Data-driven Runway Occupancy Time Prediction using Decision Trees

Hong Wei Chow*, Zhi Jun Lim†, Sameer Alam‡
Air Traffic Management Research Institute, School of Mechanical and Aerospace Engineering
Nanyang Technological University, Singapore
Email: { *chow0143 | †zhijun001, | ‡sameeralam}@ntu.edu.sg

Abstract—**With an increasing amount of flights, the demand for runways at airports increases as well. Innovative mechanisms are required to maximise the use of a runway such that the demand can be met. Such mechanisms include the prediction of Runway Occupancy Time (ROT), so that the Air Traffic Controllers (ATCs) are able to gauge how much time a particular flight needs on the runway. This allows them to prepare the next flight for the runway and effectively reduce the buffer times between flights, thus increasing the overall efficiency of the runway. The objective of this paper is to develop an explainable machine learning model to predict Runway Occupancy Time. The Decision Tree Regressor was chosen for this study and its performance was compared to other more complicated machine learning models. The Decision Tree Regressor, unlike the other machine learning algorithms, provides explicit rules on how the predictions of the ROT is derived. An example of a generated rule for runway 02L of Singapore Changi Airport is that if an aircraft is a medium aircraft from airline XXX, arriving between** 2100 **and** 2159 **hours UTC, with an approach speed of more than** 83.344 **m/s at the final approach fix, and with the trailing aircraft traveling slower, the predicted ROT will be** 42.6 **seconds. Results show that the Decision Tree Regressor has the least runtime out of all the models at** 0.28 **minutes during training and its prediction capabilities are also on par with the rest of the machine learning models. The Root Mean Square Error for the Decision Tree Regressor is** 5.96 **seconds, which is only** 0.20 **seconds away from the best performing machine learning model. This, coupled with the rules that the Decision Tree Regressor can provide, makes it easier for end-users to to accept the prediction results without compromising on the accuracy. Permutation importance was also applied to the decision tree, providing an insight into what affects the ROT the most.**

I. INTRODUCTION

As the world becomes more globalized, the demand for international air travel increases. As more flights are in operation and the number of takeoffs and landings grow, runways become a scarce resource. Simultaneous runway occupancy for arrival and departure is not permitted due to safety reasons. Air Traffic Controllers (ATCs) will need to have the runway vacated before they can clear the next flight for takeoff or landing. Buffer times in between these flights can add up and may significantly impact the throughput of the runway. The importance and dependency of Runway Occupancy Time (ROT) has always been regarded as a crucial factor in relation to runway occupational capacity [1]. This pushes the need for efficient use of runways. By being able to reasonably predict the ROT of aircrafts, ATCs can estimate when the next flight will be able to utilize the runway, regardless of arrival or departure.

Recent studies have explored the idea of using machine learning models to predict ROT. Aircraft features, weather data, as well as airport data have been used to train these models [2], [3]. While the models have been able to predict the ROT with an acceptable level of accuracy, very few of them can explain how and why a specific ROT outcome is achieved. Since there is no clear way for ATCs to see how the prediction is achieved, the confidence level in the prediction capability of machine learning algorithms might be reduced. Therefore, it is very desirable to develop an explainable prediction model that can predict ROT with an acceptable level of accuracy. The objective of this paper is to develop a machine learning model that is capable of that. Fig. 1 illustrates the problem that we are tackling in this paper which is to predict the ROT of the aircraft when the aircraft is at final approach fix. The machine learning algorithm chosen is the Decision Tree Regressor as it can be easily explainable. The Decision Tree Regressor was then compared to other more sophisticated machine learning algorithms such as Random Forest Regressor, CatBoost Regressor, Multilayer Perceptron Neural Network. These algorithms should give better prediction results but at the expense of explainability.

The rest of the paper is organized as follows. Section II presents some work that has been done regarding ROT, as well as machine learning algorithms that are trained on ROT prediction. Next, the problem formulation is described in section III along with the proposed framework which includes the data source, feature engineering and machine learning models. Section IV is devoted for data analysis. Following that, section V reports the experiments and the results of our study. Finally, section VI concludes the paper.

II. RELATED WORK

Studies have been made relating ROT and different factors affecting it such as wake turbulence, the airline, the airport, different exit systems for the runway etc. Over time, these studies utilized machine learning to come up with models that can predict the ROT under different conditions. The models develop their predictive powers using historical data.

