Online Stochastic Prediction of Mid-Flight Aircraft Trajectories

Yongzhen Arthur Pan
University of Alberta
arthurpan@ualberta.ca

Mario A. Nascimento
University of Alberta
mario.nascimento@ualberta.ca

Joerg Sander
University of Alberta
jsander@ualberta.ca

ABSTRACT

Online trajectory prediction is central to the function of air traffic control of improving the flow of air traffic and preventing collisions, particularly considering the ever-increasing number of air travellers. In this paper, we propose an approach to predict the mid-flight trajectory of an aircraft using models learned from historical trajectories. The main idea is based on Hidden Markov Models, representing the location of aircraft as states and weather conditions as observations. Using our approach, one is able to make predictions of future positions of currently mid-flight aircraft for each minute into the future, optionally concatenating these positions to form the remaining predicted trajectory of an aircraft. We evaluated the effectiveness of the proposed approach using a dataset of historical trajectories for flights over the USA. Using prediction accuracy metrics from the aviation domain, we demonstrated that our approach could accurately predict trajectories of mid-flight aircraft, achieving an effectiveness improvement of 26% in horizontal error and 32% in vertical error over baseline models with virtually no loss in prediction efficiency.

CCS CONCEPTS

• Information systems → Online analytical processing; **• Computing methodologies → Machine learning approaches**; **• Applied computing → Aerospace**.

KEYWORDS

Air Traffic Management, Aircraft Trajectory Prediction, Markov Models, Time Series

1 INTRODUCTION

The International Air Transport Association (IATA) predicts that the number of air travellers will rise to 7.2 billion passengers in 2035, nearly double the recorded 3.8 billion passengers in 2016 [8]. The immediate challenge arising from such a tremendous growth is the need to put in place better infrastructure — such as runways, terminals, security screening systems and baggage handling systems — with higher capacities to cope with the demand. A less obvious, from the passenger perspective, but extremely important component in such a challenge is to ensure the safety of all aircraft flying within unavoidably congested airspace, and a fundamental step in achieving this goal is the adoption of more accurate Trajectory Prediction (TP) systems. Trajectory Prediction systems are used by Air Traffic Control (ATC) to calculate the likely flight paths of aircrafts currently flying in the airspace in order to anticipate potential bottlenecks in the flow of air traffic or potential breaches in the safe separation of aircraft.

In providing accurate predictions of aircraft trajectories, our proposal can aid air traffic controllers in their decision-making when directing a high volume of air traffic, thereby reducing the chances for errors due to a heavy workload or fatigue. It can also be used by downstream ATC applications to pin-point areas within the airspace that may become congested, or for automatically detecting conflicts in the predicted trajectories that may result in mid-air collisions [1].

Traditional TP systems perform calculations based on the current state and intent of an aircraft — such as position, velocity and acceleration parameters and flight plan — to predict its likely trajectory over the next ten to twenty minutes [7]. The prediction accuracy of this approach can be limited due to unexpected changes in aircraft intent (such as a temporary change in altitude to maintain safe separation from another aircraft) or due to unknown aircraft specifications, as well as environmental factors such as strong winds or airspace congestion [7]. However, recent advancements in computing research have made it possible to improve the accuracy of TP systems through the learning of probabilistic models from historical data. Indeed, we explore this avenue by building Markov Models (MMs) from historical trajectory data to be used for the prediction of aircraft trajectories, with the aim of application in *real-time* air traffic management.

In order to use Hidden MMs (HMMs) for our problem, we need to map aircraft positions and weather observations into discrete states of the model. First, we divide the geographical volume of interest into discrete, evenly spaced grid cubes. Each hidden state represents the presence of the aircraft of interest in a corresponding grid cube, and each observation produced by a hidden state represents the aircraft being subjected to a particular weather condition in the corresponding grid cube. Then, we train models from historical trajectories, based on look-ahead times from 1 minute to 20 minutes. The current position of the aircraft is used as the input start state for each model to predict the most likely position of the aircraft after its corresponding look-ahead time, and the concatenation of the predicted positions of these models form the predicted trajectory of the aircraft, in the air, for the next 20 minutes.

The contributions of this paper can be summarized as follows:

- We mapped basic concepts for the Trajectory Prediction problem onto Hidden Markov Models that is able to incorporate local weather information, and used multiple models for the online stochastic prediction of mid-flight aircraft trajectories.
- We used a large number of historical trajectories over a continuous time span of 2 years for an experimental evaluation, consisting of flights between LaGuardia Airport and Chicago O'Hare Airport (which is among the top 20 busiest domestic

IWCTS'19, November 5, 2019, Chicago, IL, USA

ACM ISBN 978-1-4503-6967-1/19/11…$15.00
https://doi.org/10.1145/3357000.3366144

routes in 2017 [11]), as well as flights between Boston Logan Airport and Chicago O'Hare Airport.
- We applied relevant prediction accuracy metrics from the aviation domain [7] for a comprehensive experimental evaluation. Our proposed model achieved highly accurate predictions of mid-flight aircraft trajectories, with an improvement of 26% in horizontal error and 32% in vertical error compared to two baseline models based on conventional approaches, while not requiring more prediction time.

The rest of this paper is organized as follows. Section 2 gives an overview of related work. Section 3 provides some basic concept definitions and the problem definition, followed by the description of our proposed model and workflow in Section 4. Section 5 presents the experimental evaluation of our proposed model. Section 6 concludes the paper with a brief summary and discussion of future work.

2 RELATED WORK

The collection and analysis of data generated by moving objects, such as people, vehicles and animals, has increased dramatically in recent years due to the rapid proliferation of GPS-enabled devices. In particular, trajectory data, representing the path travelled by moving objects, contains important information that can provide insight into the travel patterns and intent of a moving object. Trajectory data can be broadly classified into two categories: objects moving within the constraints of a road network, such as cars and trains, and objects moving in a free and open space, such as aircraft, ships and animals.

The first category, trajectory data of road vehicles, has been widely explored in recent years because of the widespread adoption of map navigation applications such as Google Maps. In this category, the road network is represented as a graph in which road segments correspond to edges and intersections correspond to nodes. The trajectories of vehicles are then mapped to edges and nodes in order to identify, *e.g.*, travel patterns, traffic conditions and anomalies [22].

The second category, trajectory data of objects moving in open spaces, can also make use of a graph approach when the semantics of certain areas or locations in the open space can be exploited, such as through making use of certain waypoints or key locations [22]. However, there is less semantics available for the analysis of aircraft trajectory data in particular, because trajectories do not always strictly follow well-defined paths, and changing environmental conditions may require changes to the prepared flight plan. Recent literature has followed a clustering approach to identifying patterns in flight trajectories, which includes the identification and clustering of waypoints [6], clustering of whole trajectories [6], and clustering of sub-trajectories [2]. They usually follow a two-step approach of first finding a cluster for a trajectory (or sub-trajectory) of a moving object, then using a representative of the cluster as a prediction of future trajectories of the object. This approach is often unable to make accurate predictions since well-separation of clusters is not guaranteed, and trajectories hardly fall squarely within a single cluster.

