

## Research Article

# An Air Traffic Controller Action Extraction-Prediction Model Using Machine Learning Approach

Duc-Think Pham , Sameer Alam, and Vu Duong

*Air Traffic Management Research Institute, School of Mechanical and Aerospace Engineering, Nanyang Technological University, Singapore*

Correspondence should be addressed to Duc-Think Pham; [dtpham@ntu.edu.sg](mailto:dtpham@ntu.edu.sg)

Received 9 July 2020; Revised 21 October 2020; Accepted 26 October 2020; Published 18 November 2020

Academic Editor: Vladimir Stojanovic

Copyright © 2020 Duc-Think Pham et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In air traffic control, the airspace is divided into several smaller sectors for better management of air traffic and air traffic controller workload. Such sectors are usually managed by a team of two air traffic controllers: planning controller (*D*-side) and executive controller (*R*-side). *D*-side controller is responsible for processing flight-plan information to plan and organize the flow of traffic entering the sector. *R*-side controller deals with ensuring safety of flights in their sector. A better understanding and predictability of *D*-side controller actions, for a given traffic scenario, may help in automating some of its tasks and hence reduce workload. In this paper, we propose a learning model to predict *D*-side controller actions. The learning problem is modeled as a supervised learning problem, where the target variables are *D*-side controller actions and the explanatory variables are the aircraft 4D trajectory features. The model is trained on six months of ADS-B data over an en-route sector, and its generalization performance was assessed, using crossvalidation, on the same sector. Results indicate that the model for vertical maneuver actions provides highest prediction accuracy (99%). Besides, the model for speed change and course change action provides predictability accuracy of 80% and 87%, respectively. The model to predict the set of all the actions (altitude, speed, and course change) for each flight achieves an accuracy of 70% implying for 70% of flights; *D*-side controller's action can be predicted from trajectory information at sector entry position. In terms of operational validation, the proposed approach is envisioned as ATCO assisting tool, not an autonomous tool. Thus, there is always ATCO discretion element, and as more ATCO actions are collected, the models can be further trained for better accuracy. For future work, we will consider expanding the feature set by including parameters such as weather and wind. Moreover, human in the loop simulation will be performed to measure the effectiveness of the proposed approach.

## 1. Introduction

The primary purpose of Air Traffic Control (ATC) worldwide is to prevent collisions, organize and expedite the flow of air traffic, and provide information and other support for pilots [1]. In regions, where the Air Traffic Management (ATM) system is well developed, three types of control facilities play a critical role during the successive phase of a typical flight: (1) the airport traffic control tower (aerodrome control), (2) the terminal airspace control center (approach control), and (3) the en-route control center (area control) [2]. The en-route airspace is organized vertically and horizontally according to local air structure, and traffic flows

into a smaller area called “sectors.” A sector is generally considered as a fundamental “unit” of airspace volume from the ATM point of view. The en-route sector is usually managed by a team of two air traffic controllers: planning controller (*D*-side) and executive controller (*R*-side) [3].

Both *D*-side and *R*-side air traffic controllers (ATCOs) are responsible for airspace monitoring, conflict detection, and resolution, along with managing route or altitude modification requests from the aircraft. The difference between the two roles lies in the strategic and tactical levels of intervention. The *D*-side controller is primarily responsible for processing flight-plan information to plan, coordinate, and organize the flow of air traffic entering into a sector. The

*D*-side controller uses the flight-plan information and employs Medium Term Conflict Detection (MTCDD) tool [4] to predict aircraft trajectories in a 20 minute look-ahead time window. *D*-side controller employs a variety of strategies/actions, i.e., combination of altitude, speed, course change, hold maneuvers, etc. to maintain an orderly flow of the incoming traffic in a sector. Thus, it minimizes crossings events which may lead to loss of separation. This ensures, at a tactical level, a minimum intervention is required from the *R*-side controller while managing the air traffic in a given sector. The *R*-side controller uses the Short Term Conflict Alert (STCA) tool [5] to predict any loss of separation in a 4 to 8 minutes look-ahead time window. The *R*-side controller is mainly concerned with tactical interventions to maintain safe separation between flights.

The *D*-side controller receives flight plan information of the flight before it enters the sector (transfer of communication). At this point, the aircraft is in contact with both, the previous sector *D*-side controller as well as the next sector *D*-side controller. The *D*-side controller then negotiates with the aircraft regarding entry flight level, entry speed, and entry way-point depending upon the strategic situation in his/her sector at a certain look-ahead time. The primary objective of this planning is to maintain an orderly flow of traffic and to minimize crossings which may lead to a loss of separation (LOS) scenario for an *R*-Side controller to intervene. Once the aircraft enters the sector boundaries (transfer of control), the *D*-side hands off the aircraft to the *R*-side controller who then provides ATC services via radio communication. In some circumstances (e.g., bad weather), the aircraft may need to be handed off differently than the letter of agreement. In those cases, the *D*-side controller must coordinate with the other sector controller to ask for approval for another route which is not specified in the letter of agreement before the aircraft cross the boundary.

As ATC is becoming increasingly complex and dynamic, the role of ATCOs in an ATC system is getting more and more challenging [6]. Within the safety-critical domain of ATC, workload remains a dominant consideration in improving ATC system performance. Since the main responsibility of the *D*-side controller is to manage and organize traffic flow such that the tactical flight interventions from the *R*-side controller are minimized, it is desirable to automate the *D*-side controller tasks such that its task load is reduced. A possible way is to develop a mechanism that can learn and predict *D*-side controller's traffic management strategies for a given traffic scenario.

The contribution of this paper is modeling the learning problem by extracting the *D*-side controller's actions. The novelty is in formulating the problem as predicting a quantitative response for an observation, i.e., classifying that observation, since it involves assigning the observation to a class. Thus, we have approached the classical simulation-modeling problem as a data-driven problem where controller's strategies are learned from the natural behavior of a human *D*-side controller, the way he/she handles different traffic scenarios. This task requires visualizing, analysing, and understanding 6-month traffic trajectory data for formulating a reasonable and solvable question. Besides,

another contribution of this paper is on learning mechanisms that may offer better predictability for planning controllers under different traffic scenarios. We have adopted tree-based ensemble methods for learning ATCO's actions in a complex air traffic environment on a real dataset. The method involves segmenting the predictor space into several simple regions, and then in order to make a prediction from a given observation, a mean or mode of the training observations in the region, to which it belongs, is used. Such set of splitting rules were then used to segment the predictor space and then summarized in the form of a tree. Such multiple trees can then be combined to yield an accurate consensus [7]. Furthermore, we have demonstrated that tree-based methods can closely mirror human decision-making than other classification approaches [8]. Even though the used learning models are well-known ones, preparing the analytical dataset for training them are challenging because of the noises and missing in trajectories. Besides, detecting and removing outliers such as holding or incomplete trajectories under those conditions is also another challenge.

This paper is organized as follows. Section 2 introduces the background for this research question especially focusing on successful applications of machine learning and tree-based ensemble learning in air traffic management. Section 3 describes the overview of the proposed approach for predicting planning controller's action including data preparation steps. Section 4 discusses in detail our data analysis and data processing steps for extracting ATCO's action from trajectory data. Section 5 introduces our methodology for developing predictive models using two different ensemble learning techniques: Random Forest and Extreme Gradient Boosting. Section 6 describes our predictive models for predicting controller decision and taxi time for departure flights and introduces our experimental setup for evaluating the predictive models as well as the discussion and analysis of the results. Section 7 is a discussion section on the implementation for ATCO command from our predicted results. Finally, Section 8 presents our conclusions and future work.

## 2. Background

The quest of understanding and learning opponent's strategies in games such as Chess, Backgammon, and Game of Go and be able to predict the next move of an opponent, is well known in literature [9–12]. Machine learning methods such as deep neural networks, tree-search methods, and Bayesian reinforcement learning have recently been quite successful in learning game strategies and outperforming world champions [13–16]. However, a major assumption in such machine learning algorithms is that the training and future data must be in the same feature space and have the same distribution [17].

In air traffic, the feature space (airspace structure including its airways and way points) and the data distribution (aircraft trajectory points) varies significantly. Since every airspace is unique, thus every air traffic scenario has a different data distribution. Furthermore, traffic flow

management relies almost exclusively on the judgment of air traffic controllers for decisions [18]. Previous research studies into identifying air traffic controller's strategies from traffic data have found some interesting insights, but they were generic in nature and lack any predictability of ATCO actions given a traffic scenario. For example, in [19], it was found that, in the presence of conflict between a few aircraft, the velocity variation strategy seems to cost more (in terms of the time of flight) than the course angle deviation strategy. In [4], the authors developed an evolutionary computation framework to identify air traffic maneuvers that may expose a traffic scenario to loss of separation but fall short of generalizing it to a range of traffic scenarios. In [20], authors predicted air traffic controller workload from past sector merge and split actions but could not generalize the learning to new sectors due to overfitting of the training data. In [21], the authors proposed using game theory for conflict resolution in en-route airspace. Apart from en-route airspace, machine learning methods have also been applied in terminal airspace. For example, in [22], a simulator was designed which can simulate control of air traffic and landing clearance and departure by using backpropagation network based on various controlling parameters, but for single-runway only.

With the availability of aviation data and the significant advancements in computational power, data-driven and machine-learning-based methods have recently become a very promising approach to many challenging problems in air traffic management, such as taxi-out time prediction [23, 24], aircraft sequencing [25], aircraft performance parameter predicting [26], air traffic flow extraction [27], flight delay prediction [28, 29], and aircraft trajectory prediction [30, 31]. However, to the best of authors' knowledge, there has not been any study to extract controller actions or decision as well as learn controller strategy from real data.

In this study, we consider tree-based ensemble learning as our learning algorithm which has been applied in a variety of field, including transportation [32], energy [33, 34], networking [35–37], and air traffic management [38, 39]. It is popular for not only being able to achieve high accuracy levels [32, 40] but also enabling the interpretation of the importance of features used in the predictions. In human-centric and safety-critical domain such as air traffic control, this is a critical property to provide ATCO understanding of proposed models and solutions [41, 42]. Moreover, tree-based ensembles require minimum data preprocessing and are capable of fitting highly nonlinear data and handling big data.

### 3. Overview

The proposed learning process is demonstrated in Figure 1, which contains preprocessing data, extracting ATCO actions, and building Learning Models. The 4D trajectories points for individual flights are constructed directly from ADS-B raw data and the spatial information about the sector is collected and processed from Aeronautical Information Publication (AIP). Then, preprocessing techniques are applied to clean data and remove noises and outliers

trajectories from the raw data. Two points from each flight (first and last points in terms of *timestamp*) were used to identify the new flight trajectory upon entering the sector, which reflects the main course of the trajectory inside the sector. Only simple actions are extracted by comparing the above result with flight information (speed, altitude, and course) at the exiting point. Those can be seen as the changes in *Speed*, *Course*, and *Altitude*, which need to be applied for each flight to reach the exit point at the given 4D position (latitude, longitude, altitude, and time). Up to this point, two sets are generated: action values (continuous) and actions  $([-1, 0, 1])$  which are related to Ground Speed Rate, Vertical Speed, and  $\Delta$  Course for each flight. Finally, using the information at entry points as the input and the mentioned two sets as targets, we build two groups of random forest models: regression and classification models. Outputs of those models are the changes or applied actions for each given flight at its entry point.

Noting that, in the scope of this work, only three mentioned groups of actions are considered, and actions such as holding or more complex actions will be studied in future work. Besides, in future work, more traffic information will be included to increase the accuracy of prediction.