One of the first studies relating to ROT was done by Koenig [4] in 1978. Koenig was interested in how the behavior of airlines and their pilots could lead to a higher ROT, thus

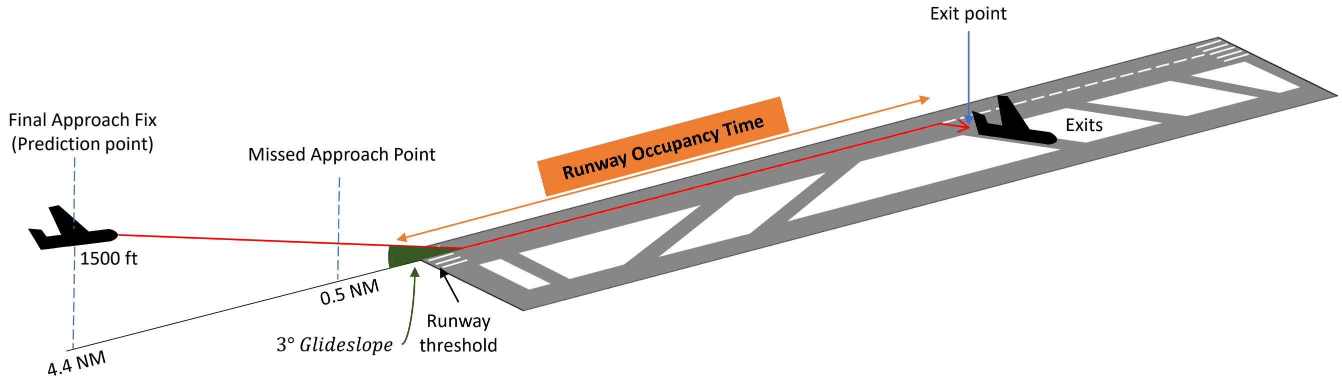

Fig. 1. Illustration of the prediction of ROT at final approach fix

delaying flights and reducing the overall throughput of the runway. The data used in this study was visually collected from six different airports in 1972 and 1973. The author observed that pilot motivational factors was a huge influence in determining the ROT. Majority of the pilots were found to exit the runway using exits that were most convenient for them to get to the location of their gate.

In 1984, Weiss and Barrer [5] analyzed the relation between ROT, the minimum longitudinal separation, and the condition of the runway (dry/wet). They had to visually estimate the ROT of aircrafts from the control tower at LaGuardia airport, Newark airport, as well as Boston Logan International airport. They concluded that the runway condition did not significantly affect the ROT. In addition, ATCs were found to increase the longitudinal separation. This lowers the overall efficiency of the runway and more could be done to increase the runway throughput. However, the data collection for ROT was unreliable as they had to estimate it from the control tower. Furthermore, the reference point between the observers during the estimation were different.

In a more recent work [6], Lim et al. explored the causal effects of landing parameters, such as the final approach speed and deviation from glideslope, on ROT. This work shifted the focus from data-driven prediction to data-driven decision. The results from their experiments on arrival flights landing on runway 02L of Singapore Changi Airport were able to provide intervention policies on the landing parameters that can reduce ROT.

III. RESEARCH FRAMEWORK

A. Problem Formulation

While there are prediction models in the literature that predict the ROT with a good accuracy, these models are mostly treated as black boxes by the end-users as they do not provide explanation on how the results are achieved. Thus, the objective of this paper is to develop an explainable ROT prediction model with reasonable prediction performance.

The scope of this research is limited to arrival flights landing on runway 02L of Singapore Changi Airport. The prediction horizon is set at 4.4 NM away from the runway 02L threshold (i.e., final approach fix (FAF)). This prediction horizon allows the Air Traffic Controller to make the necessary changes to the scheduling if required as the FAF is 3.9 NM away from the missed approach point as shown in Fig. 1. To achieve the research objective, we require trajectory data of arrival flights which start from at least 5 NM away from the runway 02L threshold and only end after the flights have exited the runway. Besides that, meteorological information is also required as it is also crucial for the prediction of ROT.

B. Proposed framework

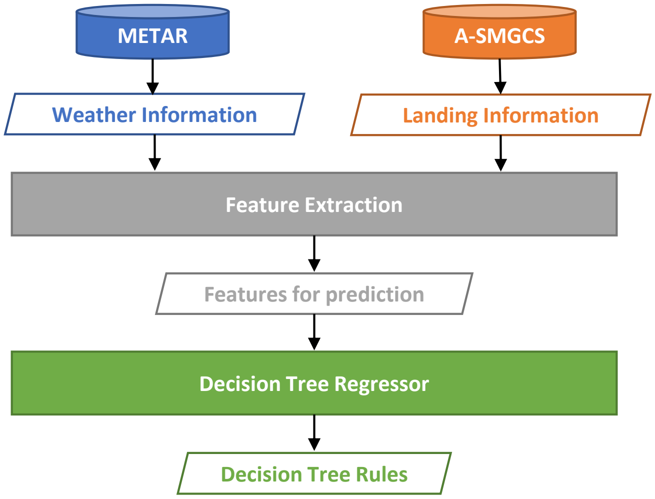

Fig. 2. Proposed framework for runway occupancy time prediction

Therefore, the Decision Tree Regressor was chosen as our primary machine learning model to predict the ROT at Changi Airport runway 02L as it provides the explainability and rules for the outcome. The proposed framework can be seen in Fig. 2. Firstly, weather and landing information from the Meteorological Aerodrome Report (METAR) and Advanced Surface Movement Guidance Control System (A-SMGCS) data will be processed. Secondly, extraction of useful features will then be performed. Lastly, the features will be fed to the Decision Tree Regressor. The model will output the predicted ROT, the different rules, and how the final outcome is achieved.

