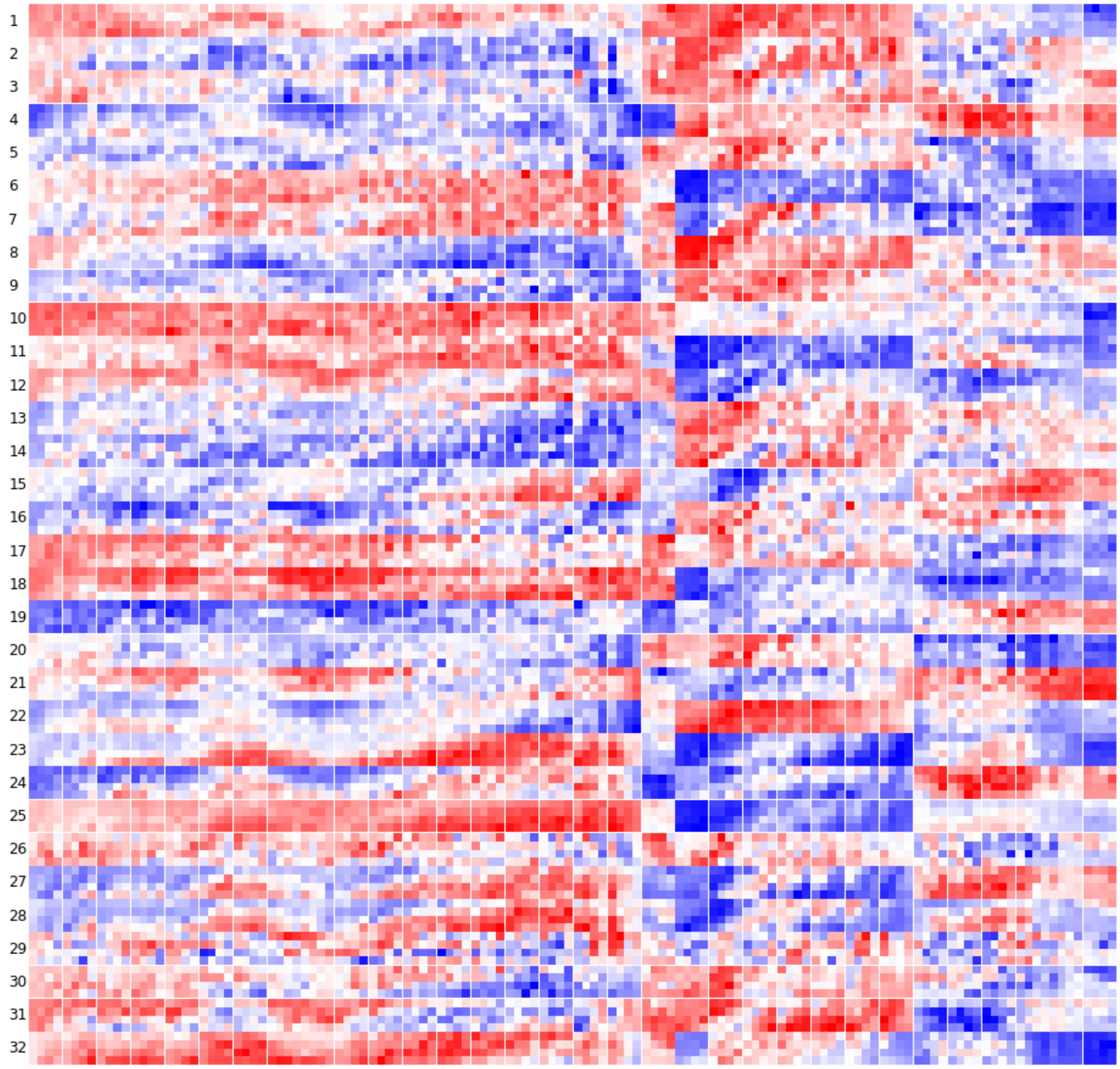


Figure 4.14: Outputs from the Third Convolutional Layer

Bibliography

[1] Alexandre Alahi et al. “Social lstm: Human trajectory prediction in crowded spaces”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 961–971.

[2] Giulio Avanzini. “Frenet-based algorithm for trajectory prediction”. In: *Journal of guidance, control, and dynamics* 27.1 (2004), pp. 127–135.