**3.1. Selected Sector.** For this research purpose, we have identified Sector 2E, an en-route area within Kuala Lumpur Flight Information Region (FIR), managed by Singapore Area Control Center (ACC), for providing air traffic service from flight level FL120 (12,000 *feet*) to flight level FL360 (36,000 *feet*) inclusive. We have chosen Sector 2E in Singapore FIR as this is the main feeder sector into the Singapore TMA, having interface with three FIR boundaries HO CHI MINH FIR, BANGKOK FIR, and KUALA LUMPUR FIR. This sector has a high degree of flight vectoring and tactical trajectory management making it a natural choice for the Controller Action Prediction problem. Figure 2 depicts the spatial characteristics of the selected sector. It takes about on average 5 minutes for a typical flight to cross sector. The sector contains 8 way-points and is crossed by 8 Air Traffic Service (ATS) Route. There is one crossing in the sector and one convergence point in the south of the sector (way-point VMR).

**3.2. Dataset and Data Preparation.** The ADS-B data is collected for the South-East Asian region for six-month period (from September 2016 to February 2017). The dataset contains three main weather conditions in Singapore: southwest monsoon season, intermonsoon period, and northeast monsoon season (Table 1) which are different in both the strength and the direction of winds. The dataset is sufficient for this study as it captures all the main weather and traffic cycles in Singapore airspace. Each sample of data contains features, as shown in Table 2, and an example of one-row sample of 4D trajectory data is shown in Table 3. Even though our dataset is not available for public, similar ADS-B data for European airspace (from OpenSky Network

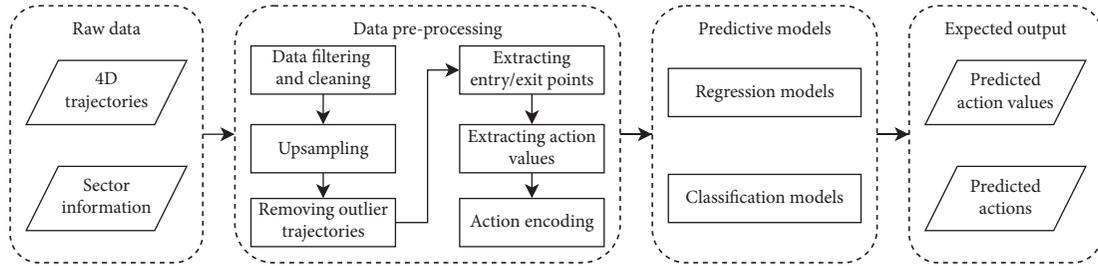


FIGURE 1: The illustration of our approach for building predictive models.

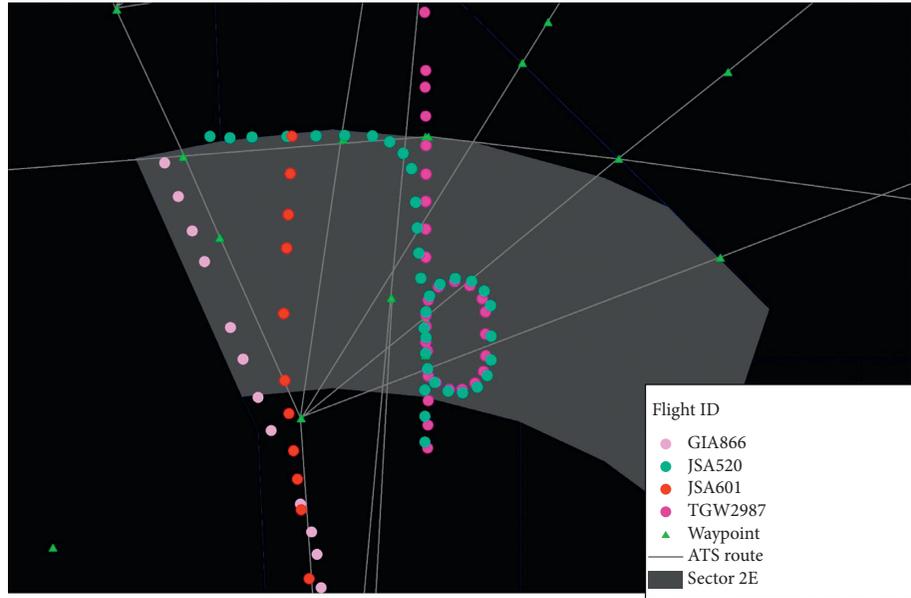


FIGURE 2: An example of four different flight trajectories reflecting speed, heading change, and hold maneuver in the sample data for sector 2E.

TABLE 1: Singapore seasonal information of the dataset.

Seasons	Dataset	Description
Southwest monsoon	09/2016	Southeasterly to southerly wind
Intermonsoon	10-11/2016	Lighter and more variable wind
Northeast monsoon	12/2016, 01/2017	Northeast monsoon-wet phase
	02/2017	Northeast monsoon-dry phase

TABLE 2: 4D trajectory data features.

Feature	Description
Position	Latitude (decimal degrees), longitude (decimal degrees), altitude (ft)
Ground speed	Horizontal speed relative to the ground (knots)
Rate of climb	Altitude change (feet per minute)
Course	Aircraft course relative to north (decimal degrees)
Flight ID	Unique serial number represents each flight
Time	Time (UTC) that data been recorded

TABLE 3: One row sample of 4D trajectory data.

Field name	Value
Flight ID	CDG4963-1482966600-schedule-0000
Time (UTC)	12/31/2016 00:58:14
Latitude (dec. deg)	34.29153
Longitude (dec. deg)	108.5708
Groundspeed (kts)	182
Altitude (ft)	3900
Rate (fps)	-514
Course (deg)	18

or ADS-B Exchange) are available for applying our proposed approach.

Each group of records represents trajectories of flights, carrying the status of the flight spatially throughout time. Figure 2 illustrates trajectories of 4 different flights passing through the sector. Sample points with the same color belong to the same flight and the time interval between each point is  $\approx 15$  seconds.

The original ADS-B dataset is a large dataset with noises and missing data points. Moreover, with the given spatial information of the sector 2E, only a subset of trajectories should be considered and investigated. Thus, some pre-processing steps need to be applied:

- (1) Because each sector is defined as a three dimension volume, thus we apply a 3D spatial-filtering to filter out all trajectories which do not pass through the sector. In this step, we filter out using both lateral (sector boundary) and vertical (FL120 to FL360) conditions. For example, we found that there were 12,141 flights that pass through the 2E sector in December 2016 data.
- (2) The second 3D spatial-filtering (similar conditions) is applied to filter out trajectory segments outside of the selected sector. It is separated from the first step just for flexibility in manipulating the filtering criteria.
- (3) Outlier trajectories are detected and removed from the dataset, in which trajectories that do not pass the sector's lateral boundary or have significantly long travel distance and time are all considered as outliers. In the context of our research problem, holding trajectories are also treated as outliers. They are rare events that only contribute less than 1.7% of whole data but may affect the predictive models significantly.
- (4) To deal with missing data points, we first remove all the flight trajectories which have less than four data points in the sector. After this step, the working dataset contains  $\approx 75\%$  of flights from original data. The remaining dataset still has missing data which makes the time interval become inconsistent. Furthermore, for later processing, we aim to extract the entry and exit position of flights on the sector's boundary; then, a dense and consistent time interval in the dataset is required. Therefore, to achieve that

the remaining trajectories are up-sampling (interval = 1 second) using interpolation techniques.

## 4. ATCO Action Extraction

In a practical scenario, pilots communicate with the *D*-side controller while entering the sector and with the *R*-side controller once inside the sector. An aircraft trajectory bears signatures of both *R*-side and *D*-side controller actions. However, *D*-side controller actions can be identified in the trajectory data by observing the trajectory prior to entering a sector.

To better understand the relationship between controller actions and air traffic trajectory data derived from ADS-B, we first visualized the 4D data with GIS. The observing airspace is visualized discretely by grids with 3 nautical miles in length and width, action in the same grid will be summed up, and every grid will be classified into 5 classes using the Jenks Natural Breaks Classification method [43], which is a data clustering method designed to reduce the variance within classes and maximize the variance between them. The color of the grid from yellow to red means the higher frequency of certain features appears in the position, and the first class was not set to visualize. Figures 3 and 4 show the spatial density of entering points and exit points of the aircraft of sector 2E, while Figures 5–7 show spatial distribution of ACT actions in the sector. These figures indicate that there are patterns in ATCO actions.

*4.1. Extracting Flight Change/Action Values.* Observing from flight trajectories as in Figure 2, there are multiple changes in trajectories of aircraft when flies over the sector. However, the flight usually enters the sector at a specific region and should be directed to follow the designed ATS routes and way-points which means all the changes should be applied for aircraft to reach a specific region to exit the sector. Figure 8 presents some examples for trajectories passing Sector 2E of Singapore FIR. Flights with the same flight' identification will share their flight plans, reflecting by the pattern in their trajectories. However, in terms of sector entry and exit positions, they show a significant dispersion. The dispersion may come from multiple factors such as weather or controller's decisions. Besides, another interesting observation is the consistent relation between flights' entry and exit positions for the demonstrated sector. The assumption is that ATCO has its own pattern or strategy in handling traffic of a given sector. Then, the flights which enter the sector at a specific region will be navigated to similar region for exiting the sector. Thus, instead of using flight plan information for prediction, we mainly focus on using entry and exit positions relation to capture and validate ATCO's actions. This approach can capture the major changes of the flight in the sector. Three values will be extracted from those pairs of points:

*4.1.1. Ground Speed Rate.* While cruising inside sector 2E, flight speed usually varies. However, because of the nature of this sector, three common and simple trends can be observed

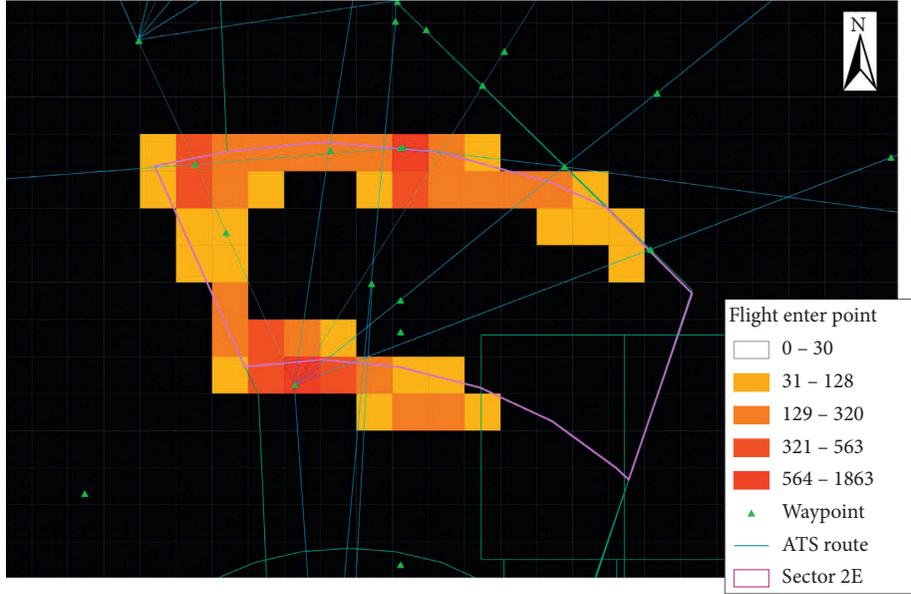


FIGURE 3: Distribution of entry points in sector 2E.