To evaluate the performance of the Decision Tree prediction model, other state-of-the-art prediction models were also explored. Root mean squared error (RMSE) and the mean absolute error (MAE) were used to compare the prediction performance of these models.

C. Data source and Data cleaning

A-SMGCS data provides the air traffic data of the flights. Information such as the 4-D trajectory (i.e., longitude, latitude, altitude, timestamp), ground speed, International Civil Aviation Organisation (ICAO) wake turbulence category, aircraft model and airline of each flight is provided. Furthermore, unlike Automatic Dependent Surveillance–Broadcast (ADS-B) data, data interpolation is not necessary as A-SMGCS data records information every second. For this research, one month (i.e., October 2017) of air traffic data for Singapore Changi Airport (WSSS) was utilised and there were a total of 5386 arrival flights landing on runway 02L.

In addition to trajectory data, meteorological information is also important for the prediction of runway occupancy time. The meteorological information were obtained from historical METARs.

Data cleaning is an essential step as it helps to remove outliers and improve the quality of the training data. Outliers are data points that are far away from the majority of the points. Thus, the prediction model may result in inaccurate predictions if these outliers are included during training. The standard deviation method was employed to remove outliers with respect to the runway occupancy time. Flights with ROT that were three standard deviations away from the mean ROT were regarded as outliers and were removed from the dataset. A total of 68 outliers were removed and we were left with 5318 flights.

D. Features Engineering

ROT is the time taken for an aircraft to vacate the runway completely after crossing the runway threshold during landing [7]. The coordinates of the runway and exit polygons were defined using Google Earth Pro. We captured the timestamps of the aircraft when it entered the runway polygon and when it entered any of the exit polygons and the difference between the two timestamps is the ROT.

Factors that affected ROT had to be extracted from the data to be used in the machine learning models. These factors were explored in the studies in section II. First, only flights departing from other airports and landing on runway 02L of WSSS were considered. Feature extraction is then performed on these flights. Features were considered based on works in [2], [3], [6] which are verified as relevant factors affecting ROT. We also wanted to analyze the impact of trailing aircraft on the ROT of leading aircraft, as well as the hour of the day, and day of the week. Furthermore, we utilised the Air traffic management airport performance (ATMAP) weather algorithm [8] developed by Eurocontrol to represent the severity of the weather in the aerodrome with a single index. There are 5 weather classes used to calculate the ATMAP score: Visibility & Ceiling, Precipitations, Freezing Conditions, Wind, and Dangerous Phenomena. A severity code is assigned to each weather class based on the METAR, and a coefficient is assigned to the severity code. The coefficients are summed up to give a score for that particular METAR report. The details of the extracted features can be found in Table I.

E. Machine Learning models

This paper aims to develop an explainable machine learning model that is capable of predicting ROT. We have chosen the Decision Tree Regressor as it fulfils that requirement. Our model will then be assessed and compared to more complicated machine learning models, such as: Random Forest Regressor, Catboost Regressor, and Multilayer Perceptron Neural Network. A short description of the models can be found below:

1) Decision Tree Regressor: The Decision Tree Regressor is a decision support tool that uses a tree-like model to display the different decisions and possible outcomes. Unlike the Decision Tree Classifier, this model is used when the dependent variable is continuous. Features will have to be encoded first before they can be used by the model. The tree decides on a split based on the mean squared error of the features. It will select the lowest value and use that feature as a splitting criteria.

2) Random Forest Regressor: Much like the Decision Tree Regressor, the Random Forest Regressor requires features to be encoded. It also uses mean squared error as a splitting criteria. The difference between the two models is that the Random Forest Regressor grows a group of trees randomly to predict the variable instead of just one tree [9]. Each tree in the Random Forest Regressor is weak by itself, but when they are used together, they provide good predictive capabilities. Over-fitting and the accuracy of the prediction is controlled by averaging the prediction values.

3) CatBoost Regressor: Catboost Regressor is a machine learning algorithm developed by a company named Yandex. While the previous two models require encoded data to work, CatBoost Regressor does not. The model can handle categorical data as a feature for its learning process [10]. The Catboost algorithm draws inspiration from decision trees and gradient boosting. Boosting takes many weak models and combines them one at a time to produce a strong predictive model. We used RMSE as the criteria in the model due to the nature of our variable.