Recently, Ayhan & Samet [3] presented a stochastic approach for pre-flight trajectory prediction using Hidden Markov Models (HMMs), where hidden states represent spatial-temporal data cubes traversed by a trajectory and observations produced by the hidden states represent weather conditions within the data cubes. An input observation sequence is generated for the HMM decoding process in order to find the most probable hidden state sequence, which represents the most probable trajectory for a particular flight. All weather observation sequences pertaining to state sequences traversed by historical trajectories are grouped together, and the centroid observations for each time instance are taken to form the input observation sequence. The Viterbi algorithm is then used to generate the most likely sequence of spatio-temporal data cubes representing the predicted trajectory of the flight for the day.

Our work is different from Ayhan & Samet [3] in the following ways:
- We focus on the online prediction of trajectories of aircraft that are currently in the air, to aid air traffic controllers in directing aircraft to navigate the airspace safely throughout the whole duration of the flight. In Ayhan & Samet [3], the goal was to make a *pre*-flight prediction of the whole trajectory of the aircraft, assuming the median flight duration of all considered flights. We instead use the current *mid*-flight position of the aircraft as input to our Markov Model, *i.e.* when it is in the air, and make a prediction of the future trajectory of the aircraft for a look-ahead time of 1 to 20 minutes.
- Unlike Ayhan & Samet [3], we do not use fixed intervals to globally bin weather parameters for the HMM, but rather derive local intervals from the actual values of weather parameters observed in each grid cube over the time period of the training data. This is to avoid the situation where weather parameters in a particular grid cube fall within the same bin most of the time, as the probability distribution of weather parameters are highly specific to each grid cube.

3 BASIC DEFINITIONS

In this section, we define some basic concepts that are used for the rest of this paper, before introducing our problem definition. Because of the shared application domain, our definitions contain some similarities with those in Ayhan & Samet [3].

3.1 Geospatial Concepts

The World Geodetic System 84 (WGS84) defines a standard coordinate system for the Earth, serving as a reference for the Global Positioning System (GPS), a global navigation satellite system that provides accurate positional and time information to people or objects equipped with GPS receivers anywhere near the surface of the Earth.

Definition 3.1. Based on the WGS84 standard, a **geographical position** gp near the surface of the Earth can be represented using 3 coordinates,

$$gp = (latitude, longitude, altitude).$$

Definition 3.2. A **positional update** pos, provided by a GPS receiver, represents the geographical position gp of a particular aircraft at a particular timestamp ts,

$$pos = (gp, ts).$$

Definition 3.3. The *continuous* sequence of positions that form the exact path of travel of an aircraft can be called an original trajectory. As the original trajectory can be infinitely long and immeasurable, we use an approximate representation of it, called a **raw trajectory**. A raw trajectory Trj_{ac} is a finite sequence of positional updates of a particular aircraft ac, captured over a sequence of timestamps in increasing order,

$$Trj_{ac} = (pos_1, \ldots, pos_n), n \in \mathbb{Z}^+, n \geq 2,$$

where each position is

$$pos_i = (gp_i, ts_i), \forall i \in [1, n].$$

3.2 Reference Grid Concepts

There is an infinite number of possible representations of geographical positions over the continuous geographical coordinates latitude, longitude and altitude. In order to simplify the space of possible representations to a discrete and finite set, we perform a mapping of geographical positions to standard reference points in a reference grid. Further to this, to enable the use of weather data in our work, we use the 2-dimensional reference grid of the Rapid Refresh (RAP) weather forecasting system by the National Oceanic and Atmospheric Administration (NOAA) [4]. Under this mapping, geographical positions over the whole of continental United States (CONUS) are first projected to local coordinates using the Lambert Conformal Conic Projection [19], before being mapped to a 2-dimensional reference point in the planar reference grid. For the vertical dimension, RAP uses 50 vertical levels following a sigma coordinate system [4], which is problematic for our purposes as the vertical levels are not uniformly separated and the values for all 50 levels vary depending on the current surface pressure at the reference point in the grid. Therefore, we use the standard in aviation [5] to divide the vertical dimension into uniformly spaced intervals based on pressure altitude[1]. Weather parameters for each altitude interval follow that of the RAP vertical level closest to the center of the interval.

Our 3-dimensional reference grid is consequently a 3-dimensional array of geographical positions,

$$rp = (rp\_latitude, rp\_longitude, rp\_altitude),$$

spaced 13.545 km apart in a 451×337 configuration in the horizontal plane and spaced 2000 ft apart over 22 levels in the vertical dimension. The geographical volume of interest has thus been divided into discrete, evenly spaced volumes. Each volume, called a grid cube gc, is represented by a corresponding reference point rp at its center.

Given the reference grid, a raw trajectory Trj can be transformed into an aligned trajectory $\overline{Trj}$, by replacing the geographical position of each positional update in the raw trajectory with the reference point closest to it.

In what follows, we refer to the reference point rp representing a grid cube gc as $rp = ref(gc)$. We also refer to the set of all grid cubes as G.

[1]Pressure altitude, though expressed in feet, is a measurement of atmospheric pressure at a certain height, and conversion of units between millibars and feet can be done using a formula published by NOAA [15].

3.3 Weather Concepts

RAP is an hourly updated weather forecasting system that provides a location-specific forecast of hourly weather conditions up to the next 19 hours[2] [4]. Due to the accumulation of errors in numerically computing the interactions of environmental variables over time, the accuracy of weather forecasting decreases as the forecast hour into the future increases [21]. As the aim of our work is to examine the effect of current weather conditions on the trajectory of an aircraft, we will be using only the forecast of weather conditions for the current hour.

Definition 3.4. A **weather vector** $wv_{dy,hr,gc}$, describing the weather condition in a particular grid cube gc at a particular hour hr of a day[3] dy, consists of 4 weather parameters, namely specific humidity $sh_{dy,hr,gc}$, temperature (Kelvins) $tk_{dy,hr,gc}$, wind speed $ws_{dy,hr,gc}$ and wind direction $wd_{dy,hr,gc}$,

$$wv_{dy,hr,gc} = \begin{bmatrix} sh_{dy,hr,gc} \\ tk_{dy,hr,gc} \\ ws_{dy,hr,gc} \\ wd_{dy,hr,gc} \end{bmatrix}.$$

Given a timestamp ts, we also write $wv_{ts,gc}$ as a shortcut for $wv_{date(ts),hour(ts),gc}$.

3.4 Weather Binning Concepts

We adopt a binning approach for discretizing weather vectors, such that each continuous weather vector wv can be mapped to a corresponding vector of discrete bin values wb to be represented in a Markov Model. Because of the varying distributions of the weather parameters over different grid cubes, each grid cube requires a different set of bin intervals for each of the weather parameters. The wind direction parameter has a circular distribution, and is best divided using the 4 cardinal directions (north, south, east, west), giving a total of 4 bin intervals of 90 degrees each, as shown in Table 1. As for the parameters specific humidity, temperature, and wind speed, we obtain mean and standard deviation values for each parameter in each grid cube $gc \in G$, which we then use to determine 3 bin intervals, as shown in Table 2.

In what follows, we refer to the bin vector wb mapped from a weather vector wv as $wb = bin(wv)$. We also refer to the set of all possible bin vectors as W.

Table 1: Bin intervals for wind direction (wd)

Bin intervals for wd (in degrees)	
Bin no.	Interval
1	$315(=-45) < wd < 45$
2	$45 < wd < 135$
3	$135 < wd < 225$
4	$225 < wd < 315(=-45)$

[2]For the current hour and 18 hours into the future.

[3]We use the terms **day** and **date** interchangeably.