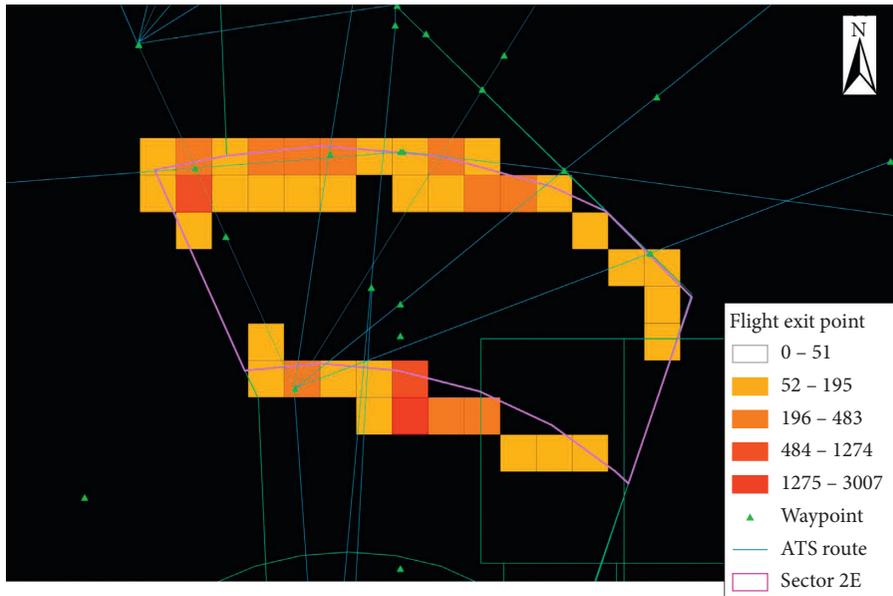


FIGURE 4: Distribution of exit points in sector 2E.

from data: maintaining speed, increasing (speed up), and decreasing (slow down), see in Figure 9. It indicates that the rate of ground speed change is quite stable and can be used as an action of flight. From that observation, the rate of change is extracted and considered for the next learning steps. In detail, it is computed based on estimating the required rate for a flight with a given speed at the entry point to travel to the exit point. In the case of unstable ground speed, the estimated rate of change in ground speed can be considered as the average rate of change:

$$\begin{aligned} \text{Distance} &= \bar{v}.T = \int_0^T v_t dt = \int_0^T (v_0 + a.t) dt \\ \bar{v}.T &= v_0.T + \frac{1}{2}a.T^2 \\ a &= \frac{2(\bar{v} - v_0)}{T}, \end{aligned} \quad (1)$$

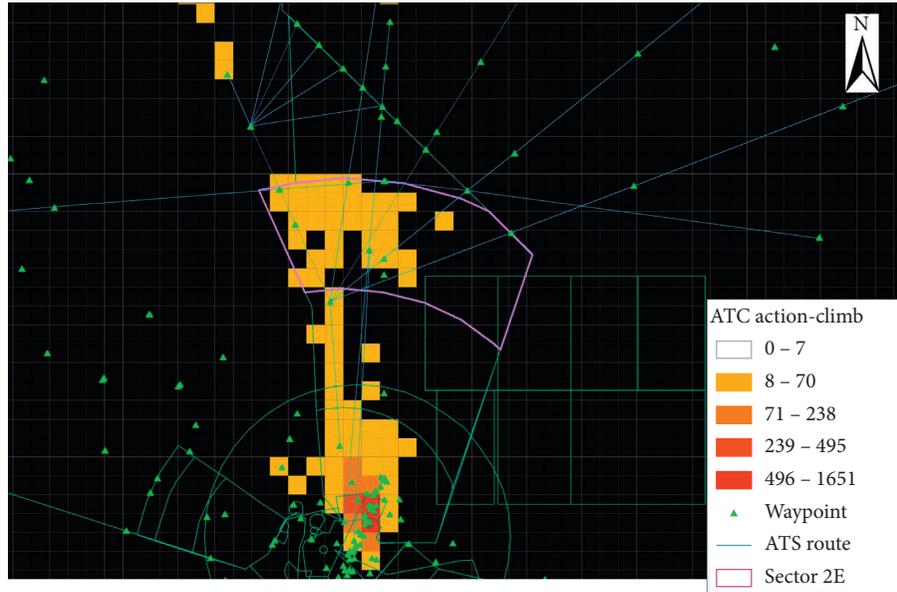


FIGURE 5: Distribution of ATCO action-climb.

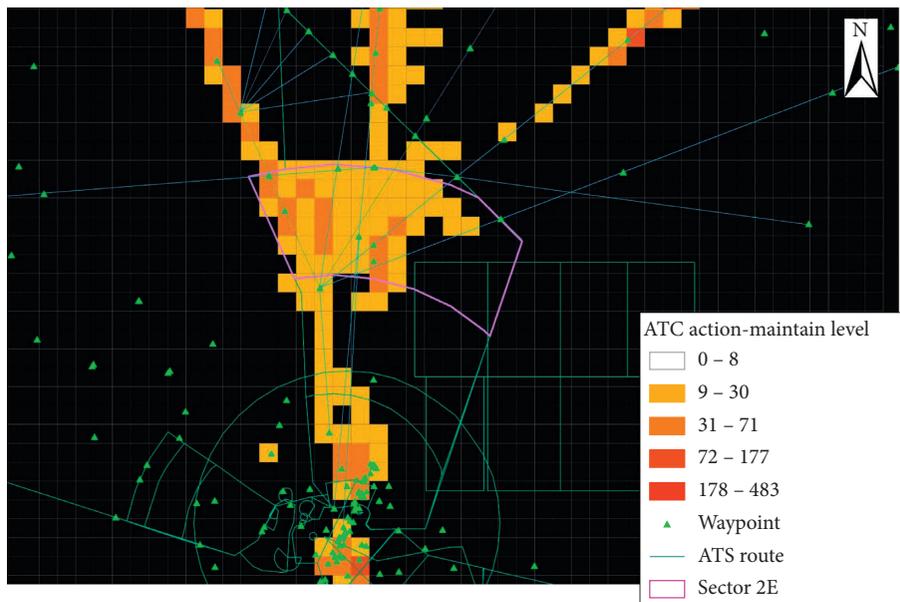


FIGURE 6: Distribution of ATCO action-maintain.

where  $\bar{v}$  is the average speed of aircraft through the sector,  $T$  is the total travel duration,  $v_0$  is the speed at entry point, and  $a$  is the acceleration (ground speed rate).

**4.1.2. Vertical Speed.** The actions related to vertical speed: similar to Ground Speed Rate, we also observe some common trends in altitude changes from data. The vertical speed is used as the vertical actions and computed simply based on the ratio of difference in altitude between entry and exit points and the travel duration.

**4.1.3.  $\Delta$ Course.** It is the difference between course at the entry point and “course in sector.” Since course of the flight varies throughout the sector and course at exit point also does not reflect the travel direction, we simplify the definition for “course in sector” as the direction from entry and exit points which is the expected direction for flight to travel through our sector. We use the  $\Delta$  Course instead of “course in sector” because it reflects the turning actions of flights after entering sector.

The detail of action extraction algorithm from entry and exit points is illustrated in Algorithm 1.

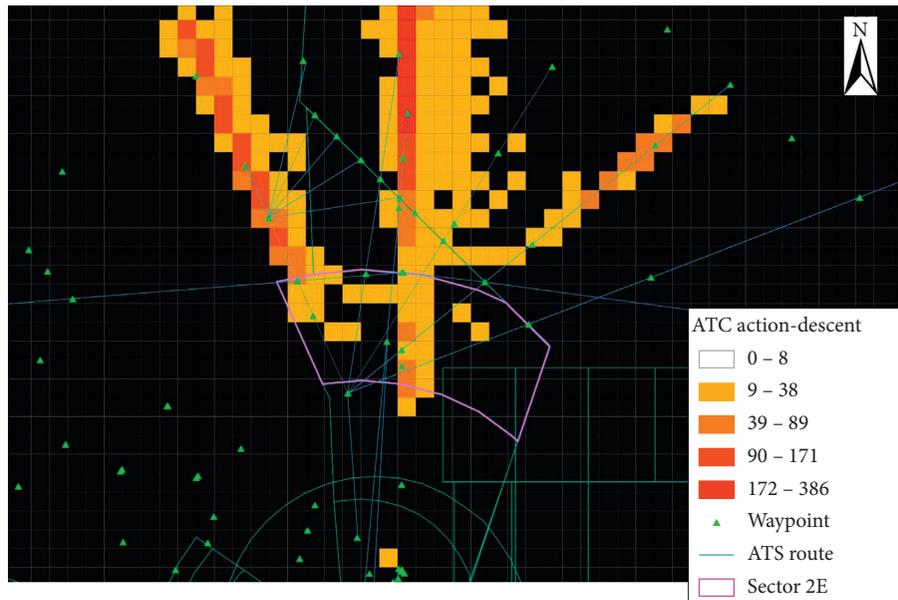


FIGURE 7: Distribution of ATCO action-descent.

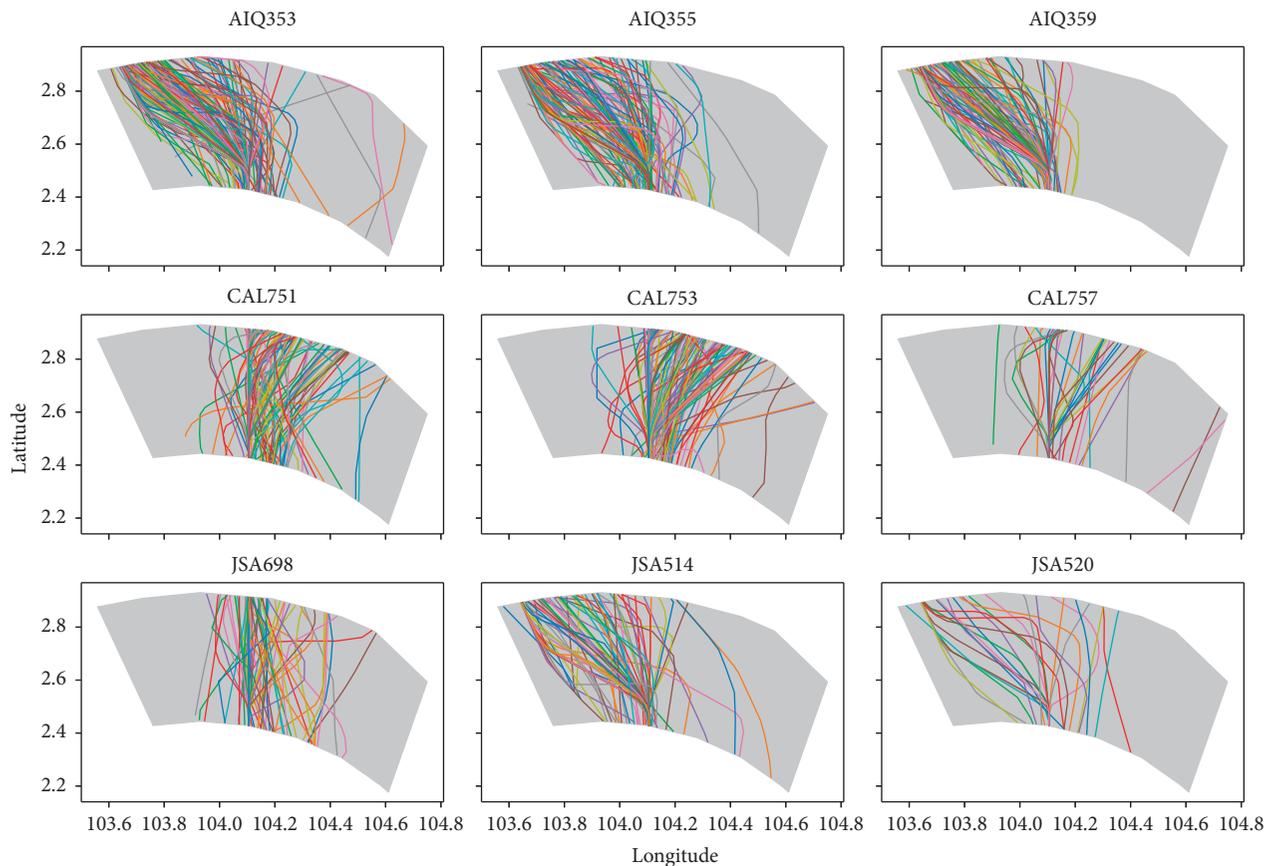


FIGURE 8: Each subplot is an example of multiple trajectories of the same flight's identification passing sector 2E for landing at Singapore Changi Airport in our dataset. The flows come from the North to the South with various entry points and exit point.