[3] Samet Ayhan and Hanan Samet. “Aircraft trajectory prediction made easy with predictive analytics”. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2016, pp. 21–30.

[4] Moshe E Ben-Akiva, Steven R Lerman, and Steven R Lerman. *Discrete choice analysis: theory and application to travel demand*. Vol. 9. MIT press, 1985.

[5] Jose V Benavides et al. “Implementation of a trajectory prediction function for trajectory based operations”. In: *AIAA Atmospheric Flight Mechanics Conference*. 2014, p. 2198.

[6] Jon Louis Bentley. “K-d trees for semidynamic point sets”. In: *Proceedings of the sixth annual symposium on Computational geometry*. 1990, pp. 187–197.

[7] Esther Calvo et al. “A new method to validate the route extension metric against fuel efficiency”. In: *2015 ATM Seminar*. 2015.

[8] Gano Chatterji. “Short-term trajectory prediction methods”. In: *Guidance, Navigation, and Control Conference and Exhibit*. 1999, p. 4233.

[9] Feng Cheng and John Gulding. “Computing wind-optimal routes for flight performance benchmarking”. In: *16th AIAA Aviation Technology, Integration, and Operations Conference*. 2016, p. 4361.

[10] Patrick Pakyan Choi and Martial Hebert. “Learning and predicting moving object trajectory: a piecewise trajectory segment approach”. In: *Robotics Institute* 337 (2006), pp. 1–17.

[11] Jan Chorowski et al. “End-to-end continuous speech recognition using attention-based recurrent nn: First results”. In: *arXiv preprint arXiv:1412.1602* (2014).

[12] Junyoung Chung et al. “A recurrent latent variable model for sequential data”. In: *Advances in neural information processing systems*. 2015, pp. 2980–2988.

[13] Council of European Union. *Commission Implementing Regulation (EU) No 390/2013, laying down a performance scheme for air navigation services and network functions.* 2013. URL: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%5C%3A32013R0390.

[14] Arjen De Leege, Marinus van Paassen, and Max Mulder. "A machine learning approach to trajectory prediction". In: *AIAA Guidance, Navigation, and Control (GNC) Conference*. 2013, p. 4782.

[15] Jeffrey Donahue et al. "Long-term recurrent convolutional networks for visual recognition and description". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 2625–2634.

[16] Adric Eckstein. "Automated flight track taxonomy for measuring benefits from performance based navigation". In: *2009 Integrated Communications, Navigation and Surveillance Conference*. IEEE. 2009, pp. 1–12.

[17] Marco Enriquez. "Identifying temporally persistent flows in the terminal airspace via spectral clustering". In: *Tenth USA/Europe Air Traffic Management Research and Development Seminar (ATM2013)/Federal Aviation Administration (FAA) and EUROCONTROL. Chicago, IL, USA*. 2013, pp. 10–13.

[18] Marco Enriquez and Christopher Kurcz. "A simple and robust flow detection algorithm based on spectral clustering". In: *ICRAT Conference*. 2012.

[19] Martin Ester et al. "A density-based algorithm for discovering clusters in large spatial databases with noise." In: *Kdd*. Vol. 96. 34. 1996, pp. 226–231.

[20] EUROCONTROL. *Free Route Airspace developments*. European Commission, 2016.

[21] EUROCONTROL. *Horizontal Flight Efficiency, Achieved Distances*. Accessed: 2016-11-09. URL: https://www.eurocontrol.int/sites/default/files/content/documents/single-sky/pru/news-related/2013-05-08-slides-workshop-achieved-distance.pdf.

[22] EUROCONTROL. *Predicting flight routes with a Deep Neural Network in the operational Air Traffic Flow and Capacity Management system*. Accessed: 2020-04-09. URL: https://www.eurocontrol.int/sites/default/files/2019-06/traffic-prediction-improvements-tpi-factsheet.pdf.

[23] Federal Aviation Administration. *Aeronautical Information Manual*. 2017.

[24] Federal Aviation Administration. *Instrument Flying Handbook*. Skyhorse Publishing Inc., 2012.

[25] Federal Aviation Administration. *JO 7210.3W - Facility Operation and Administration*. 2010. URL: https://www.faa.gov/regulations_policies/orders_notices/index.cfm/go/document.information/documentid/215049.