4) Multilayer Perceptron Neural Network: Artificial Neural Networks (ANN) are models that mimic a human brain. The models try to simulate the way the brain stores and process information. The Multilayer Perceptron Neural Network is an example of such a model. It is a feedforward ANN [11], whereby the connections between the nodes do not form a cycle. The Multilayer Perceptron Neural Network is made up of at least three layers: the input layer, the hidden layer, and

TABLE I

FEATURES EXTRACTED

Feature Name	**Description**
Overall_Velocity_FAF	Velocity of the aircraft at the Final Approach Fix.
atmap_total	ATMAP score which serves as a proxy for the severity of weather.
Headwind	Speed of wind blowing against the landing aircraft (measured in knots). Positive value means wind direction is against aircraft, negative value means wind direction is along aircraft .
Crosswind	Speed of wind blowing perpendicular to landing aircraft (measured in knots). Positive value means wind is pushing aircraft to the left, negative value means wind is pushing aircraft to the right.
wake_type	ICAO wake turbulence category: Medium (M), Heavy (H)
Airline	Airline for the corresponding flight.
Aircraft_model	Model of the aircraft.
Speed_case	Determines if the trailing aircraft is travelling faster (Close) or slower (Open) than the aircraft of interest. NFC indicates that there is no following aircraft.
DoW	Day of the week. Values range from 0 (Monday) to 6 (Sunday).
HoD	Hour of the Day. Values range from 0 (0000Z) to 23 (23000Z)
Terminal	Expected terminal where aircraft will stop and allow for disembarkation. Values 1 - 4 represent Terminal 1 - 4 respectively, 5 represents Cargo Terminal.

the output layer. The inputs are attached to a weight and the models learns by adjusting the weights to reduce the difference between the predicted and actual outputs.

IV. DATA ANALYSIS

Data analysis was conducted so as to obtain the statistical information of the runway occupancy time of flights landing on runway 02L in Singapore Changi Airport. There are a total of 5318 flights after data cleaning. 2739 of the flights are medium aircraft and remaining 2579 flights are heavy aircraft. Table II tabulates the mean and standard deviation of Runway Occupancy Time for all aircraft, medium aircraft and heavy aircraft respectively.

TABLE II

MEAN AND STANDARD DEVIATION OF RUNWAY OCCUPANCY TIME (IN SECONDS) FOR ALL FLIGHTS, MEDIUM FLIGHTS AND HEAVY FLIGHTS

	All	Medium	Heavy
Mean (s)	52.37	50.58	54.27
Std. dev. (s)	6.53	5.28	7.16

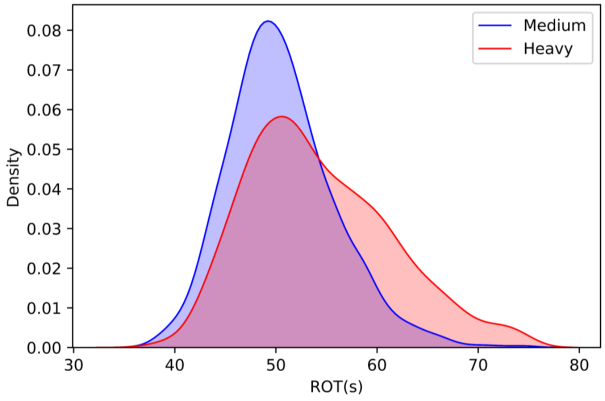

Fig. 3. ROT distributions of medium and heavy aircraft

The average runway occupancy time of all flights landing on runway 02L in October 2017 is 52.37 seconds. Furthermore, the standard deviation of the runway occupancy time is about 6.53 seconds. The standard deviation represents the variation in the runway occupancy times. It not only limits the performance of the prediction model, it also serves as a benchmark for the prediction performance. As we dug deeper into the runway occupancy time of the two different wake categories, we found out that, on average, the medium aircraft have a shorter ROT (i.e., 50.58 seconds) as compared to that of the heavy aircraft (i.e, 54.27 seconds). Furthermore, there is a larger variation (i.e., 7.16 seconds) in the ROT of the heavy aircraft as compared to that of the medium aircraft (i.e., 5.28 seconds). Fig. 3 displays the ROT distributions for the heavy and medium aircraft respectively and it shows that the ROT distribution of the heavy aircraft has a wider spread than that of the medium aircraft.

TABLE III

STATISTICS OF THE UTILISATION OF EACH RUNWAY EXIT BY MEDIUM AND HEAVY AIRCRAFT

	Medium		Heavy	
	Number of flights	Percentage of flights (%)	Number of flights	Percentage of flights (%)
Exit 1	2621	95.69	1709	66.27
Exit 2	118	4.31	831	32.22
Exit 3	0	0	39	1.51

The wider distribution of the heavy aircraft is due to the different runway exits used by the medium and heavy aircraft as shown in Table III. For runway 02L, there are three high-speed exits and the heavy aircraft utilised all of the three high-speed runway exits while the medium aircraft only utilised the first two exits. One reason for such observation is that the cargo terminal is situated nearest to exit 3 and heavy aircrafts made up the bulk, 82.3%, of the cargo flights landing on runway 02L. Thus, the pilots of cargo flights might prefer to take the later runway exits such as exit 2 or exit 3 so as to reduce the taxiing distance. This is also verified by the data as out of all the flights that utilised exit 3, 74.4% of them are cargo flights. We can infer from this statistics that the designated airport terminal of the arrival flights may affect the runway exit choice and thus affecting the runway occupancy time. Thus, we have included the designated terminal of the arrival flights as one of the features.