Table 2: Bin intervals for specific humidity (sh), temperature (tk) and wind speed (ws) in each grid cube $gc \in G$

Bin intervals for sh (same calculations apply for tk and ws)	
Bin no.	Interval
1	$(\mu_{sh,gc} - \sigma_{sh,gc}) < sh < (\mu_{sh,gc} + \sigma_{sh,gc})$
2	$sh > (\mu_{sh,gc} + \sigma_{sh,gc})$
3	$sh < (\mu_{sh,gc} - \sigma_{sh,gc})$

3.5 Problem Definition

Given a set of D historical trajectories of flights between two airports belonging to a particular airline, as well as the trajectory of a flight belonging to the airline on the current day since its departure timestamp ts_{dep} until the current timestamp ts_{now}, **our goal is to find the most probable trajectory of the flight for the next L minutes**, where the time L is called time from prediction, or look-ahead time[4]. Also, we note that our approach is designed for *mid*-flight predictions, which are bound to be more useful in ATC applications than *pre*-flight predictions.

4 PREDICTION MODELS

In this section, we introduce the standard components of an HMM before describing our mapping of concepts of the TP problem to an HMM and our approach for estimating probabilities in the HMM. We also present a workflow for building our proposed model for the TP problem.

4.1 Hidden Markov Models (HMMs)

An HMM is specified by the following components:

- A set of N **possible states**, $S = \{s_1, \ldots, s_N\}$.
- A **transition probability matrix**,
$$A = \begin{bmatrix} a(s_1, s_1) & \ldots & a(s_1, s_N) \\ \vdots & \ddots & \vdots \\ a(s_N, s_1) & \ldots & a(s_N, s_N) \end{bmatrix},$$
where each value $a(s_i, s_j)$ represents the probability of state s_i transitioning to state s_j, such that $\sum_{s_j} a(s_i, s_j) = 1, \forall s_i \in S$.
- A set of M **possible observations**, $U = \{u_1, \ldots, u_M\}$.
- An **emission probability matrix**,
$$B = \begin{bmatrix} b(s_1, u_1) & \ldots & b(s_1, u_M) \\ \vdots & \ddots & \vdots \\ b(s_N, u_1) & \ldots & b(s_N, u_M) \end{bmatrix},$$
where each value $b(s_j, u_k)$ represents the probability of state s_j producing observation u_k, such that $\sum_{u_k} b(s_j, u_k) = 1, \forall s_j \in S$.
- Special **start and end states**, s_0 and s_E, respectively, that are not associated with observations, with transition probabilities out of the start state $a(s_0, s_1), \ldots, a(s_0, s_N)$, and transition probabilities into the end state $a(s_1, s_E), \ldots, a(s_N, s_E)$.

HMMs have to address three fundamental problems: Likelihood Computation, Decoding, and Learning [10]. For these problems, the input to the model is an **observation sequence** $O = (o_1, \ldots, o_T)$, *i.e.* a sample sequence of T observations, such that $o_t \in U, \forall t \in [1, T]$.

In the HMM Decoding Problem, the Viterbi algorithm [10] is used to find the **most probable hidden state sequence** for a given input observation sequence $\widehat{Q} = (\widehat{q}_1, \ldots, \widehat{q}_T)$, *i.e.* the most probable sequence of T hidden states based on an input observation sequence O, such that $\widehat{q}_t \in S, \forall t \in [1, T]$.

4.2 Mapping the TP Problem to an HMM

We can model a particular flight route by probabilities of state changes in a Markov Model, based on historical trajectories of the flight route across the geographical space. We can then predict the most probable trajectory of the flight by finding the most probable state sequence, using the Markov Model, through the following mapping:

- The set of possible states S corresponds to the set of grid cubes G, each of which is represented by a reference point. Since there are $451 \times 337 \times 22 = 3,343,714$ reference points, we have a total of $N = 3,343,714$ possible states. A state $s_i \in S$ is said to be true for a particular aircraft at a particular time, when the aircraft is in the corresponding grid cube $gc_i \in G$ at that time.
- The transition probability $a(s_i, s_j)$ represents the probability of the aircraft moving from grid cube gc_i, corresponding to s_i, to grid cube gc_j, corresponding to s_j.

By extending the model to a Hidden Markov Model consisting of states which are partially observable through observations, we can incorporate the influence of weather conditions on the trajectory of the aircraft by treating weather vectors as observations produced as the aircraft transitions through the hidden states:

- The set of possible observations U corresponds to the set of bin vectors W, each representing an interval of possible weather vectors. Since there are $4 \times 3 \times 3 \times 3 = 108$ bins, we have a total of $M = 108$ possible observations. An observation $u_k \in U$ is said to be true for a particular aircraft in a particular state s_j at a particular time, when the aircraft is in the corresponding grid cube gc_j at that time, and is experiencing the weather condition represented by wv that falls into the corresponding bin vector $bin(wv) = wb_k \in W$.
- The emission probability $b(s_j, u_k)$ represents the probability of an aircraft experiencing a weather condition that falls into bin vector wb_k, corresponding to u_k, when the aircraft is in grid cube gc_j, corresponding to s_j.

One inherent limitation of HMMs is that there is no explicit representation of the time duration in each state [9]. In a first approach, we used a time duration of 1 minute for each learned state transition probability, and then applied the Viterbi algorithm to find the most likely state sequence after L transitions to predict a trajectory L minutes into the future. With this approach, however, we observed that the resulting predictions for larger values of L (e.g. $L = 15$), often resulted in large errors: over-estimations along the climbing and descending phases and under-estimations along the cruising phase of flights, because of the difference in speeds during each phase.

To deal with this difficulty in accurate time-keeping over a large number of state transitions in the model, we propose an alternative

[4]Most conflict detection applications are interested in a look-ahead time of ten to twenty minutes [7].

approach. Instead of a single model based on 1 minute transition probabilities, where we need L state transitions to predict a trajectory L minutes into the future, we train L models, $\mathcal{M}_1, \ldots, \mathcal{M}_L$. Model $\mathcal{M}_t$, $t = 1, \ldots, L$, is trained to capture the transition probabilities $a(s_i, s_j)$ that an aircraft moves from grid cube gc_i, corresponding to s_i, to grid cube gc_j, corresponding to s_j, after t minutes. We can then use these models to make a prediction for each of the $1, \ldots, L$ minutes into the future based a single state transition. Because of this, our approach is not only able to make more accurate predictions, but also requires a much shorter running time as compared to running the Viterbi algorithm over L state transitions.

We also need to determine the start and end states in the HMM for the TP problem. Based on the problem definition, predictions shall be made anytime during the flight on the current day, with the implication that the start state should represent the current location of the aircraft. Because of this, the start state in our model is not a special state, but rather, a state belonging to the set of possible states, $s_i \in S, i \in [1, N]$. We write $q_{ts_{now}} = s_i$, to indicate and "select" the start state as the state at the current timestamp ts_{now}. Transition probabilities out of the start state are those found in the transition probability matrix A for state s_i, *i.e.* $a(s_i, s_1), \ldots, a(s_i, s_N)$. The end state in our model is a special state s_E that is not in the set of possible states S. Since there is no reason to prefer any particular state to be the last state in the predicted hidden state sequence (*i.e.* the state before the special end state s_E), we omit transition probabilities into the end state, and also the end state itself, from our model.