**4.2. Encoding ATCO Actions from Action Values.** As the requirement for supervised learning, we need a set of actions as labels for building classification models. Thus, for each

flight, the set of actions should be converted from extracted action values. Figure 10 displays idea on how the labels are encoded from values. There are three types of actions which

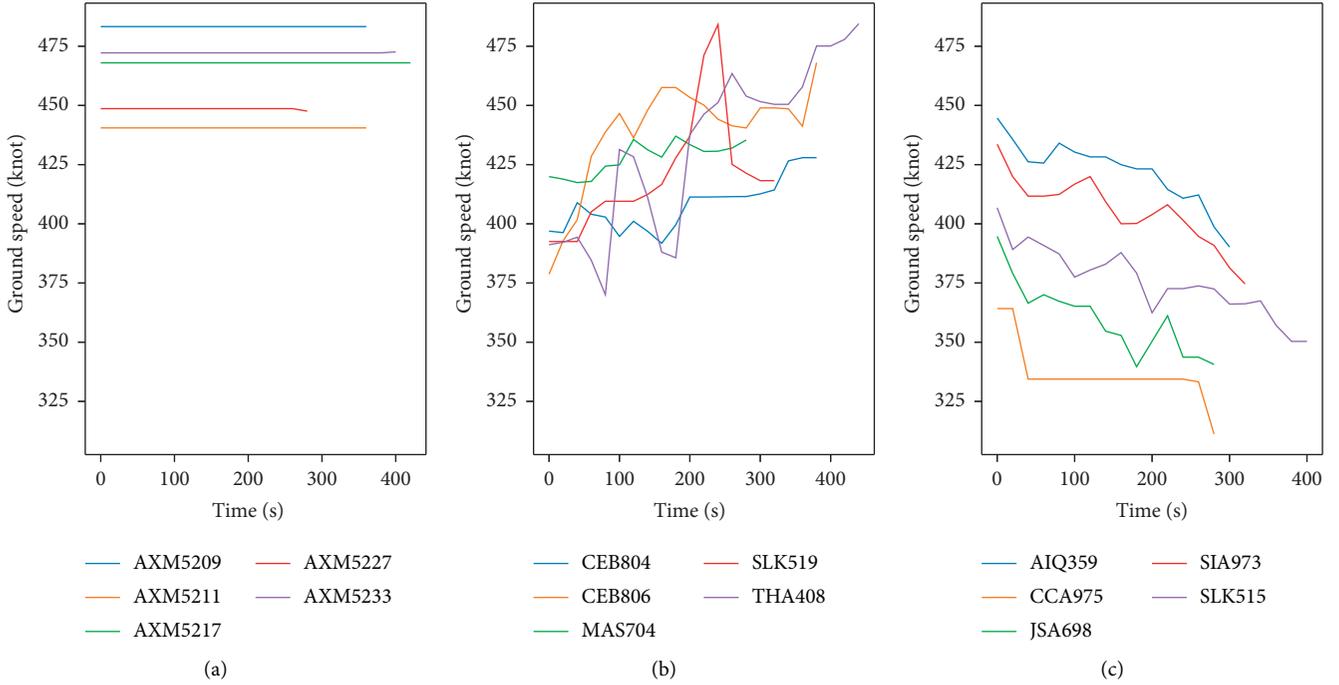


FIGURE 9: Example of trends in ground speed from real data. Three figures share the same  $y$ -axes which is the ground speed in *knot*. (a) Maintaining speed, (b) increasing speed, and (c) decreasing speed.

```

Input: Entry and Exit Points
Output: G Speed Rate, V Speed,  $\Delta$  Course
Travel_Distance = Euclidean_distance(Entry, Exit)
Travel_Time = Exit.Time - Entry.Time
Avg_GSpeed = Travel_Distance/Travel_Time
GSpeed Rate =  $2 * \text{Avg\_G Speed} - \text{Entry.Speed}/\text{Travel\_Time}$ 
VSpeed =  $\text{Exit.Alt} - \text{Entry.Alt}/\text{Travel\_Time}$ 
 $\theta = a \tan 2(\text{Exit.Lat} - \text{Entry.Lat}/\text{Exit.Lng} - \text{Entry.Lng}) \in (-\pi, \pi]$ 
if  $\theta \geq 0$  then
    Course_in_Sector =  $\theta$ 
else
    Course_in_Sector =  $2\pi + \theta$ 
end if
 $\Delta$ Course = Course_in_Sector - Entry.Course
return G Speed Rate, V Speed,  $\Delta$  Course

```

ALGORITHM 1: Action extraction from entry and exit points.

are related to ground speed rate, vertical speed, and delta course. Each of them is encoded into 3 actions:  $-1$ ,  $0$ , and  $1$  based on the thresholds selected as follows:

**Speed Action:** the  $\pm 10$  knots variance in the cruising speed of an aircraft can be considered as maintaining speed since those changes can happen without the need of permission from ATCO. In our work, we assume that any change of cruising speed more than 10 knots is considered as applied speed control. Thus, if the absolute change of speed between exit and entry points is less than 10 knots, we consider it as maintaining speed. Besides the expected travel time of the sector is 5 mins.

From both of that,  $\text{threshold\_sr} = \pm 0.017(\text{m/s}^2)$  is selected.

**Vertical Action:** due to the altimetry system error for aircraft in en-route phase, the recorded altitudes may have some dispersion with its true values. In [44], for safety reason, the authors have specified those errors are less than 2 flight level (200 ft). Inspired from that, in this work, if the absolute change in altitude between two points is less than 100 feet, we consider it as maintaining flight level. Besides the expected travel time of the sector is 5 mins. Thus,  $\text{threshold\_vs} = \pm 20(\text{feet/minute})$  is selected. The *climb* and *descend* actions are mainly

distributed corresponding to south and north of the sector which will have a strong influence in building predictive model given entry information of a flight.

Course Action: due to the errors in the navigation system, wind, or several other factors, even though the aircraft is maintaining its course, the recorded course may also have some variations. From analysing the data and considering the reasonable values for course error, we have selected  $\pm 3$  degrees as the error threshold for this action. Then, if the absolute of delta course is less than 3 degrees, we consider it as maintaining course.  $\text{threshold\_dc} = \pm 3$  (degree) is selected for course action encoding.

Figure 11 illustrates the distribution of all extracted actions. The distributions of speed actions and course actions have bell shapes (in (a) and (c)). From (c) and (d), we conclude that course action has balanced distribution. However, the mean of ground speed rate is positive; therefore, there are more *speed up* actions than others in speed action. It is confirmed by (b): around 86% of speed action in this sector is *speed up*. However, since every action is equivalently considered, we do not solve unbalanced problem in the learning model. In (e), the changes in vertical speed can be seen as two separated normal distributions. Then, there are only two major actions: *climb* and *descend* corresponding to two distributions. The maintain level is kept, but there are limited samples for this action, as seen in (f).

## 5. Methodology

In this part, we will describe our approach for predicting the next actions of flight after entering the sector. The training data includes features of flight at the entry point and the extracted actions from real data as the targets. Table 4 mentions list of features and all targets. We propose Random Forest Method [45] and Extreme Gradient Boosting (XGBoost) [46] for building predictive models of ATCO actions.

**5.1. Tree-Based Ensemble Learning Method.** The tree-based ensemble learning method is used for building our predictive models for both classification and regression tasks. In general, it constructs multiple decision trees which are trained and combined together to reduce the variance of the model (illustrated in Figure 12). It is used in different domains and predictive problems as it provides high accuracy with simple implementation. It is highly robust since it can deal with outliers/noises without skewing the prediction results and avoids overfitting due to the diversity of trees. One of the key advantages of tree-based ensemble learning method which suits our problem is its capability to handle unbalanced datasets and able to work with different types of features and range of feature values. Furthermore, the 4D trajectories are derived from ADS-B data which usually contains noisy data points and the input features have different meanings and scales. Moreover, the interpretability of the model is also considered for understanding the

important factors for predicting actions; thus, tree-based ensemble learning method is found suitable for this purpose. In this study, the two algorithms which we considered are Random Forest (RF) and XGBoost-Extreme Gradient Boosting (XGB).

*Random Forest* [45]. It applies a bagging technique to decision trees. To train the model, it creates a large number of trees by resampling the given data and combines them (using averages or voting technique) at the end of the process.

*XGBoost-Extreme Gradient Boosting* [46, 47]. Among tree-based ensemble algorithms, Gradient Tree Boosting [48] has shown its success in many applications and provided state-of-the-art results on many standard classification benchmarks [49]. It applies a boosting technique to decision trees. It also creates and combines a large number of trees; however, instead of combining them at the end, it starts the process at the beginning. The algorithm will train each subsequent tree using the residuals (the difference between the predicted and true values) of previous ones. In this work, we use a scalable machine learning system for tree boosting, called XGBoost. It has been widely used in a number of machine learning and data mining challenges with real-life data in Kaggle and KDDCup. Besides all advantages of tree boosting algorithms, the most important factor behind success of XGBoost is its scalability in all scenarios.

**5.2. Building Predictive Models.** There are two levels of prediction that should be investigated: whether we can predict the abstract action or the magnitude of each action. Each kind of setting can answer different questions. The first can support the analysis of controller strategy, while the second one can support for understanding and studying details of controller behaviors. However, they are useful and have strong relations; thus, in this work, we introduce both kinds of prediction using Random Forest and XGBoost Regression and Random Forest and XGBoost Classification models (illustrated in Figure 13). Because each dimension of actions (Speed, Vertical, and Course) has a different range of values and can be affected by different features, we build separated models for each action. Six regression models and six classification models are developed to predict the value for each kind of action. However, for classification, two 3-actions models using Random Forest (RF) and XGBoost (XGB) are also developed to explore the potential of combining three models into one for action prediction. These models are considered for the completeness of our approach rather than the practical requirements, since in practice, the controller usually performs only one kind of action/decision in handling traffic.

(1) Regression models (RF, XGB):

Model RR1/XR1: predict ground speed rate  
 Model RR2/XR2: predict vertical speed  
 Model RR3/XR3: predict  $\Delta$  Course

We use  $R^2$  score as performance metric for this group of models.

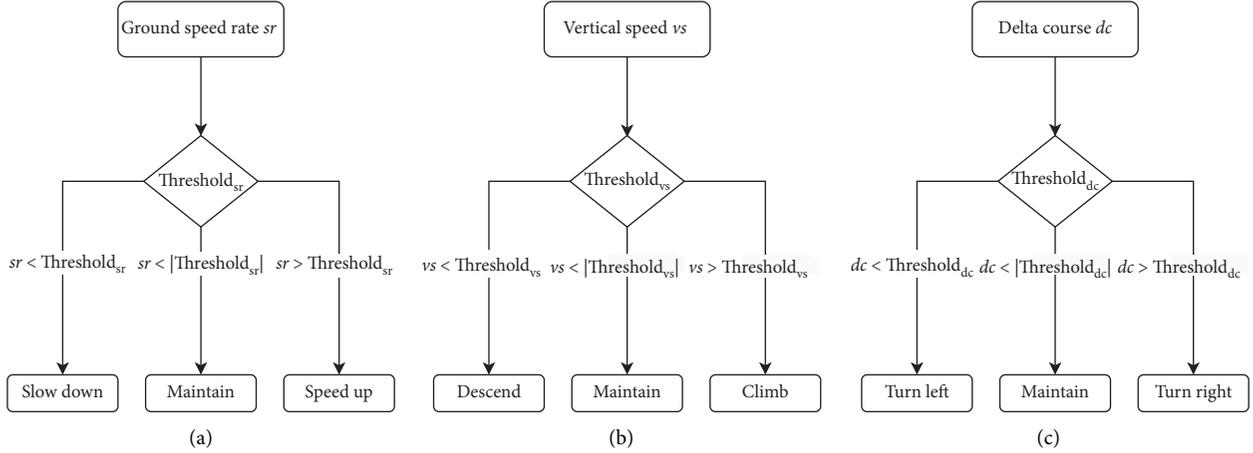


FIGURE 10: The illustration of the encoding step for extracting speed and vertical and course actions from continuous action values.