[26] Federal Aviation Administration. *The Future of the NAS*. US Department of Transportation, 2016.

[27] Federal Aviation Administration. *Traffic Flow Management in the National Airspace System*. 2009. URL: `https://www.fly.faa.gov/Products/Training/Traffic_Management_for_Pilots/TFM_in_the_NAS_Booklet_ca10.pdf`.

[28] Federal Aviation Administration. *Traffic Flow Management System (TFMS) Improving Demand Predictions (IDP)*. US Department of Transportation, 2015.

[29] Federal Aviation Administration and EUROCONTROL. *2013 Comparison of Air Traffic Management-Related Operational Performance: U.S./Europe*. European Commission, US Department of Transportation, 2014.

[30] Federal Aviation Administration and EUROCONTROL. *2015 Comparison of Air Traffic Management-Related Operational Performance: U.S./Europe*. European Commission, US Department of Transportation, 2016.

[31] Ian Fuller, Jean-Claude Hustache, and Tarja Kettunen. "Enhanced flight efficiency indicators". In: *Eurocontrol Experimental Centre, EEC/SEE/2004/011* (2004).

[32] Maxime Gariel, Ashok N Srivastava, and Eric Feron. "Trajectory clustering and an application to airspace monitoring". In: *IEEE Transactions on Intelligent Transportation Systems* 12.4 (2011), pp. 1511–1524.

[33] Brunilde Girardet et al. "Wind-optimal path planning: Application to aircraft trajectories". In: *2014 13th International Conference on Control Automation Robotics & Vision (ICARCV)*. IEEE. 2014, pp. 1403–1408.

[34] Ross Girshick. "Fast r-cnn". In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1440–1448.

[35] Ross Girshick et al. "Rich feature hierarchies for accurate object detection and semantic segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 580–587.

[36] Alex Graves and Navdeep Jaitly. "Towards end-to-end speech recognition with recurrent neural networks". In: *International conference on machine learning*. 2014, pp. 1764–1772.

[37] Mark Hansen et al. "En route performance in the national airspace system". In: *96th Transportation Research Board Annual Meeting*. 2017.

[38] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[39] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory". In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[40] International Civil Aviation Organization. *Global Air Navigation Plan Resources*. Accessed: 2020-04-13. URL: `http://www.icao.int/airnavigation/Pages/GANP-Resources.aspx`.

[41] Eugenia Kalnay et al. "The NCEP/NCAR 40-year reanalysis project". In: *Bulletin of the American meteorological Society* 77.3 (1996), pp. 437–472.

[42] Andrej Karpathy, Armand Joulin, and Li F Fei-Fei. “Deep fragment embeddings for bidirectional image sentence mapping”. In: *Advances in neural information processing systems*. 2014, pp. 1889–1897.

[43] Amy M Kim and Mark Hansen. “Collaborative Rerouting in the Airspace Flow Program”. In: (2019).

[44] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “Imagenet classification with deep convolutional neural networks”. In: *Advances in neural information processing systems*. 2012, pp. 1097–1105.

[45] Jae-Gil Lee, Jiawei Han, and Xiaolei Li. “Trajectory outlier detection: A partition-and-detect framework”. In: *2008 IEEE 24th International Conference on Data Engineering*. IEEE. 2008, pp. 140–149.

[46] Jae-Gil Lee, Jiawei Han, and Kyu-Young Whang. “Trajectory clustering: a partition-and-group framework”. In: *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*. 2007, pp. 593–604.

[47] Ziheng Lin et al. “Deep generative models of urban mobility”. In: *IEEE Transactions on Intelligent Transportation Systems* (2017).

[48] James MacQueen et al. “Some methods for classification and analysis of multivariate observations”. In: *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*. Vol. 1. 14. Oakland, CA, USA. 1967, pp. 281–297.

[49] David McNally et al. “Operational Evaluatioin of Dynamic Weather Routes at American Airlines”. In: (2013).

[50] David McNally et al. “Dynamic weather routes: two years of operational testing at American Airlines”. In: (2015).

[51] Nimrod Megiddo. “The weighted Euclidean 1-center problem”. In: *Mathematics of Operations Research* 8.4 (1983), pp. 498–504.