In summary, assuming we have a transition probability matrix and an emission probability matrix (described in the next subsection), we have the following input and output to our model:

- Input: start state $q_{ts_{now}}$ and set of observations $W_{ts_{now}}$. Instead of taking an observation sequence, our model takes as input the start state $q_{ts_{now}} \in S$ at the current timestamp ts_{now}, its corresponding grid cube $gc_{ts_{now}}$, as well as the set of weather conditions observed in all grid cubes for the following L minutes, $W_{ts_{now}} = \{bin(wv_{ts,gc}) \in W \mid gc \in G, ts \in [ts_{now} + 1, ts_{now} + L]\}$, where we assume timestamps to be whole minutes.
- Output: most probable hidden states $\widehat{q}_{ts_{now}+1}, \ldots, \widehat{q}_{ts_{now}+L}$. We use each of the L models to find the most probable next state for each look-ahead time of t minutes, $t = 1, \ldots, L$. The concatenation of these states in ascending order of look-ahead time forms a sequence of most probable hidden states, $\widehat{Q} = (\widehat{q}_{ts_{now}+1}, \ldots, \widehat{q}_{ts_{now}+L})$, which can be mapped into a predicted trajectory for the next L minutes, by assigning as positions the reference points of grid cubes corresponding to the predicted states, *i.e.* $\widehat{Trj} = (\widehat{pos}_{ts_{now}+1}, \ldots, \widehat{pos}_{ts_{now}+L}) = (ref(\widehat{gc}_{ts_{now}+1}), \ldots, ref(\widehat{gc}_{ts_{now}+L}))$.

4.3 Estimating Probabilities for the HMM

Our estimation of transition probabilities follows the usual approach of counting occurrences, using a training dataset consisting of D aligned historical trajectories $\{\overline{Trj}_1, \ldots, \overline{Trj}_D\}$.

For an HMM with look-ahead time of t minutes, $t \in [1, L]$, $a(s_i, s_j)$ represents the probability of state $s_i \in S$ transitioning to state $s_j \in S$, and can be estimated by counting, how many times aircrafts on historical flights were in grid cube gc_j, corresponding to state s_j, when they were in grid cube gc_i, corresponding to state s_j, t minutes earlier, *i.e.* ,

$$a(s_i, s_j) = P((ref(gc_j), ts) \in \overline{Trj} \mid (ref(gc_i), ts - t) \in \overline{Trj}) = \frac{|\{\overline{Trj} \mid (ref(gc_j), ts) \in \overline{Trj} \wedge (ref(gc_i), ts - t) \in \overline{Trj}\}|}{|\{\overline{Trj} \mid (ref(gc_i), ts - t) \in \overline{Trj}\}|}, \quad (1)$$

where $(ref(gc_j), ts) \in \overline{Trj}$ denotes that an aligned trajectory $\overline{Trj}$ contains a position $ref(gc_i)$ (*i.e.* the reference point of grid cube gc_i, corresponding to state s_i) with timestamp ts.

As for the estimation of emission probabilities, we follow a similar approach. However, given a total of $M = 108$ bins where each weather vector can fall into, a huge amount of data would be required to estimate the emission probabilities without obtaining zero probability values. To mitigate this situation, we perform Laplace Smoothing in our estimation by adding a pseudo-count of 1 for each bin ("Add-One Smoothing") [12].

$b(s_j, u_k)$ represents the probability of state $s_j \in S$ producing observation $u_k \in U$, and can be estimated by counting (with Add-One Smoothing), how many times aircrafts on historical flights experienced a weather condition that falls into bin vector wb_k, corresponding to observation u_k, when they were in grid cube gc_j, corresponding to state s_j, *i.e.* ,

$$b(s_j, u_k) = P(bin(wv_{ts,gc_j}) = wb_k \mid (ref(gc_j), ts) \in \overline{Trj}) \approx \frac{|\{\overline{Trj} \mid bin(wv_{ts,gc_j}) = wb_k \wedge (ref(gc_j), ts) \in \overline{Trj}\}| + 1}{|\{\overline{Trj} \mid (ref(gc_j), ts) \in \overline{Trj}\}| + 108}, \quad (2)$$

where $(ref(gc_j), ts) \in \overline{Trj}$ denotes that an aligned trajectory $\overline{Trj}$ contains a position $ref(gc_i)$ (*i.e.* the reference point of grid cube gc_i, corresponding to state s_i) with timestamp ts, and $bin(wv_{ts,gc_j}) = wb_k$ denotes that the weather condition wv_{ts,gc_j} in grid cube gc_j at timestamp ts falls into the bin vector wb_k.

Given a state $q_{ts_{now}}$, the most probable next state with look-ahead time t, $\widehat{q}_{ts_{now}+t}$, is the one that maximizes the product of the transition and emission probabilities, *i.e.* ,

$$\widehat{q}_{ts_{now}+t} = \underset{s_j \in S}{\operatorname{argmax}} \left(a(q_{ts_{now}}, s_j) \times b(s_j, u_k)\right), \quad (3)$$

where observation u_k corresponds to the bin vector wb_k of the weather vector $wv_{ts_{now}+t,gc_j}$, which represents the weather condition at timestamp $ts_{now} + t$ in grid cube gc_j, corresponding to state s_j.

4.4 Proposed Workflow

The proposed workflow for building our Trajectory Prediction models includes data pre-processing, data discretization, and model training, as shown in Figure 1.

The use of weather data by our proposed HMM approach involves loading and computing a large amount of weather data computed hourly by the RAP system [4, 17] over the continental US for a period of 2 years. Through pre-binning of weather data, we can speed up the training and evaluation of models such that each weather vector is binned only once, and can be used by multiple models being evaluated on the same dataset. The size of the data

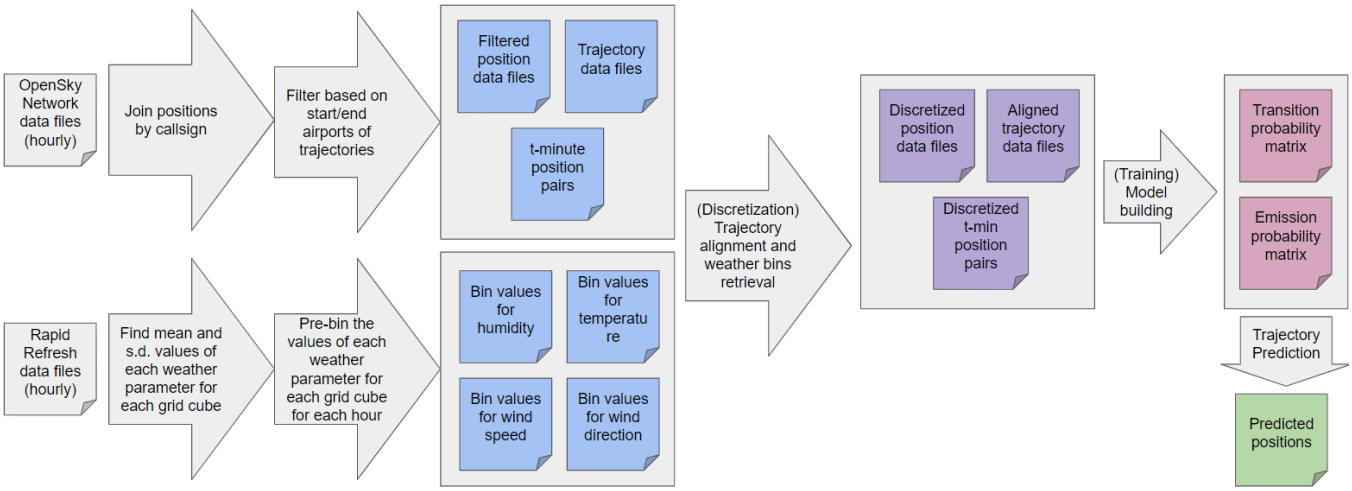

Figure 1: Overview of our proposed workflow

that is loaded into memory during model training is also reduced, from about 26–30 MB (compressed) to 3 MB to store all bin vectors of a particular hour. To do so, we first determine the mean and standard deviation values of each weather parameter for each grid cube over the whole training set. We then use these values to bin the weather vectors for each hour of the whole 2 years' weather dataset.