(2) Classification models (RF, XGB):

- Model RC1/XC1: predict speed action
- Model RC2/XC2: predict vertical action
- Model RC3/XC3: predict course action
- Model RC4/XC4: predict all 3-Actions.

We use Accuracy = Number of correct predictions/Total number of predictions as performance metric for this group of models.

## 6. Experiments and Results

As mentioned, we use RF and XGB methods to build both groups of predictive models. For both groups, we apply the same experiment setup:

The models are trained and tested with all 6-month of data.

Using flight information at the entry points as the input for the predictive model, the targets are extracted actions (mentioned in Table 4).

Parameter tuning: the range of hyperparameters for both types of models is manually selected to cover the optimal hyperparameter sets.

For regression models: number of estimators runs from 50 to 300 with the step is 50 (6 values) while max\_depth of forest varies from 6 to 58, step 2 (27 values).

For classification models: number of estimators runs from 50 to 300 with the step is 50 (6 values) while max\_depth of forest varies from 6 to 58 with the step is 2 (27 values).

The  $k$ -fold crossvalidation ( $k = 10$ ) technique is used for evaluating and the performance is averaged to select best set of parameters. It is popular for handling the overfitting issue in machine learning. The dataset is split into ten folds in which, for each fold, it is used for testing, and the remaining ones are used for training

models. The computed errors are averaged to estimate the model performance in general.

Experiment results for parameter tuning of RF regression models are shown in Figure 14. This heatmap shows the  $R^2 \pm$  Score for RF regression models with each pair of parameters. The lighter the color is, the higher the score of the model is; thus, the result indicates that the number of estimators ( $\geq 150$ ) does not affect the performance of models, instead, Max\_Depth plays a more important role. Max\_Depth  $\geq 10$  makes all three models stable with small variance. Another interesting observation is the significant improvement of Model RR3 ( $\Delta$ Course) when we increase the Max\_Depth of the trees from 6 to 20. It emphasizes the importance of this parameter tuning step for selecting suitable hyperparameters for our models. Thus, all RF Regression Models are trained with [Estimator = 150, Max\_Depth = 10]. Similar phenomena can be observed during the parameter tuning process of XGB Regression models. As a result, a common set of parameters [Estimator = 100, Max\_Depth = 8] is selected for all XGB regression models.

Table 5 presents the performance of regression models for each kind of action. Note that  $R^2$  score is best at 1 and worst at 0. All six models can predict action values for each given flight with a high value of  $R^2$  score. Model performance is reported for each month as well as the overall performance. The overall performance of models is a little higher in intermonsoon period (October and November) and is the highest in February which is the dry phase of northeast monsoon season. As we observe, the performance of models for December is lower than other months. Since December is the transition month in which winds and weather conditions are unstable, the accurate predictions are more challenging. Comparing models from two learning methods, we can see that XGB models outperform RF models generally for vertical speed and  $\Delta$ Course. While for ground speed rate, they vary with different datasets.

For models which are built on whole six-month dataset (all data), the models for ground speed rate have the lowest performance with the overall  $R^2$  scores as 0.667 for RF and 0.677 for XGB. The models achieve high performance (RF:

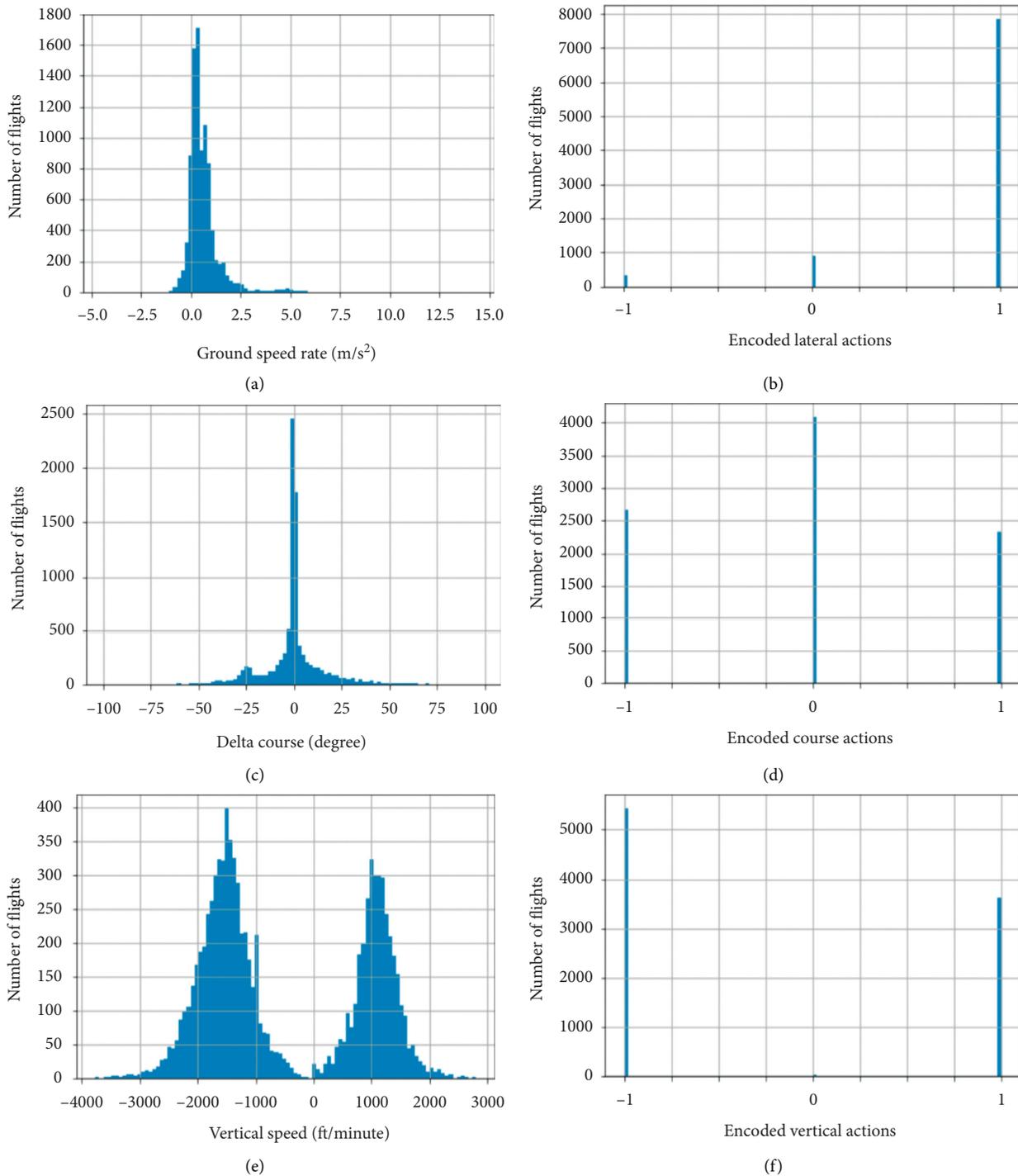


FIGURE 11: (a), (c), and (e) are distributions of computed trajectories' changes (actions) from ADS-B data. (b), (d), and (f) are distributions of encoded actions.

0.793, XGB: 0.858) for  $\Delta$  Course and (RF: 0.835, XGB: 0.870) for vertical speed. The XGB models have shown better performance in prediction comparing to RF models. Besides, as we can observe from Figure 15, the group of black dots in the bottom of the figure is close to each other, while their corresponding red dots are spatially diverse which is overcome by considering features on origin and destination of the flight.

As mentioned, the tree-based models can provide the feature importance information which reflects the contribution of each feature in those models. The list of top-10 features that have the most contribution for three RF regression models is shown in Table 6. The two features which have most contributions ( $> 66\%$ ) for predicting Ground Speed Rate (Model RR1) are Entry\_Speed and Destination\_WSSS. In Model RR2 (Vertical), there is only one

TABLE 4: Flight information (features) at entry point and actions and their values.

Features	Action values	Actions
Longitude	Ground speed rate	(1) speed up
Latitude	Vertical speed	(0) maintain speed
Altitude	$\Delta$ Course	(-1) slow down
Ground speed		(1) cimb
Vertical speed		(0) maintain level
Course		(-1) descent
Destination		(1) turn right
Origin		(0) maintain course
Month of year		(-1) turn left
Day of week		
Hour of day		
Flight ident		

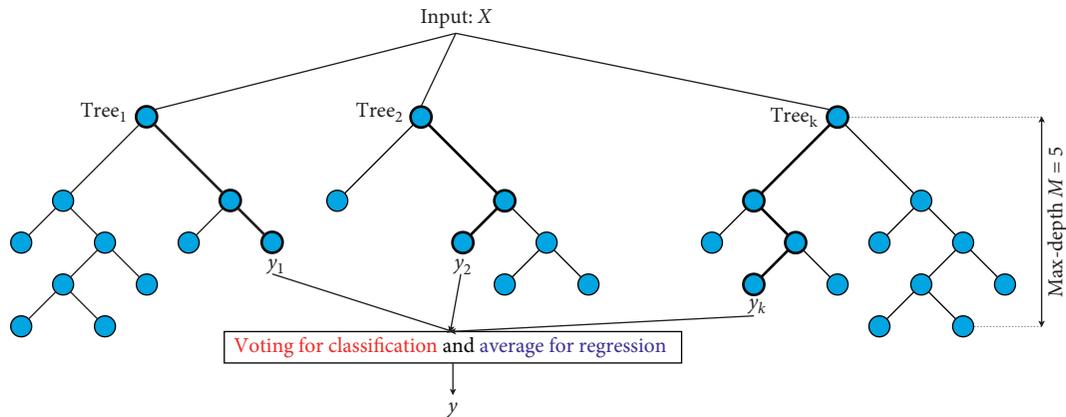


FIGURE 12: The illustration of tree-based ensemble learning method. The model contains  $k$  tree with maximum depth of each tree is  $M = 5$ . For each input vector  $X$  each tree will predict one target value. The predicted values of all the trees then are combined to the final prediction  $y$ .