[52] Jere S Meserole and John W Moore. “What is system wide information management (SWIM)?” In: *2006 ieee/aiaa 25TH Digital Avionics Systems Conference*. IEEE. 2006, pp. 1–8.

[53] TA Meyers et al. “A preliminary analysis of the impact of miles-in-trail restrictions on NAS flight operations”. In: *6th US/Europe Air Traffic Management Seminar*. 2005.

[54] Tysen Mueller, John Sorensen, and George Couluris. “Strategic aircraft trajectory prediction uncertainty and statistical sector traffic load modeling”. In: *AIAA Guidance, Navigation, and Control Conference and Exhibit*. 2002, p. 4765.

[55] Mary O’Keeffe, Diarmuid Houlihan, and Nicholas Hunt. *AFRA Real Time Simulation Project Final Report*. European Commission, 2015.

[56] Paul Ostwald, Tejal Topiwala, and James DeArmon. “The miles-in-trail impact assessment capability”. In: *6th AIAA Aviation Technology, Integration and Operations Conference (ATIO)*. 2006, p. 7780.

[57] Kee Palopo et al. “Wind-optimal routing in the national airspace system”. In: *Journal of Aircraft* 47.5 (2010), pp. 1584–1592.

[58] Emanuel Parzen. “On estimation of a probability density function and mode”. In: *The annals of mathematical statistics* 33.3 (1962), pp. 1065–1076.

[59] Subramanian Ramasamy et al. “Next generation flight management system for real-time trajectory based operations”. In: *Applied Mechanics and Materials*. Vol. 629. Trans Tech Publ. 2014, pp. 344–349.

[60] Subramanian Ramasamy et al. “Novel flight management system for real-time 4-dimensional trajectory based operations”. In: *AIAA Guidance, Navigation, and Control (GNC) Conference*. 2013, p. 4763.

[61] Herbert E Rauch, F Tung, and Charlotte T Striebel. “Maximum likelihood estimates of linear dynamic systems”. In: *AIAA journal* 3.8 (1965), pp. 1445–1450.

[62] Shaoqing Ren et al. “Faster r-cnn: Towards real-time object detection with region proposal networks”. In: *Advances in neural information processing systems*. 2015, pp. 91–99.

[63] Tom Reynolds. “Analysis of lateral flight inefficiency in global air traffic management”. In: *The 26th Congress of ICAS and 8th AIAA ATIO*. 2008, p. 8865.

[64] Kapil Sheth, Sebastian Gutierrez-Nolasco, and Julien Petersen. “Analysis and modeling of miles-in-trail restrictions in the national airspace system”. In: *2013 Aviation Technology, Integration, and Operations Conference*. 2013, p. 4247.

[65] Karen Simonyan and Andrew Zisserman. “Very deep convolutional networks for large-scale image recognition”. In: *arXiv preprint arXiv:1409.1556* (2014).

[66] Banavar Sridhar et al. “Strategic planning of efficient oceanic flights”. In: *11th US/Europe Air Traffic Management Seminar*. 2015.

[67] Jonathan Standley et al. “SWIM segment 2 deployment and utilization in NextGen R&D programs”. In: *2012 Integrated Communications, Navigation and Surveillance Conference*. IEEE. 2012, G8–1.

[68] Bob Stephens. “System-wide information management (SWIM) demonstration security architecture”. In: *2006 ieee/aiaa 25TH Digital Avionics Systems Conference*. IEEE. 2006, pp. 1–12.

[69] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. “Sequence to sequence learning with neural networks”. In: *Advances in neural information processing systems*. 2014, pp. 3104–3112.

[70] Christian Szegedy et al. “Going deeper with convolutions”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 1–9.

[71] Ivan Tereshchenko and Mark Hansen. “Relative trajectory cost estimation for CTOP applications using multivariate nonparametric finite mixture logit”. In: *CEAS Aeronautical Journal* (2020), pp. 1–11.

[72] Ivan Tereshchenko et al. “Relative Trajectory Cost Prediction for Trajectory Options Set Generation in CTOP Simulations”. In: *2018 Aviation Technology, Integration, and Operations Conference*. 2018, p. 3041.

[73] Kenneth E Train. *Discrete choice methods with simulation*. Cambridge university press, 2009.