V. EXPERIMENT & RESULTS

Prior to the training of the four different machine learning models, the dataset was split into training and test set with $80:20$ ratio. The training set was further split into four-fold cross-validation set to tune the hyper-parameters of each machine learning model using grid search. Fig. 4 illustrates the formation of the training, test and cross-validation sets and how the final trained model was selected through grid search and cross-validation.

The purpose of the test set was to measure and compare the prediction performance of the final trained models. In order to have a reflective and accurate performance of the models, the models have to be tested on unseen data (i.e., data that were not used during the training of the models). Thus, 20% of the dataset were allocated as the test set before any training took place.

Tuning of the hyper-parameters of the machine learning models is also crucial to prevent overfitting. Overfitting occurs when the machine learning model learns the noise that are present in the training data and as a result, the model fits the training data very well but predicts the test data poorly. Grid search and four-fold cross-validation helps to tune the hyper-parameters by providing various parameter combinations to the model. The outputs are the average RMSE for each of these parameter combinations. The final model was then trained using the parameter combination with the least average RMSE.

A. Decision Tree Regressor

For the Decision Tree Regressor, the depth of the tree, *max_depth*, and the minimum samples in each leaf, *min_samples_leaf*, were tuned. During the grid search process, the range of values being searched for *max_depth* was $[1, 10]$ and the search values for *min_samples_leaf* were $[1, 20]$. After the grid search process and cross-validations, the optimal parameter values were 5 for *max_depth* and 14 for *min_samples_leaf*. If hyper-parameters tuning was not performed, there will be no number assigned to *max_depth* and the *min_samples_leaf* will be 1. This means that the decision tree will keep growing until all the leaves are pure or there are no more information gain. Such fully grown tree is very susceptible to overfitting as the noise within the training data are likely to be modelled inside the tree. The presence of overfitting is reported in Table IV.

TABLE IV
PREDICTION PERFORMANCE COMPARISON FOR DECISION TREE REGRESSOR WITH AND WITHOUT HYPER-PARAMETERS TUNING

	Without Hyper-parameters Tuning	With Hyper-parameters Tuning
Train RMSE (s)	0.14	5.75
Test RMSE (s)	7.79	5.96
R-squared	-0.40	0.18

Train RMSE is the RMSE when the final prediction model predicts the ROT of the training set. Likewise, test RMSE is