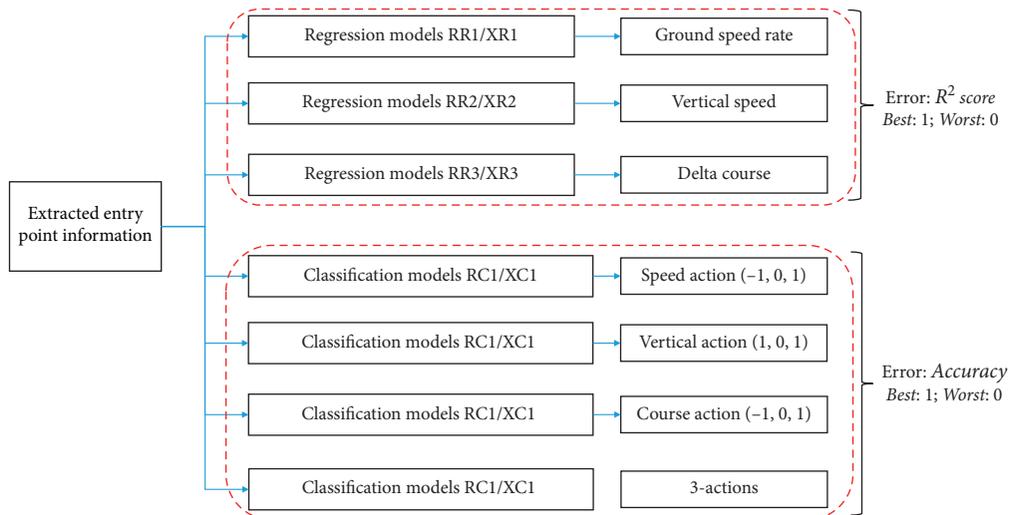


FIGURE 13: Illustration of developed predictive models: regression models of Random Forest (RR) and XGBoost (XR) and classification models of Random Forest (RC) and XGBoost (XC).  $R^2$  score is used as metric for regression models, while Accuracy is the metric for classification models.

dominant feature ( $> 82\%$ ) which is Destination\_WSSS. For Model RR3 (Course), the entry position and course of the flight have a significant impact on the course through the

selected sector ( $> 86\%$ ) in which course and longitude are more important than Latitude. Features for date and time information of the flight, such as Hour Of Day, Day of Week,

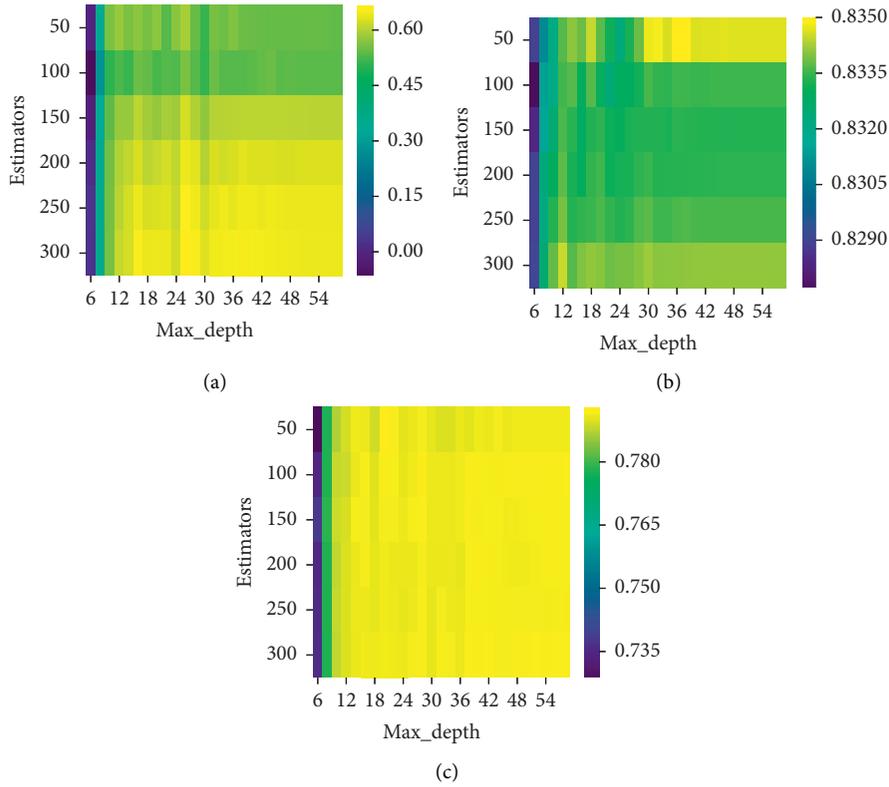


FIGURE 14: Parameter tuning for regression models. Estimators are [50 : 300, step 50] and Max\_Depth are [6 : 58, step 2]. (a) GSpeed Rate, (b) VSpeed, and (c)  $\Delta$ Course.

TABLE 5: Experiment results for predicting actions in  $R^2$  score.

Dataset	GSpeed rate		Vertical speed		$\Delta$ Course	
	RF	XGB	RF	XGB	RF	XGB
Sep-16	0.617 $\pm$ 0.037	<b>0.666</b> $\pm$ 0.046	0.838 $\pm$ 0.015	<b>0.859</b> $\pm$ 0.020	0.804 $\pm$ 0.042	<b>0.828</b> $\pm$ 0.049
Oct-16	<b>0.666</b> $\pm$ 0.042	0.622 $\pm$ 0.037	<b>0.846</b> $\pm$ 0.023	0.836 $\pm$ 0.023	0.758 $\pm$ 0.042	<b>0.796</b> $\pm$ 0.029
Nov-16	0.664 $\pm$ 0.048	<b>0.690</b> $\pm$ 0.055	<b>0.882</b> $\pm$ 0.013	0.859 $\pm$ 0.017	0.831 $\pm$ 0.029	<b>0.855</b> $\pm$ 0.020
Dec-16	0.614 $\pm$ 0.046	<b>0.637</b> $\pm$ 0.051	0.806 $\pm$ 0.021	<b>0.853</b> $\pm$ 0.013	<b>0.733</b> $\pm$ 0.030	0.724 $\pm$ 0.026
Jan-17	<b>0.649</b> $\pm$ 0.060	0.633 $\pm$ 0.030	0.828 $\pm$ 0.016	<b>0.873</b> $\pm$ 0.012	0.804 $\pm$ 0.021	<b>0.824</b> $\pm$ 0.023
Feb-17	<b>0.693</b> $\pm$ 0.048	0.654 $\pm$ 0.040	0.839 $\pm$ 0.016	<b>0.902</b> $\pm$ 0.013	<b>0.914</b> $\pm$ 0.043	0.877 $\pm$ 0.044
All data	0.667 $\pm$ 0.016	<b>0.677</b> $\pm$ 0.018	0.835 $\pm$ 0.006	<b>0.870</b> $\pm$ 0.007	0.793 $\pm$ 0.007	<b>0.858</b> $\pm$ 0.011

and Month of Year, have a small contribution in the trained models, which also implies that the season does not have a strong impact on our predictive models.

The parameter tuning for classification models show similar characteristics as regression models. An example of tuning process for RF models can be observed in Figure 16. For each value of the number of estimators, the model performance will vary and become stable with  $\text{Max\_Depth} \geq 24$ . The number of estimators ( $\geq 50$ ) does not affect the performance of classification models. Thus, all RF classification models are trained with [Estimator=50, Max\_Depth=24] and all XGB Classification Models are trained with [Estimator=250, Max\_Depth=8].

Table 7 shows the performances of classification models. Eight predictive models are developed and

evaluated in this study. Their performances are quite stable for different months. The overall performances are accessed by training these models with whole six-month dataset. As a result, the models for vertical actions provide the highest prediction accuracy ( $> 99.0\%$ ). Two models for speed action achieve approximately 80% in accuracy while the XGB model for course action can achieve up to 86.5% accuracy. The best model to predict the set of all the actions (altitude, speed, and course change) for each flight is the XGB model, and it achieves an accuracy 70% implying that, for 70% of flights,  $D$ -side controller's actions can be predicted from trajectory information at sector entry position. The 3-action model has lower performance comparing to the individual action model due to the increase in complexity of the output with 3 dimensions and 27 possible labels.

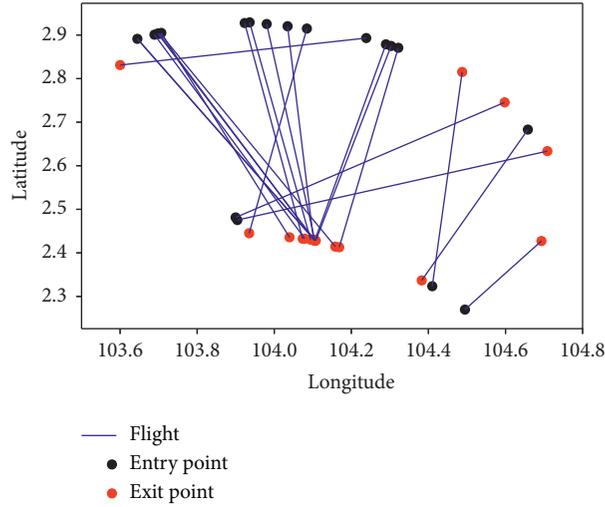


FIGURE 15: Example for flight direction. Black dots are entry points and red dots are exit points.

TABLE 6: Feature importance of random forest models for action value prediction.

Model RR1	FI_RR1 (%)	Model RR2	FI_RR2 (%)	Model RR3	FI_RR3 (%)
Speed	34.2	<i>D_WSSS</i>	82.7	Course	46.6
<i>D_WSSS</i>	32.1	<i>V_Speed</i>	5.2	Longitude	30.1
Altitude	6.3	Altitude	4.0	Latitude	10.4
<i>T_Remain</i>	5.0	<i>T_Remain</i>	2.7	<i>D_RPLL</i>	1.7
<i>V_Speed</i>	3.2	Speed	0.8	<i>T_Remain</i>	1.6
Longitude	2.9	Course	0.8	Altitude	1.4
Course	2.8	Longitude	0.6	<i>D_WSSS</i>	1.1
Latitude	2.1	Latitude	0.5	<i>V_Speed</i>	0.9
Hour	1.7	Hour	0.4	Speed	0.9
Month	1.5	Month	0.3	Hour	0.8

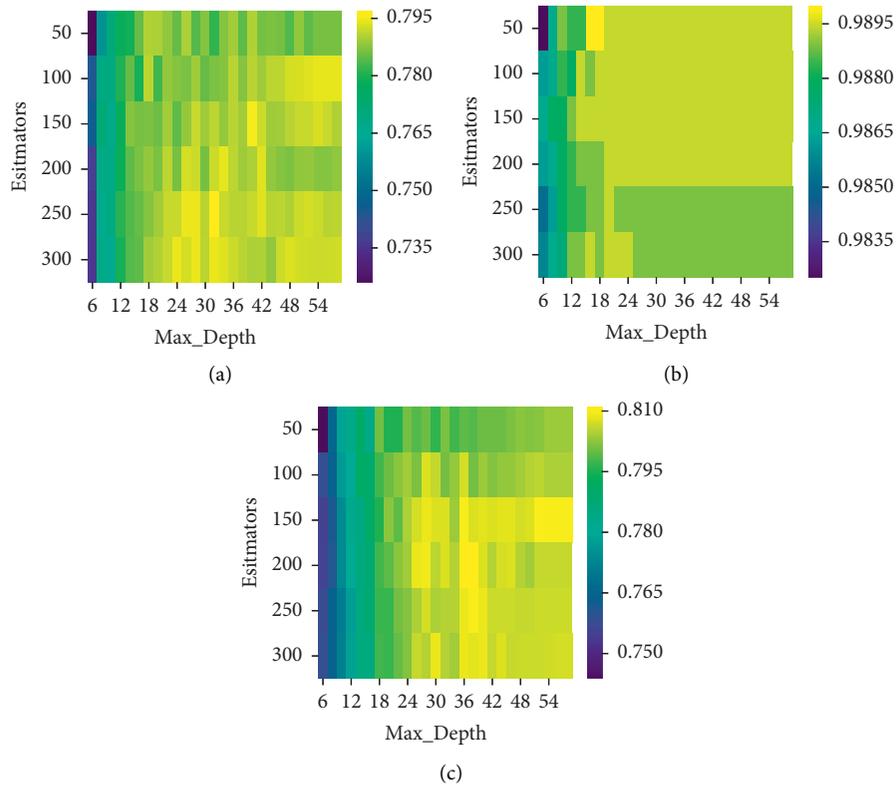


FIGURE 16: Illustration for parameter tuning process for classification models. Estimators are [50 : 300, step 50] and Max\_Depth are [6 : 58, step 2]. (a) Speed action, (b) vertical action, and (c) course action.