[74] US Department of Transportation. *Airline Service Quality Performance Reports*. 1987.

[75] Oriol Vinyals et al. “Show and tell: A neural image caption generator”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 3156–3164.

[76] Michail Vlachos, George Kollios, and Dimitrios Gunopulos. “Discovering similar multidimensional trajectories”. In: *Proceedings 18th international conference on data engineering*. IEEE. 2002, pp. 673–684.

Appendix A

List of US 34 Core Airports

City	ICAO Code	IATA Code	Country	Avg. Daily IFR Departures in 2013
Atlanta	KATL	ATL	USA	1238
Chicago	KORD	ORD	USA	1203
Dallas	KDFW	DFW	USA	928
Los Angeles	KLAX	LAX	USA	836
Denver	KDEN	DEN	USA	804
Charlotte	KCLT	CLT	USA	758
Houston	KIAH	IAH	USA	691
Phoenix	KPHX	PHX	USA	596
Philadelphia	KPHL	PHL	USA	592
Minneapolis	KMSP	MSP	USA	591
Detroit	KDTW	DTW	USA	582
San Francisco	KSFO	SFO	USA	572
Newark	KEWR	EWR	USA	564
New York	KJFK	JFK	USA	557
Las Vegas	KLAS	LAS	USA	554
Miami	KMIA	MIA	USA	540
New York	KLGA	LGA	USA	508
Boston	KBOS	BOS	USA	495
Washington	KIAD	IAD	USA	451
Seattle	KSEA	SEA	USA	430
Orlando	KMCO	MCO	USA	406
Washington	KDCA	DCA	USA	398
Salt Lake City	KSLC	SLC	USA	392
Baltimore	KBWI	BWI	USA	344
Chicago	KMDW	MDW	USA	332
Ft. Lauderdale	KFLL	FLL	USA	332
Memphis	KMEM	MEM	USA	316
Portland	KPDX	PDX	USA	279
Houston	IHOU	HOU	USA	258
St. Louis	KSTL	STL	USA	256
San Diego	KSAN	SAN	USA	255
Cleveland	KCLE	CLE	USA	248
Tampa	KTPA	TPA	USA	244
Raleigh-Durham	KRDU	RDU	USA	230

Appendix B

List of European 34 Core Airports

City	ICAO Code	IATA Code	Country	Avg. Daily IFR Departures in 2013
Paris	LFPG	CDG	FRANCE	655
Frankfurt	EDDF	FRA	GERMANY	648
London	EGLL	LHR	UNITED KINGDOM	646
Amsterdam	EHAM	AMS	NETHERLANDS	597
Munich	EDDM	MUC	GERMANY	519
Madrid	LEMD	MAD	SPAIN	456
Rome	LIRF	FCO	ITALY	414
Barcelona	LEBL	BCN	SPAIN	379
Zurich	LSZH	ZRH	SWITZERLAND	350
London	EGKK	LGW	UNITED KINGDOM	343
Vienna	LOWW	VIE	AUSTRIA	339
Copenhagen	EKCH	CPH	DENMARK	336
Oslo	ENGM	OSL	NORWAY	330
Paris	LFPO	ORY	FRANCE	320
Stockholm	ESSA	ARN	SWEDEN	301
Brussels	EBBR	BRU	BELGIUM	289
Dusseldorf	EDDL	DUS	GERMANY	288
Geneva	LSGG	GVA	SWITZERLAND	243
Berlin	EDDT	TXL	GERMANY	237
Palma	LEPA	PMI	SPAIN	232
Dublin	EIDW	DUB	IRELAND	232
Manchester	EGCC	MAN	UNITED KINGDOM	231
Helsinki	EFHK	HEL	FINLAND	230
Milan	LIMC	MXP	ITALY	226
Lisbon	LPPT	LIS	PORTUGAL	200
London	EGSS	STN	UNITED KINGDOM	196
Nice	LFMN	NCE	FRANCE	192
Hamburg	EDDH	HAM	GERMANY	187
Athens	LGAV	ATH	GREECE	186
Prague	LKPR	PRG	CZECH REPUBLIC	171
Cologne	EDDK	CGN	GERMANY	160
Lyon	LFLL	LYS	FRANCE	159
Stuttgart	EDDS	STR	GERMANY	156
Milan	LIML	LIN	ITALY	153