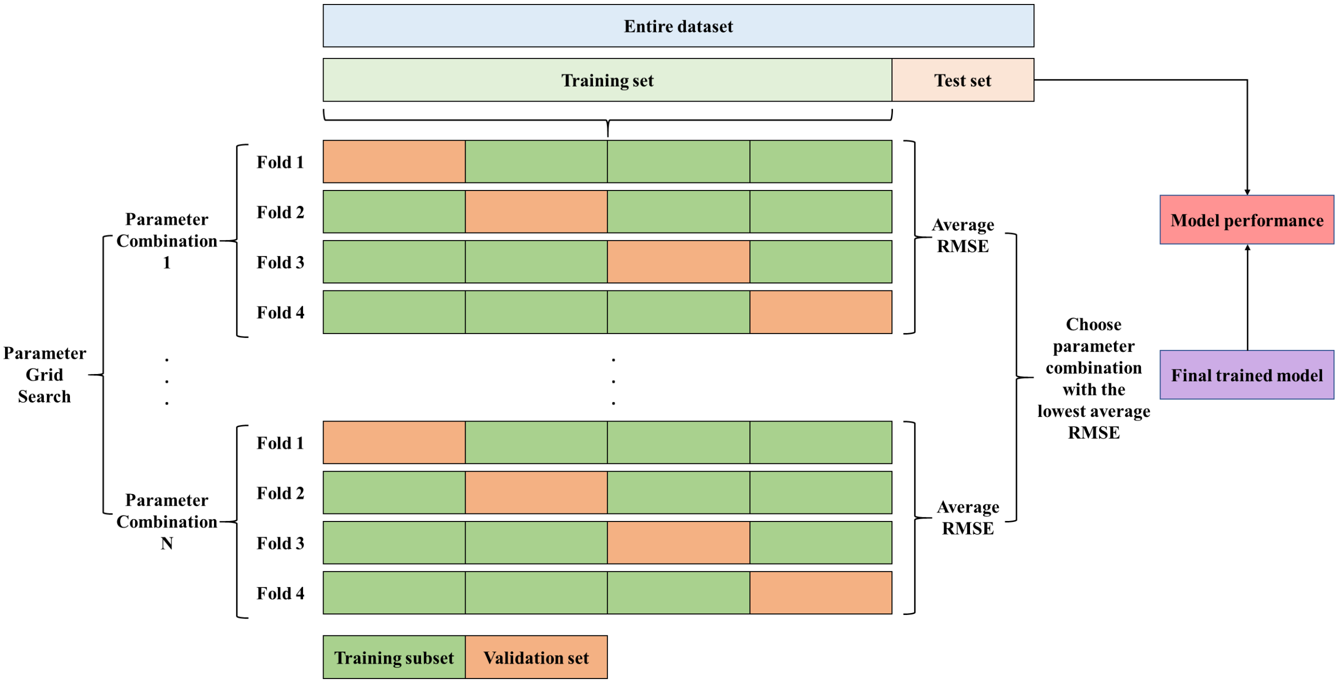

Fig. 4. Training, test and cross-validation sets

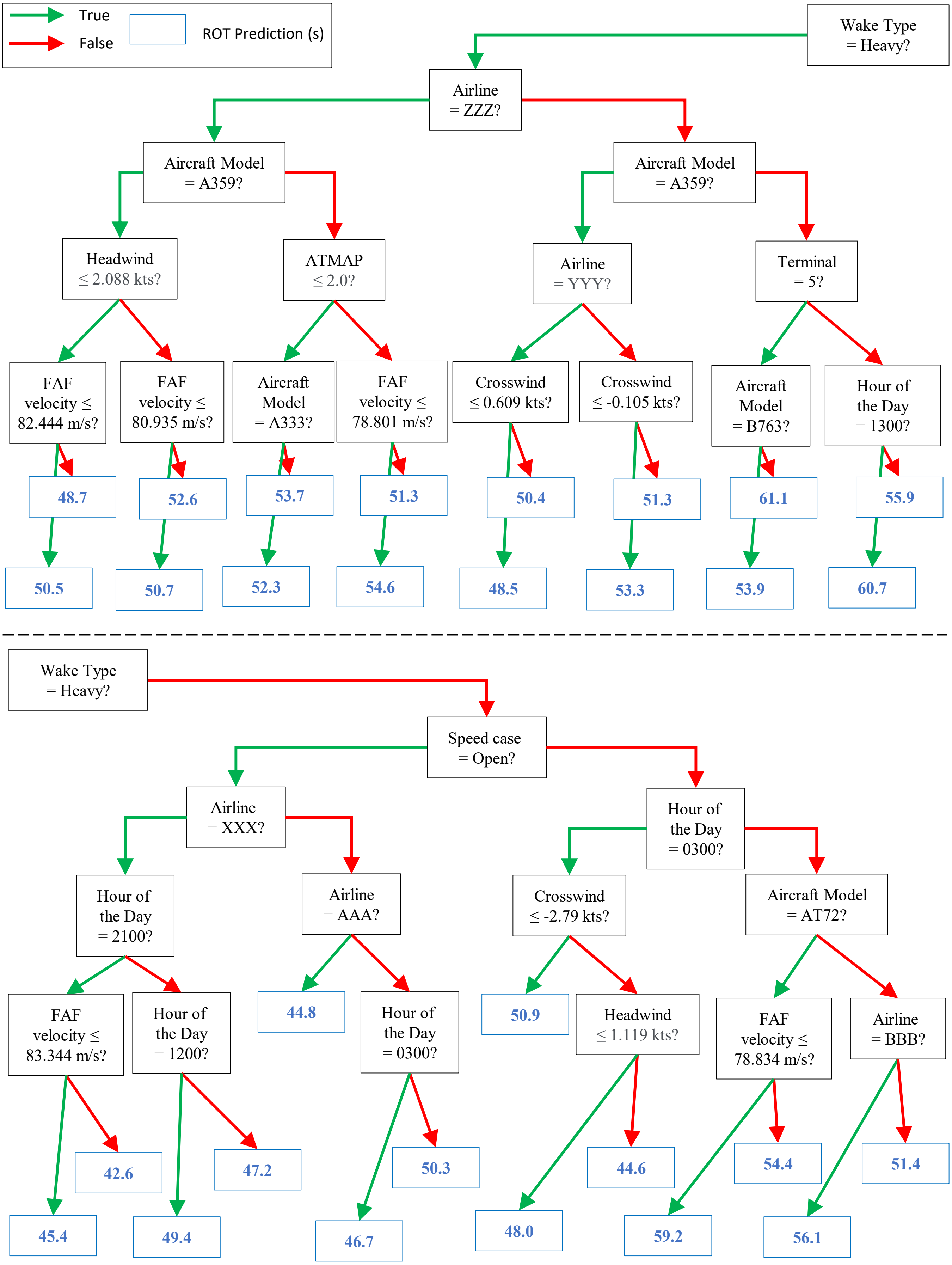

Fig. 5. Trained Decision Tree illustrating prediction flow

TABLE V
LIST OF DECISION TREE RULES SORTED BY PREDICTED ROT IN SECONDS

Rule #	Rules	Predicted ROT
1	wake_type = M, Speed_case = Open, Airline = XXX, HoD = 21, Overall_Velocity_FAF $>$83.344	42.6
2	wake_type = M, Speed_case $\neq$ Open, HoD = 3, Crosswind $>$-2.79, Headwind $>$1.119	44.6
3	wake_type = M, Speed_case = Open, Airline $\neq$ XXX, Airline = AAA	44.8
4	wake_type = M, Speed_case = Open, Airline = XXX, HoD = 21, Overall_Velocity_FAF $\leq$ 83.344	45.4
5	wake_type = M, Speed_case = Open, Airline $\neq$ XXX, Airline $\neq$ AAA, HoD = 3	46.7
6	wake_type = M, Speed_case = Open, Airline = XXX, HoD $\neq$ 21, HoD $\neq$ 12	47.2
7	wake_type = M, Speed_case $\neq$ Open, HoD = 3, Crosswind $>$-2.79, Headwind $\leq$ 1.119	48.0
8	wake_type = H, Airline $\neq$ ZZZ, Aircraft_model = A359, Airline = YYY, Crosswind $\leq$ 0.609	48.5
9	wake_type = H, Airline = ZZZ, Aircraft_model = A359, Headwind $\leq$ 2.088, Overall_Velocity_FAF $>$82.444	48.7
10	wake_type = M, Speed_case = Open, Airline = XXX, HoD $\neq$ 21, HoD = 12	49.4
11	wake_type = M, Speed_case = Open, Airline $\neq$ XXX, Airline $\neq$ AAA, HoD $\neq$ 3	50.3
12	wake_type = H, Airline $\neq$ ZZZ, Aircraft_model = A359, Airline = YYY, Crosswind $>$0.609	50.4
13	wake_type = H, Airline = ZZZ, Aircraft_model = A359, Headwind $\leq$ 2.088, Overall_Velocity_FAF $\leq$ 82.444	50.5
14	wake_type = H, Airline = ZZZ, Aircraft_model = A359, Headwind $>$2.088, Overall_Velocity_FAF $\leq$ 80.935	50.7
15	wake_type = M, Speed_case $\neq$ Open, HoD = 3, Crosswind $\leq$ -2.79	50.9
16	wake_type = H, Airline $\neq$ ZZZ, Aircraft_model = A359, Airline $\neq$ YYY, Crosswind $>$-0.105	51.3
17	wake_type = H, Airline = ZZZ, Aircraft_model $\neq$ A359, atmap total $>$2, Overall_Velocity_FAF $>$78.801	51.3
18	wake_type = M, Speed_case $\neq$ Open, HoD $\neq$ 3, Aircraft_model $\neq$ AT72, Airline $\neq$ BBB	51.4
19	wake_type = H, Airline = ZZZ, Aircraft_model $\neq$ A359, atmap total $\leq$ 2, Aircraft_model = A333	52.3
20	wake_type = H, Airline = ZZZ, Aircraft_model = A359, Headwind $>$2.088, Overall_Velocity_FAF $>$80.935	52.6