TABLE 7: Predictive accuracy of predictive models: speed action, vertical action, course action, and 3-action.

Dataset	S_Action		V_Action		C_Action		3-action	
	RF	XGB	RF	XGB	RF	XGB	RF	XGB
09-2016	75.7% ± 1.9%	80.3% ± 3.0%	99.3% ± 0.5%	98.9% ± 0.9%	80.4% ± 3.3%	87.1% ± 2.2%	66.0% ± 2.5%	71.0% ± 2.5%
10-2016	78.4% ± 1.6%	76.8% ± 2.2%	98.7% ± 0.5%	98.4% ± 0.6%	76.4% ± 2.3%	84.4% ± 2.2%	63.9% ± 2.5%	65.2% ± 2.3%
11-2016	81.8% ± 2.2%	82.7% ± 3.0%	99.2% ± 0.4%	99.7% ± 0.6%	77.9% ± 2.8%	85.2% ± 1.2%	68.7% ± 1.6%	70.6% ± 1.7%
12-2016	78.6% ± 1.5%	75.5% ± 2.5%	99.1% ± 0.4%	99.7% ± 0.6%	76.1% ± 2.7%	82.0% ± 1.4%	67.3% ± 1.8%	70.4% ± 2.3%
01-2017	78.7% ± 2.8%	73.6% ± 1.7%	99.3% ± 0.5%	99.0% ± 0.3%	80.3% ± 2.9%	84.2% ± 1.8%	68.3% ± 2.3%	63.8% ± 1.9%
02-2017	77.9% ± 1.8%	76.8% ± 1.8%	99.0% ± 0.5%	99.2% ± 0.3%	86.5% ± 2.0%	90.0% ± 1.8%	74.1% ± 1.5%	69.7% ± 1.7%
All data	79.7% ± 1.5%	78.5% ± 2.0%	99.0% ± 0.2%	99.2% ± 0.3%	81.1% ± 1.2%	86.5% ± 1.1%	67.2% ± 1.45%	69.5% ± 0.7%

TABLE 8: Feature importance of random forest models for action prediction.

RC1	FI1 (%)	RC2	FI2 (%)	RC3	FI3 (%)	RC4	FI4 (%)
Speed	14	Latitude	17	Course	19	Latitude	17
V_Speed	10	V_Speed	15	Longitude	16	Course	13
T_Remain	10	T_Remain	15	Latitude	14	Longitude	10
Latitude	8	O_WSSS	14	Altitude	8	T_Remain	9
Course	7	D_WSSS	8	T_Remain	5	V_Speed	9
Altitude	7	Course	4	Speed	4	Speed	7
Longitude	7	D_VTBS	3	V_Speed	4	Altitude	6
Month	4	Altitude	3	O_RKSI	2	D_WSSS	4
Day	4	Longitude	3	Month	2	O_WSSS	3
Hour	3	Hour	2	Day	2	Month	3

Though predictive models for individual action achieve good performance with high accuracy, the performance of 3-action models is presented for completeness of the results. It is highly unlikely that the controller will issue three actions together for the same flight. In terms of operational validation, the proposed approach is envisioned as ATCO assisting tool, not an autonomous tool. Thus, there is always ATCO discretion element, and as more ATCO actions are collected, the models can be further trained for better accuracy.

Table 8 shows feature importance for four RF classification models. Different from RF regression models where a few features have a significant contribution, the feature importance or features' contributions of all RF classification models are scattered. The feature importance of 3-action RF Model has highlighted the impact of features such as location, speed, course, and vertical speed of entry point, date-time information, and the relation of this flight with Changi Airport (WSSS) as its origin or destination.

This proposed learning mechanism also has an advantage for scalability. The two selected tree-based ensemble techniques are well known for their low computational cost with parallel computing option. Moreover, XGBoost is specially designed for handling big data. In terms of running time, the training time of the selected model varies from 40 to 50 seconds for the input with 25,000 samples. All the experiments are performed on a desktop with Processor Intel Xeron W-2123 CPU 3.6 Hz and RAM 32 GB.

## 7. Implementation of ATCO's Command

For the complex and stochastic environment of ATC, the tasks and required actions of planning controllers are

TABLE 9: Examples for basic ATCO's command for speed control, vectoring, and vertical speed.

Speed control
Maintain 350 knots or greater
Reduce speed 250 knots or less for sequencing
Increase speed mach .83 or greater for the next 5 minutes
Vectoring
Continue present heading
Fly heading (three digits)
Turn (left/right) heading (three digits)
Turn (left/right) (number of degrees) degrees
Vertical speed
PRE078 descend FL 120 at 1500 feet per minute or less
EKR365 climb FL 370 at 1000 feet per minute or greater

more than what are covered in the scope of this work. However, as we target the most fundamental actions of ATCO, the predicted results are useful in predicting the ATCO action to form appropriated commands. Some examples of ATCO's command for pilot regarding speed control, vectoring, and vertical speed are presented in Table 9.

The process to convert our results into similar ATCO's commands can be observed in the flow diagram in Figure 17. The aircraft information at the entry point such as airspeed, ground speed (or wind speed), heading, course (or wind vector), and flight level is necessary input to generate the command. By combining the input with the predicted actions, we can estimate the target values of airspeed, heading, and flight level for the given aircraft. Then, by using a list of predefined command templates, we can generate the expected command.

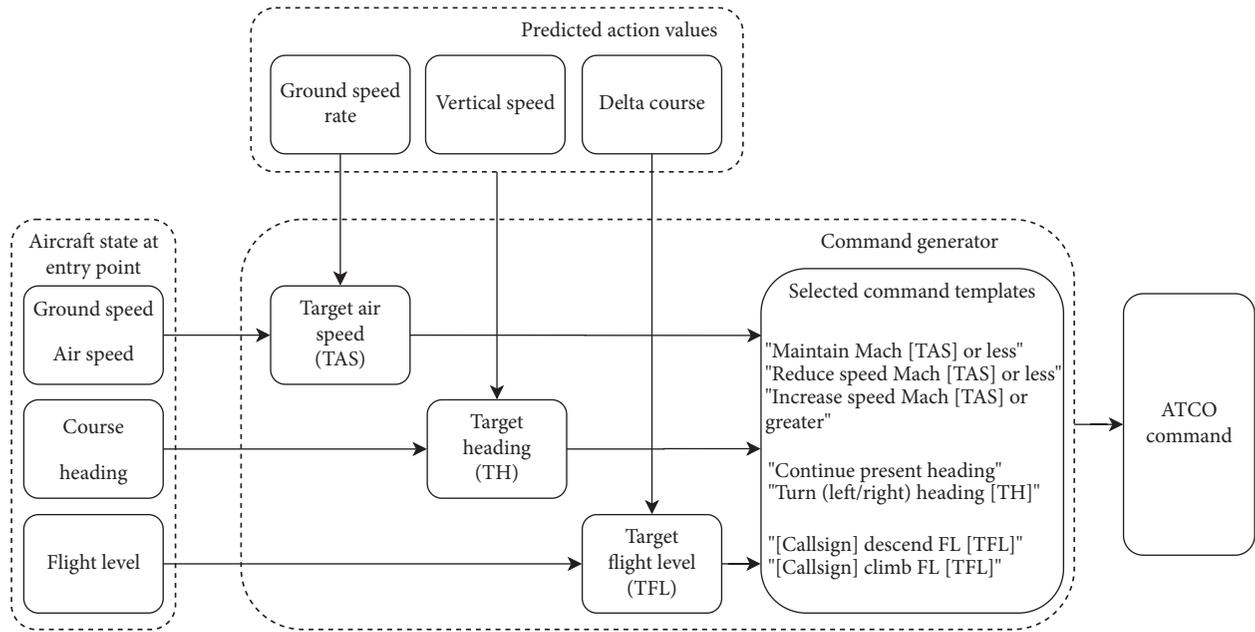


FIGURE 17: The flow diagram for converting from predicted results into ATCO's command.

## 8. Conclusions

In this paper, we have looked into learning and predicting the *D*-side controller's action for a given traffic scenario in a sector using two tree-based regression and classification method known as Random Forest and XGBoost. This learning problem was modeled as a classification problem where the target variable is *D*-side controller actions and the explanatory variables are the aircraft 4D trajectory features prior to entering a sector. The air traffic trajectories constructed through ADS-B data are analyzed spatial-temporal with sector data to establish that patterns in *D*-side controllers exist. Two groups of models were developed, one to predict the actions and the other to predict the associate action value. We used flight information at the sector entry point as the input for the predictive model and the targets are extracted actions. The model for vertical action provided the highest accuracy with 99%, whereas models for speed and course action provide predictability accuracy of 80% and 87%, respectively. This was attributed to highly complex sector entry and exit point configuration that makes learning challenging. The high prediction accuracy of altitude change maneuvers by the ATCO is significant as the airspace sector used in this research is a Transition Sector (Sector 2E in the Singapore ACC). The transition sector is typically where the flights are climbing to their cruise flight level or descending from the cruise flight level to terminal airspace. In such sectors, ATCO actions concerning altitude change are quite important, as there is a significant number of climbing and descending maneuvers happening in the sector. The model that predicts the set of all 3 kinds of actions (multiple-output) for each flight achieved an accuracy of 70%. This means that, for 70% flights, ATCO actions can be predicted

on all actions by using flight information at the sector entry position. The lower predictability can be due to an imbalance of the training data for the controller's actions, leading to poor generalization performance. The noise and low quality of ADS-B data can be another area of improvement as the model is as good as the data used to train it. Besides, as mentioned, this work only focuses on predicting simple actions from individual flight entry information.

The results and findings of this research can be used in several ways, in terms of application. For example,

The set of predicted maneuvers can be organized into patterns to derive and understand ATCO strategies, for traffic management, in a given sector

The results can also provide an insight into any biases ATCOs might have in managing traffic in a given sector, which can be addressed

For future work, we will apply clustering models to identify clusters of loss of separation events in spatial-temporal plane. Detected events can be used to evaluate the model performance in term of conflict solving. Then, some clustering methods will be applied over whole trajectories to detect and extract more complex actions from ADS-B data. Finally, more features from sector and traffic will be considered for learning and predicting to improve the practical of those models. Expanding the feature set by including parameters such as weather and wind could be useful; however, in the absence of meteorological information at that flight level makes that challenging. However, we will try to approximate the wind data by extrapolating the surface winds (available from METAR Data) to a higher altitude as our future work. We will also conduct Human in the Loop simulation to measure the effectiveness of the proposed

algorithm as well as comparing it with traditional tools used by Air Traffic Controllers for Aircraft Trajectory Prediction to better understand the predictability of ATCO's actions.

There can be several practical implications and difficulties in implementing the results of the research. Even though, the proposed approach is envisioned as ATCO assisting tool, not an autonomous tool. The biggest challenge is the acceptance of such machine learning-based tool by the controllers and the safety regulators. The design of such a tool will be also a challenge in an already overcrowded ATCO's working environment. There are issues of trust with such automation/advisory systems, where ATCOs do not fully accept the proposed solutions/advisories generated by the system. Thus, there will be a need of a human factor study and safety validation scenarios, to realize the benefits of such research. Eventually, this research may help develop an AI Agent, which can augment a D-side controller to manage and coordinate strategic traffic flow within and beyond sectors, thereby managing their workload.