Aircraft movement data is collected and stored by the OpenSky Network [14, 18] in the form of positional updates, which we download in hourly portions for a period of 2 years. We join positional updates in chronological order to form trajectories based on their callsign. As positions along a trajectory may be unevenly spaced due to poor sensor coverage, we discard trajectories that have 'gaps' in positional updates of longer than 5 minutes. Also, since positional updates along a trajectory can be within a few seconds apart from each other, we form a raw trajectory by keeping only positional updates that are spaced 1 minute apart. We also check the start and end positions of each raw trajectory, keeping only those that start and end at the departure and arrival airports of the flight routes that we are interested in. Lastly, based on look-ahead times $t = 1, \ldots, L$, for each position on the raw trajectory, we select a position that occurs t minutes after, allowing for a difference of no more than 15 seconds. This gives us pairs of positions that can be used for counting the transition probability matrix during model training.

For data discretization, we map each positional update in a raw trajectory to the reference point nearest to it, giving us an aligned trajectory containing discretized positions (*i.e.* reference points with timestamps). For each discretized position, we also load its weather bin vector using its corresponding grid cube and timestamp. This gives us the bin vector for each position that can be used for counting the emission probability matrix during model training.

Training the model involves counting of occurrences in discretized t minute position pairs for estimating the transition and emission probability matrices, based on Equations 1 & 2. With the matrices, trajectory prediction for a particular t minute test position pair is done by using the first position as input for finding the most probable next state, based on Equation 3. The predicted position can then be compared with the second position in the position pair for measuring of prediction error, as described in the next section on experimental evaluation.

5 EXPERIMENTAL EVALUATION

In this section, we present a description of the data and models used for comparison, followed by the accuracy metrics that we adopt and an analysis of the experimental results we obtained.

5.1 Data Description

The weather data used for our work comes from the Rapid Refresh (RAP) weather forecasting system developed by the National Oceanic and Atmospheric Administration (NOAA) [4, 17]. The website for the RAP system provides a link to an archive of data generated since May 01, 2012. A single file, recording the weather conditions over the whole of the continental United States is archived for each hour of a day. The trajectory data used for our work is based on the Automatic Dependent Surveillance Broadcast (ADS-B) technology, and comes from historical aircraft trajectory data archived by the OpenSky Network [14, 18].

We downloaded trajectory and weather data over a total of 25 months, from 2017-Jan to 2019-Jan (inclusive). The size of the trajectory data collected for each flight route, together with median flight times, is shown in Table 3, while the break down of the number of trajectories by month is shown in Figure 2. The small number (≤ 50) of trajectories in certain months, especially for the first few months of the dataset, can be attributed to poor sensor coverage, resulting in trajectories having large 'gaps' being dropped during the pre-processing phase. We present most of our comparisons based on the busiest flight route $R1$, as it has the largest number of trajectories.

Finally, all experiments were run on a computer with Intel Core i7-6700HQ 2.6GHz CPU and 8 GB DDR4 Memory, using the Java software libraries GeoTools [20] for geospatial calculations and NetCDF [13] for reading RAP weather data.

Table 3: Meta-data about the flight routes used in the experiments (LGA, ORD, and BOS are the IATA codes for LaGuardia Airport, Chicago O'Hare Airport, and Boston Logan airports, respectively)

Route code	Origin / Destination	Mean flight time (minutes)	Number of trajectories	Number of positions
R1	LGA-ORD	116	8036	917299
R2	ORD-LGA	97	4020	389777
R3	BOS-ORD	133	2876	379938
R4	ORD-BOS	108	2049	221820

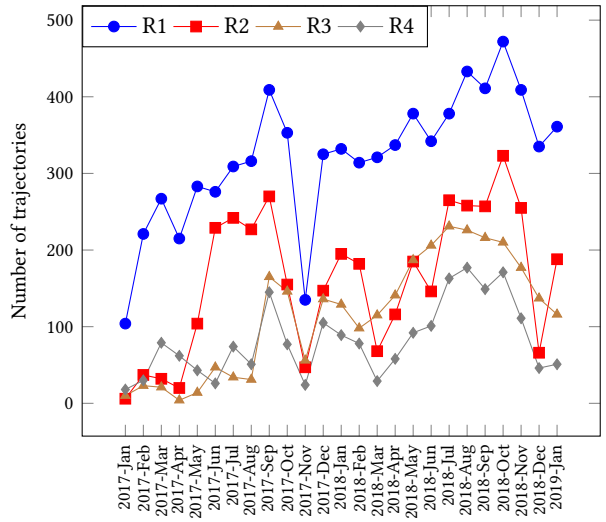

Figure 2: Number of trajectories for each month

5.2 Baseline Models

We implement two baseline models for comparison. The first is called the Kinematics-Based Model (KBM), which predicts the trajectory of the aircraft based on the current track angle, ground speed and vertical rate of the aircraft. Using GeoTools [20], we make a prediction by projecting the current position of the aircraft t minutes into the future based on these kinematic parameters.

We also implement a Median Trajectory Model (MTM) inspired by Ayhan & Samet [3] — though our work is not directly comparable because of the different intended application (*mid*-flight vs. *pre*-flight) and prediction time horizon (up to 20 minutes vs. time since departure). For each trajectory in the training set of D historical trajectories, we first calculate the sum of its Dynamic Time Warping (DTW) distances to all other trajectories. We then select the trajectory that has the smallest sum of DTW distances as the median trajectory. To make a prediction of the trajectory of a flight based on its current position, we find the positional update on the median trajectory that is nearest to the current position. Based on a look-ahead time of t minutes, we then select the positional update on the median trajectory that is t minutes after, as the position predicted by the MTM model.

5.3 Model Training

For choosing the train-test split of the data, we use an approach that captures possible seasonal differences in weather patterns. For each month in the second year (*e.g.* 2018-Jun), we train a prediction model using the same month in the previous year and its immediate neighbouring months (*e.g.* 2017-May, 2017-Jun, & 2017-Jul). Thus, we start our testing from the month of 2018-Feb (requiring 2017-Jan, 2017-Feb, & 2017-Mar for training) instead of 2018-Jan, and finish our testing on the final month of 2019-Jan (requiring 2017-Dec, 2018-Jan, & 2018-Feb for training) instead of 2018-Dec, spanning a total of 25 months of data.

We present all the discussed models in Table 4. We note that the HMM approach is unable to make predictions for some positions in the test sets. This happens in cases when a test position falls into a grid cube that no position from a trajectory in the training set had fallen into. In order to have a fair comparison, we cannot simply ignore these occurrences; we thus fall back on the KBM model to make predictions in these cases. Such occurrences, however, represent less than 5% of the test data and do not have a significant effect on the overall prediction accuracy of the HMM.