21	wake_type = H, Airline $\neq$ ZZZ, Aircraft_model = A359, Airline $\neq$ YYY, Crosswind $\leq$ -0.105	53.3
22	wake_type = H, Airline = ZZZ, Aircraft_model $\neq$ A359, atmap total $\leq$ 2, Aircraft_model $\neq$ A333	53.7
23	wake_type = H, Airline $\neq$ ZZZ, Aircraft_model $\neq$ A359, Terminal = 5, Aircraft_model = B763	53.9
24	wake_type = M, Speed_case $\neq$ Open, HoD $\neq$ 3, Aircraft_model = AT72, Overall_Velocity_FAF $>$78.834	54.4
25	wake_type = H, Airline = ZZZ, Aircraft_model $\neq$ A359, atmap total $>$2, Overall_Velocity_FAF $\leq$ 78.801	54.6
26	wake_type = H, Airline $\neq$ ZZZ, Aircraft_model $\neq$ A359, Terminal $\neq$ 5, HoD $\neq$ 13	55.9
27	wake_type = M, Speed_case $\neq$ Open, HoD $\neq$ 3, Aircraft_model $\neq$ AT72, Airline = BBB	56.1
28	wake_type = M, Speed_case $\neq$ Open, HoD $\neq$ 3, Aircraft_model = AT72, Overall_Velocity_FAF $\leq$ 78.834	59.2
29	wake_type = H, Airline $\neq$ ZZZ, Aircraft_model $\neq$ A359, Terminal $\neq$ 5, HoD = 13	60.7
30	wake_type = H, Airline $\neq$ ZZZ, Aircraft_model $\neq$ A359, Terminal = 5, Aircraft_model $\neq$ B763	61.1

the RMSE when the final prediction model predicts the ROT of the test set. As seen from the table, the train RMSE (i.e., 0.14 seconds) is much lesser than the test RMSE (i.e., 7.79 seconds) when hyper-parameters tuning was not performed. On the contrary, when the parameters are tuned, the train and test RMSE are comparable to be each other. This observation implies that overfitting is present when the parameters are not tuned as the prediction model predicted the training set with high accuracy but preformed badly when prediction was done on the test set. Furthermore, the results also imply that hyper-parameter tuning does reduce the overfitting tendency of the prediction model. Another observation is that the overfitted prediction model has negative R-squared values while the tuned prediction model has a positive R-squared value. The R-squared is the percentage of the variance of ROT that is explained by the prediction model and higher R-squared is preferred. Thus, negative R-squared value suggests that the prediction model did not perform better than just taking the average ROT value as the predicted value.