## Nomenclature

ACC:	Area control center
ADS – B:	Automatic-dependent surveillance-broadcast
AIP:	Aeronautical information publication
ATC:	Air traffic control
ATCO(s):	Air traffic controller (s)
ATM:	Air traffic management
ATS:	Air traffic service
FIR:	Flight information region
LOS:	Loss of separation
METAR:	Meteorological aerodrome reports
MTCDD:	Medium term conflict detection
RF:	Random forest (machine learning technique)
STCA:	Short term conflict alert
TMA:	Terminal control area
XGB:	Extreme gradient boosting or xgboost (machine learning technique).

## Data Availability

The used ADS-B data, or flight trajectories, belong to ATMRI and NTU, with a restriction on publicly sharing the dataset.

## Disclosure

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not reflect the views of National Research Foundation, Singapore, and the Civil Aviation Authority of Singapore. Part of this research is published as a PhD thesis of first author.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This research was supported by the National Research Foundation, Singapore, and the Civil Aviation Authority of Singapore, under the Aviation Transformation Programme.

## References

- [1] ICAO, *Pans, Air Traffic Management. Doc-4444*, ICAO, Montreal, Canada, 2007.
- [2] M. Nolan, *Fundamentals of Air Traffic Control*, Cengage Learning, Boston, MA, USA, 2010.
- [3] D. McNally, H. Erzberger, R. Bach, and W. Chan, "A controller tool for transition airspace," in *Proceedings of the Guidance, Navigation, and Control Conference and Exhibit*, p. 4298, Portland, OR, USA, August 1999.
- [4] S. Alam, A. Ha, C. J. Lokan, M. Ellejmi, and S. Kirby, "Computational red teaming to investigate failure patterns in medium term conflict detection," in *Proceedings of the 8th Eurocontrol Innovation Research Workshop, Eurocontrol Experimental Center*, Brtigny-sur-Orge, France, December 2009.
- [5] A. Shakarian and A. Haraldsdottir, "Required total system performance and results of a short term conflict alert simulation study," in *Proceedings of the 4th US/Europe Air Traffic Management R & D Seminar*, pp. 1–9, Sante Fe, NM, USA, December 2001.
- [6] M. Strohmeier, M. Schafer, V. Lenders, and I. Martinovic, "Realities and challenges of nextgen air traffic management: the case of ads-b," *IEEE Communications Magazine*, vol. 52, no. 5, pp. 111–118, 2014.
- [7] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning*, Vol. 112, Springer, Berlin, Germany, 2013.
- [8] J. Friedman, T. Hastie, and R. Tibshirani, *The elements of Statistical Learning*, Vol. 1, Springer Series in Statistics, New York, NY, USA, 2001.
- [9] D. B. Fogel, T. J. Hays, S. L. Hahn, and Q. James, "A self-learning evolutionary chess program," *Proceedings of the IEEE*, vol. 92, no. 12, pp. 1947–1954, 2004.
- [10] D. Cannel and S. Markovitch, "Learning Models of Opponent's Strategy Game Playing," in *Proceedings of the 1993 AAAI Fall Symposium on Games: Learning and Planning*, pp. 140–147, AAAI Press Technical Report FS93-02, Menlo Park, CA, USA, 1993.
- [11] J. B. Pollack and A. D. Blair, "Co-evolution in the successful learning of backgammon strategy," *Machine Learning*, vol. 32, no. 3, pp. 225–240, 1998.
- [12] J. Fürnkranz, "Machine learning in computer chess: the next generation," *ICGA Journal*, vol. 19, no. 3, pp. 147–161, 1996.
- [13] D. Silver, A. Huang, C. J. Maddison et al., "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.
- [14] T. Dyster, S. A. Sheth, and G. M. McKhann, "Ready or not, here we go," *Neurosurgery*, vol. 78, no. 6, pp. N11–N12, 2016.
- [15] H. Sathy, A. Patel, and V. Padmanabhan, "Real time strategy games: a reinforcement learning approach," *Procedia Computer Science*, vol. 54, pp. 257–264, 2015.
- [16] D. Bloembergen, K. Tuyls, D. Hennes, and M. Kaisers, "Evolutionary dynamics of multi-agent learning: a survey," *Journal of Artificial Intelligence Research*, vol. 53, pp. 659–697, 2015.
- [17] K. Weiss, T. M. Khoshgoftaar, and D. D. Wang, "A survey of transfer learning," *Journal of Big Data*, vol. 3, no. 1, p. 9, 2016.

- [18] A. R. Odoni, "The flow management problem in air traffic control," in *Flow Control of Congested Networks*, pp. 269–288, Springer, Berlin, Germany, 1987.
- [19] L. Pallottino, E. M. Feron, and A. Bicchi, "Conflict resolution problems for air traffic management systems solved with mixed integer programming," *IEEE Transactions on Intelligent Transportation Systems*, vol. 3, no. 1, pp. 3–11, 2002.
- [20] D. Gianazza, "Learning air traffic controller workload from past sector operations," in *Proceedings of the ATM Seminar, 12th USA/Europe Air Traffic Management R&D Seminar*, Seattle, WA, USA, June 2017.
- [21] T. Wu and W. Du, "A distributed approach to aircraft conflict resolution based on satisficing game theory," in *Advances in Intelligent Systems and Computing, Foundations of Intelligent Systems*, pp. 383–393, Springer, Berlin, Germany, 2014.
- [22] V. B. Kulkarni, "Intelligent air traffic controller simulation using artificial neural networks," in *Proceedings of the 2015 International Conference on Industrial Instrumentation and Control (ICIC)*, pp. 1027–1031, IEEE, Pune, India, May 2015.
- [23] S. Ravizza, J. Chen, J. A. D. Atkin, P. Stewart, and E. K. Burke, "Aircraft taxi time prediction: comparisons and insights," *Applied Soft Computing*, vol. 14, pp. 397–406, 2014.
- [24] H. Lee, W. Malik, and Y. C. Jung, "Taxi-out time prediction for departures at charlotte airport using machine learning techniques," in *Proceedings of the 16th AIAA Aviation Technology, Integration, and Operations Conference*, p. 3910, Washington, DC, USA, June 2016.
- [25] M. Ahmed, S. Alam, and M. Barlow, "A cooperative co-evolutionary optimisation model for best-fit aircraft sequence and feasible runway configuration in a multi-runway airport," *Aerospace*, vol. 5, no. 3, p. 85, 2018.
- [26] R. Alligier, D. Gianazza, and N. Durand, "Machine learning and mass estimation methods for ground-based aircraft climb prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 6, pp. 3138–3149, 2015.
- [27] M. C. R. Murca, R. J. H. Richard DeLaura, R. Jordan, T. Reynolds, and H. Balakrishnan, "Trajectory clustering and classification for characterization of air traffic flows," in *Proceedings of the 16th AIAA Aviation Technology, Integration, and Operations Conference*, p. 3760, Washington, DC, USA, June 2016.
- [28] N. Takeichi, R. Kaida, A. Shimomura, and T. Yamauchi, "Prediction of delay due to air traffic control by machine learning," in *Proceedings of the AIAA Modeling and Simulation Technologies Conference*, p. 1323, Grapevine, TX, USA, January 2017.
- [29] C. Sun, Y. J. Kim, B. Simon, and D. Mavris, "Prediction of weather-induced airline delays based on machine learning algorithms," in *Proceedings of the 2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*, pp. 1–6, IEEE, Sacramento, CA, USA, September 2016.
- [30] H. Georgiou, S. Karagiorgou, Y. Kontoulis et al., "Moving objects analytics: survey on future location & trajectory prediction methods," 2018, <https://arxiv.org/abs/1807.04639>.
- [31] R. Wu, G. Luo, J. Shao, L. Tian, and C. Peng, "Location prediction on trajectory data: a review," *Big Data Mining and Analytics*, vol. 1, no. 2, pp. 108–127, 2018.
- [32] Y. Zhang and A. Haghani, "A gradient boosting method to improve travel time prediction," *Transportation Research Part C: Emerging Technologies*, vol. 58, pp. 308–324, 2015.
- [33] S. Papadopoulos, E. Azar, W.-L. Woon, and C. E. Kontokosta, "Evaluation of tree-based ensemble learning algorithms for building energy performance estimation," *Journal of Building Performance Simulation*, vol. 11, no. 3, pp. 322–332, 2018.
- [34] Y. Li and Z. Yang, "Application of eos-elm with binary jaya-based feature selection to real-time transient stability assessment using PMU data," *IEEE Access*, vol. 5, pp. 23092–23101, 2017.
- [35] A. Pescape, A. Montieri, G. Aceto, and D. Ciunzo, "Anonymity services tor, i2p, jondonym: classifying in the dark (web)," *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 3, pp. 662–675, 2018.
- [36] V. F. Taylor, R. Spolaor, M. Conti, and I. Martinovic, "Appscanner: automatic fingerprinting of smartphone apps from encrypted network traffic," in *Proceedings of the 2016 IEEE European Symposium on Security and Privacy (EuroSec&P)*, pp. 439–454, IEEE, Saarbrücken, Germany, March 2016.
- [37] G. Aceto, D. Ciunzo, A. Montieri, and A. Pescapé, "Multi-classification approaches for classifying mobile app traffic," *Journal of Network and Computer Applications*, vol. 103, pp. 131–145, 2018.
- [38] J. J. Rebollo and H. Balakrishnan, "Characterization and prediction of air traffic delays," *Transportation Research Part C: Emerging Technologies*, vol. 44, pp. 231–241, 2014.
- [39] P. Monmousseau, D. Delahaye, A. Marzuoli, and E. Feron, "Predicting and analyzing us air traffic delays using passenger-centric data-sources," in *Proceedings of the Thirteenth USA/Europe Air Traffic Management Research and Development Seminar (ATM2019)*, Vienna, Austria, June, 2019.
- [40] R. Caruana and A. Niculescu-Mizil, "An empirical comparison of supervised learning algorithms," in *Proceedings of the 23rd International Conference on Machine Learning*, pp. 161–168, Pittsburgh, PA, USA, June 2006.
- [41] X. Zhang and S. Mahadevan, "Ensemble machine learning models for aviation incident risk prediction," *Decision Support Systems*, vol. 116, pp. 48–63, 2019.
- [42] A.-D. Nguyen, D.-T. Pham, N. Lillith, and S. Alam, "Model generalization in arrival runway occupancy time prediction by feature equivalences," in *Proceedings of the 9th International Conference for Research in Air Transportation (ICRAT)*, Tampa, FL, USA, September 2020.
- [43] G. F. Jenks and F. C. Caspall, "Error on choroplethic maps: definition, measurement, reduction," *Annals of the Association of American Geographers*, vol. 61, no. 2, pp. 217–244, 1971.
- [44] S. Alam, C. Lokan, and H. Abbass, "What can make an airspace unsafe? characterizing collision risk using multi-objective optimization," in *Proceedings of the 2012 IEEE Congress on Evolutionary Computation*, pp. 1–8, IEEE, Brisbane, Australia, June 2012.
- [45] A. Liaw and M. Wiener, "Classification and regression by randomforest," *R News*, vol. 2, no. 3, pp. 18–22, 2002.
- [46] T. Chen, H. Tong, M. Benesty, V. Khotilovich, and Y. Tang, *Xgboost: Extreme Gradient Boosting. R Package version 0.4-2*, NYC Data Science Academy, New York, NY, USA, 2015.
- [47] T. Chen and C. Guestrin, "Xgboost: a scalable tree boosting system," in *Proceedings of the 22nd Acm Sigkdd International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, San Francisco, CA, USA, August 2016.
- [48] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001.
- [49] P. Li, "Robust logitboost and adaptive base class (ABC) logitboost," 2012, <https://arxiv.org/abs/1203.3491>.
- [50] D.-T. Pham, *Machine learning-based flight trajectories prediction and air traffic conflict resolution advisory*, PhD Thesis, PSL Research University, Paris, France, 2019.