Table 4: Prediction models used for comparison

Model code	Name of model	Training approach
KBM	Kinematics-Based Model	Training data not required
MTM	Median Trajectory Model	3 months in previous year
HMM	Hidden Markov Model	3 months in previous year

5.4 Accuracy Metrics

We use the metrics presented by Paglione & Oaks [16] to evaluate the accuracy of our TP models by comparing predicted aircraft positions against actual aircraft positions.

We measure error between a predicted position $\widehat{pos}_{ts_{now}+t}$ and the actual aircraft position $pos_{ts_{now}+t}$ for a look-ahead time of t minutes. There are two types of errors that can be measured: horizontal error and vertical error. Horizontal error is the horizontal distance between the two positions $pos_{ts_{now}+t}$ and $\widehat{pos}_{ts_{now}+t}$, and is always positive. Vertical error is the difference in pressure altitude between the two positions $pos_{ts_{now}+t}$ and $\widehat{pos}_{ts_{now}+t}$, and is positive (negative) when the actual position $pos_{ts_{now}+t}$ is above (below) the predicted position $\widehat{pos}_{ts_{now}+t}$. We omit the sign when calculating statistics for vertical error, *i.e.* only the absolute value of the vertical error is considered.

We compute the mean and the standard deviation of the errors for all predictions within a test trajectory to obtain the intra-trajectory horizontal and vertical errors.

After that, we compute the mean and the standard deviation of all intra-trajectory horizontal and vertical errors within each test month (giving each trajectory an equal weight despite unequal number of predictions in each trajectory due to different flight lengths) to obtain the intra-month horizontal and vertical errors.

Finally, we compute the mean and the standard deviation of all intra-month horizontal and vertical errors (giving each month an equal weight despite unequal number of trajectories in each month)

to obtain the overall mean and standard deviation of errors for the whole test set.

5.5 Comparison of Prediction Time

To compare the prediction times of the models, we computed the average time each model takes for predicting a trajectory of 20 minutes into the future, by finding the mean of all prediction times taken by the model for all 4 flight routes and all 12 months of test data. We found that the HMM approach was able to make predictions as efficiently as the baseline models, with an average prediction time of 39.8 milliseconds, compared to 36.9 and 40.4 milliseconds for the baseline models KBM and MTM, respectively.

5.6 Comparison of Prediction Accuracy

The mean overall horizontal and vertical errors for all look-ahead time and for all routes are presented in Figure 3. It can be seen that the HMM is able to make more accurate predictions than the baseline models, especially for longer look-ahead times of more than 7 minutes. While the KBM baseline is able to make highly accurate predictions for short look-ahead times (7 minutes or less), its prediction accuracy drops rapidly as look-ahead time increases, due to the increasing uncertainty in the changing kinematics parameters. The errors for the MTM baseline suggest that it performs equally well with the HMM near a look-ahead time of 20 minutes, as seen from their errors converging for the vertical error of route R2 and both horizontal and vertical errors for R4. To verify whether this observation is correct, we study the intra-month errors at a look-ahead time of 20 minutes presented in Figure 4. Contrary to what we might conclude from looking at Figure 3 alone, Figure 4 shows that the MTM has a larger standard deviation for intra-month errors than the HMM almost all of the time, which indicates inconsistent performance. This inconsistent performance is due to the tendency of the MTM to select the trajectory that represents the shortest and most direct route between the departure and the arrival airports, resulting in very accurate predictions for the bulk of test trajectories, but extremely bad predictions for some trajectories that take a longer route due to external factors such as congestion at the arrival airport. Furthermore, it is noticeable for routes R2, R3, and R4 that the HMM did not perform as well for the first few months of the year compared to other months. Upon close inspection of the data, we attribute that to the HMM not having enough training data for the first 3 months for route R2, and for the first 8 months for routes R3 and R4, as seen in Figure 2.

We also present the mean and standard deviation of overall errors at a look-ahead time of 20 minutes in Figure 5. The KBM baseline had the best performance for route R3 and the worst for route R2, which is because route R3 has the longest mean flight time among all routes, while route R2 has the shortest mean flight time, as seen in Table 3. Flights on a longer route spend a larger proportion of the flight time cruising in a straight line towards the destination airport, allowing the KBM to make accurate predictions because there is little change in direction or speed of the flight trajectory for these large portions of the flights. The MTM baseline has worse performance for routes R1 and R2 among the 4 routes, as seen from the high mean and high standard deviation of both horizontal and vertical errors for routes R1 and R2. This is due to route R1 being the largest set of trajectories among all routes and route R2 being the second largest, as seen in Table 3, thus having more variability among trajectories and resulting in the inability of the median trajectory approach to make accurate predictions. The HMM, on the contrary, has the best performance for route R1 compared to other routes, which is the result of having a large amount of training data.

We can thus conclude that our proposed HMM outperforms, in general, the baseline models significantly, except when making predictions for very short look-ahead times, for which the KBM approach seems more appropriate. A KBM approach, however, is in general not appropriate for look-ahead times larger than a few minutes, giving predictions with rapidly increasing errors as the look-ahead time increases (even though the *average* prediction accuracy of KBM improves with increasing mean flight time because longer flights cruise for a longer time just on a straight line with more or less constant speed). The MTM approach also performs overall significantly worse than our proposed HMM, and is typically even outperformed by the KBM approach for up to intermediate look-ahead times. It is clear that the larger the training set, the better the performance of our proposed HMM model. Therefore, we expect the performance gap between our proposed HMM and the baseline models to increase as more and more historical flight data becomes available.

As demonstrated for the maximum look-ahead time of 20 minutes for route R1, which has the largest dataset available for the training, our proposed HMM achieved a mean overall horizontal error of 22.4 km and mean overall vertical error of 2100 ft. In the horizontal dimension, the error of 22.4 km is approximately 1.65 times of a grid cube away (having width of 13.545 km), and is in relation to an aircraft being able to travel 219 km in most cases (*i.e.* median) and 264 km in extreme cases (*i.e.* 90^{th} percentile) within 20 minutes. Looking at the vertical dimension, the error of 2100 ft is approximately 1.05 times of a vertical interval away (having a separation of 2000 ft), and is in relation to an aircraft being able to ascend or descend 15500 ft in most cases (*i.e.* median) and 26800 ft in extreme cases (*i.e.* 90^{th} percentile) within 20 minutes. The errors represent an improvement of 26.3% in horizontal error and 31.8% in vertical error over MTM, the best-performing baseline model.

6 CONCLUSIONS

In this paper, we have presented a learning approach to the online prediction of mid-flight aircraft trajectories. Through the mapping of the problem onto Hidden Markov Models, we were able to incorporate local weather information for trajectory prediction. We used a rich dataset of historical trajectories and performed extensive experiments using two baseline models for comparison. Using prediction accuracy metrics for both horizontal and vertical errors, we demonstrated that we were able achieve better effectiveness, *i.e.* better prediction accuracy compared to conventional approaches, just as efficiently as the baseline models, on an off-the-shelf laptop computer.

In the future, we plan to conduct more in-depth studies of the effect of incorporating weather and possibly other information in trajectory prediction. This will also involve conducting experiments

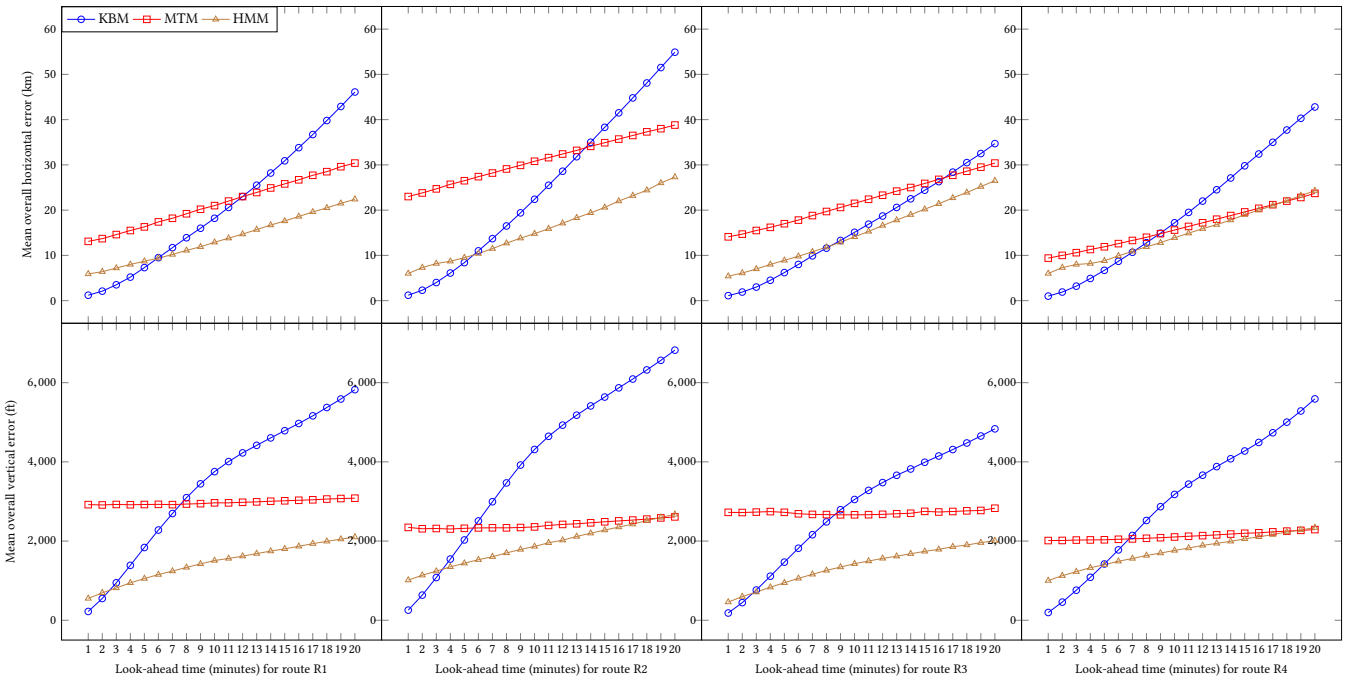

Figure 3: Mean overall errors for all look-ahead times

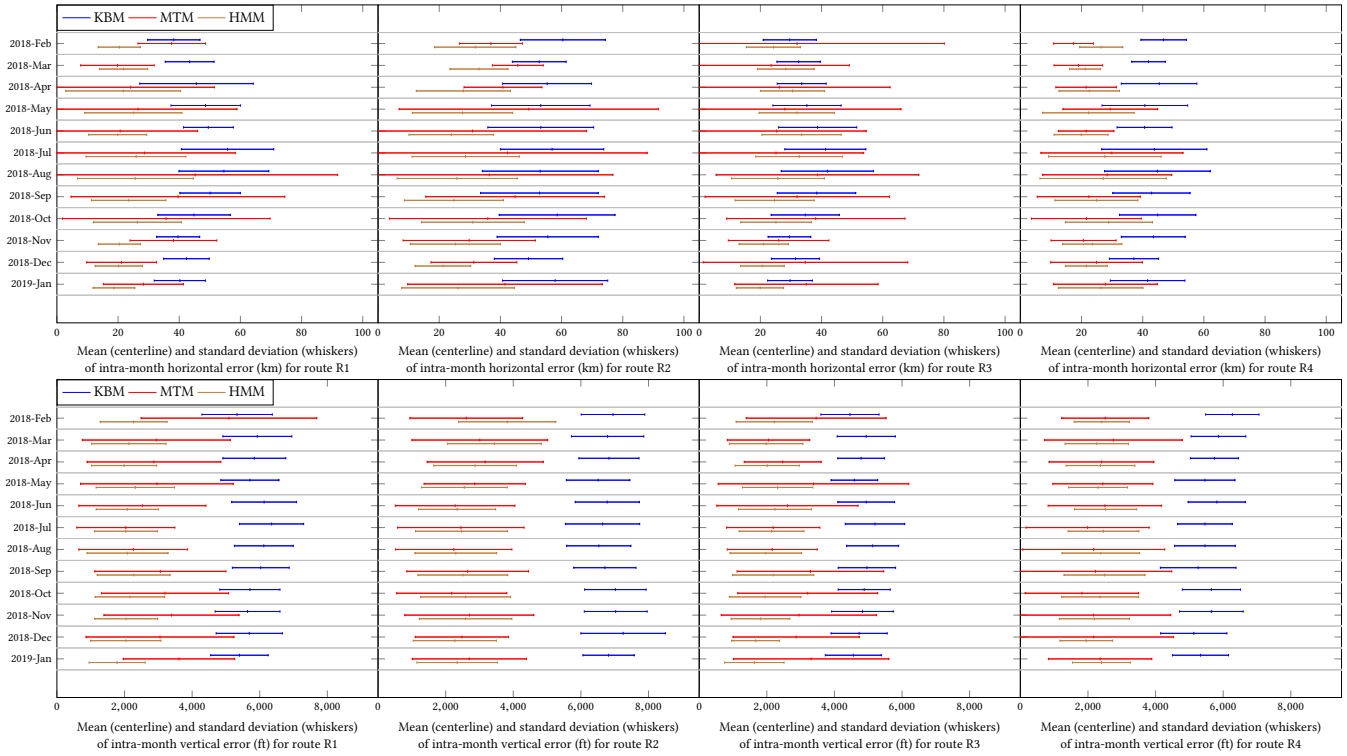

Figure 4: Mean and standard deviation of intra-month errors at look-ahead time of 20 minutes

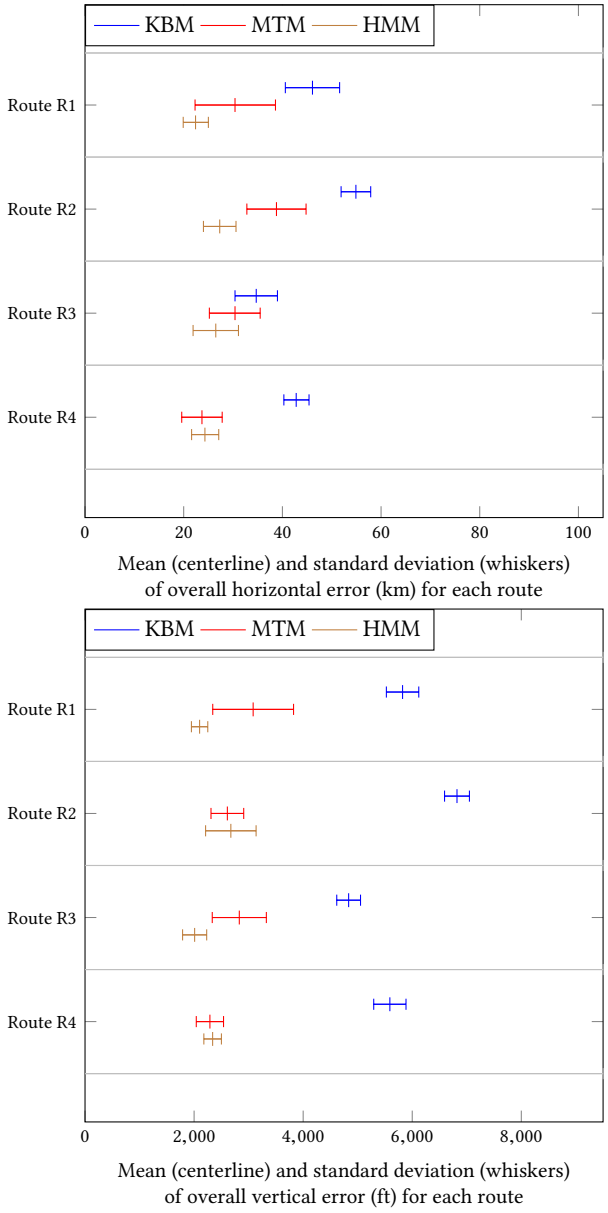

Figure 5: Mean and standard deviation of overall errors at look-ahead time of 20 minutes

on more varied flight routes, and exploring other ways of incorporating such information. We are also interested in investigating more sophisticated models, such as higher-order Markov Models, to achieve an even better accuracy than what we have presented in this study.

ACKNOWLEDGEMENTS

This work was supported in part by DSO National Laboratories, Singapore, and NSERC, Canada. We are also grateful to The OpenSky Network for providing the data used in this study.

REFERENCES

[1] Samet Ayhan, Pablo Costas, and Hanan Samet. 2018. Prescriptive analytics system for long-range aircraft conflict detection and resolution. In *Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems - SIGSPATIAL '18*. ACM Press. https://doi.org/10.1145/3274895.3274947

[2] Samet Ayhan and Hanan Samet. 2015. DICLERGE. In *Proceedings of the 8th ACM SIGSPATIAL International Workshop on Computational Transportation Science - IWCTS'15*. ACM Press. https://doi.org/10.1145/2834882.2834887

[3] Samet Ayhan and Hanan Samet. 2016. Aircraft Trajectory Prediction Made Easy with Predictive Analytics. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16*. ACM Press. https://doi.org/10.1145/2939672.2939694

[4] Stanley G. Benjamin, Stephen S. Weygandt, John M. Brown, Ming Hu, Curtis R. Alexander, Tatiana G. Smirnova, Joseph B. Olson, Eric P. James, David C. Dowell, Georg A. Grell, Haidao Lin, Steven E. Peckham, Tracy Lorraine Smith, William R. Moninger, Jaymes S. Kenyon, and Geoffrey S. Manikin. 2016. A North American Hourly Assimilation and Model Forecast Cycle: The Rapid Refresh. *Monthly Weather Review* 144, 4 (apr 2016), 1669–1694. https://doi.org/10.1175/mwr-d-15-0242.1

[5] Federal Aviation Administration (FAA). 2016. Pilot's Handbook of Aeronautical Knowledge, Chapter 11: Aircraft Performance. Retrieved March 19, 2019 from https://www.faa.gov/regulations_policies/handbooks_manuals/aviation/phak/

[6] Maxime Gariel, Ashok N. Srivastava, and Eric Feron. 2011. Trajectory Clustering and an Application to Airspace Monitoring. *IEEE Transactions on Intelligent Transportation Systems* 12, 4 (dec 2011), 1511–1524. https://doi.org/10.1109/tits.2011.2160628

[7] Chester Gong and Dave McNally. 2004. A Methodology for Automated Trajectory Prediction Analysis. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*. American Institute of Aeronautics and Astronautics. https://doi.org/10.2514/6.2004-4788

[8] International Air Transport Association (IATA). 2016. IATA Forecasts Passenger Demand to Double Over 20 Years. Retrieved March 19, 2019 from https://www.iata.org/pressroom/pr/Pages/2016-10-18-02.aspx

[9] B. H. Juang and L. R. Rabiner. 1991. Hidden Markov Models for Speech Recognition. *Technometrics* 33, 3 (aug 1991), 251–272. https://doi.org/10.1080/00401706.1991.10484833

[10] Daniel Jurafsky and James H. Martin. 2018. Speech and Language Processing (3rd edition draft). Chapter A: Hidden Markov Models (Draft of September 11, 2018). Retrieved March 19, 2019 from https://web.stanford.edu/~jurafsky/slp3/

[11] OAG Punctuality League. 2018. On-time performance for airlines and airports and Top 20 busiest routes Based on full year data 2017. Retrieved March 19, 2019 from https://www.oag.com/2018-airport-airline-on-time-performance-report

[12] I. C. Mogotsi. 2009. Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze: Introduction to information retrieval. *Information Retrieval* 13, 2 (Sept. 2009), 192–195. https://doi.org/10.1007/s10791-009-9115-y

[13] Network Common Data Form (NetCDF). 2019. Network Common Data Form (NetCDF). Retrieved March 19, 2019 from https://www.unidata.ucar.edu/software/netcdf/

[14] The OpenSky Network. 2019. The OpenSky Network. Retrieved March 19, 2019 from https://opensky-network.org

[15] National Weather Service (National Oceanic and Atmospheric Administration). 2019. Formula for calculation of pressure altitude. Retrieved March 19, 2019 from https://www.weather.gov/media/epz/wxcalc/pressureAltitude.pdf

[16] Mike Paglione and Robert Oaks. 2007. Implementation and Metrics for a Trajectory Prediction Validation Methodology. In *AIAA Guidance, Navigation and Control Conference and Exhibit*. American Institute of Aeronautics and Astronautics. https://doi.org/10.2514/6.2007-6517

[17] Rapid Refresh (RAP). 2019. Rapid Refresh (RAP). Retrieved March 19, 2019 from https://rapidrefresh.noaa.gov

[18] Matthias Schafer, Martin Strohmeier, Vincent Lenders, Ivan Martinovic, and Matthias Wilhelm. 2014. Bringing up OpenSky: A large-scale ADS-B sensor network for research. In *IPSN-14 Proceedings of the 13th International Symposium on Information Processing in Sensor Networks*. IEEE. https://doi.org/10.1109/ipsn.2014.6846743

[19] John Snyder. 1987. Map Projections – A Working Manual. 1395 (01 1987).

[20] GeoTools: The Open Source JAVA GIS Toolkit. 2019. GeoTools: The Open Source JAVA GIS Toolkit. Retrieved March 19, 2019 from http://www.geotools.org/

[21] Thomas Tomkins Warner. 2009. *Numerical Weather and Climate Prediction*. Cambridge University Press. https://doi.org/10.1017/cbo9780511763243

[22] Yu Zheng. 2015. Trajectory Data Mining. *ACM Transactions on Intelligent Systems and Technology* 6, 3 (may 2015), 1–41. https://doi.org/10.1145/2743025