The generated decision tree is illustrated in Fig. 5. Due to space constraint, the illustration of the decision tree is split into two where the nodes belonging to the left child of the root node is displayed at the top and the nodes belonging to the right child of the root node is displayed at the bottom. All the left child nodes are the true responses while all the right child nodes are the false responses to their parent nodes. The leaf nodes are coloured blue and the number represents the predicted ROT in seconds. After the decision tree was generated, rules were extracted by following the path of the root node to each of the leaf nodes through the decision nodes. Each decision node was split based on the lowest RMSE of the different features. The airline's names were masked to protect their identity.

The set of rules, ranked in ascending order of predicted ROT, for the Decision Tree Regressor as well as the predicted value can be found in Table V. From the table, the shortest ROT (i.e., 42.6 seconds) belongs to a medium aircraft from airline XXX, arriving between 2100 and 2159 hours UTC (0500 and 0559 GMT+8 respectively), with a speed of more than 83.344 m/s at the final approach fix, and with the trailing aircraft traveling slower. The flight with the longest ROT (i.e., 61.1 seconds) belongs to a heavy aircraft heading to the Cargo Terminal. This aircraft does not belong to airline ZZZ and the aircraft model is neither an A359 nor a B763. Based on our observations, flights with medium wake type are dominant (about 67%) in the top half of the rules (i.e., those with smaller ROT).

Features importance was also investigated to understand the significance of each of the features. The two common techniques are impurity-based and permutation-based feature importance. The impurity-based feature importance measures the significance of each feature by finding the sum of information gain due to all the splits performed by that feature. However, this method has the tendency to favour variables with high cardinality [12], [13]. On the other hand, the permutation importance can tackle this drawback. It investigates the importance of each feature by randomly permutating the values of that feature and records how much the prediction performance is affected by the permutation. If the permutation adversely impacts the performance of the prediction model, this means that the permutated feature is of high importance. Likewise, if the prediction performance is only affected marginally by the permutated feature, this means that the feature is not significant in the prediction model. Fig. 6 illustrates the permutation importance of all the features. The feature that ranks the highest in terms of importance is the wake turbulence category of the aircraft, followed by the airline, the aircraft model, and the hour of the day. The day of the week is of no significance for this prediction model as permutating the values do not affect the prediction at all.

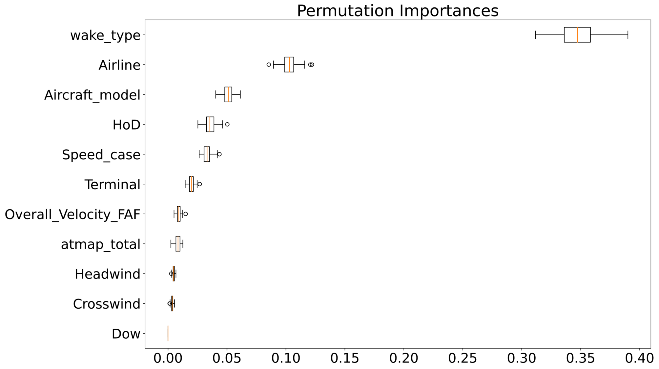

Fig. 6. Permutation importance of the features

B. *Other Machine Learning models*

Similar to the Decision Tree Regressor, the other learning machine models were trained and tested with the $80 : 20$ ratio split. Hyper-parameters tuning was also performed on these models. In addition, the time taken for the entire training process was recorded for the different machine learning models. The performance and runtime for the models, Random Forest (RF) Regressor, CatBoost (CB) Regressor, Multilayer Perceptron Neural Network (MLP) and Decision Tree (DT) Regressor, are compared in Table VI. The performance metrics used are MAE, RMSE and R-squared. The test MAE and RMSE of the models are comparative, with the lowest RMSE (i.e., 5.76 seconds) belonging to the CatBoost Regressor, and the highest RMSE (i.e., 6.10 seconds) belonging to the Multilayer Perceptron Neural Network. The Multilayer Perceptron Neural Network also took the longest to run, taking about 124 minutes, compared to the 27 minutes that the CatBoost Regressor took. The fastest machine learning model was the Decision Tree Regressor, at 0.28 minutes.

TABLE VI
PREDICTION PERFORMANCE COMPARISON FOR ALL MACHINE LEARNING MODELS

	RF	CB	MLP	DT
Test MAE (s)	4.50	4.35	4.74	4.62
Test RMSE (s)	5.88	5.76	6.10	5.96
R-squared	0.20	0.24	0.14	0.18
Time taken to run (mins)	76.22	26.53	123.67	0.28

VI. CONCLUSIONS

This study develops an explainable prediction model that can be used to predict the ROT of aircraft landing on runway 02L at Singapore Changi Airport when the aircraft is at the final approach fix. There are two parts to this study: the first part is to develop the prediction model, and the second is to test the model and compare it with other more complicated machine learning models. We began by extracting the ROT and required features from the data source. The data was cleaned and then split into train and test sets for the machine learning models. The models used the train set to generate its rules and the capability was assessed on the test set. We also tuned the hyper-parameters of the models to prevent overfitting.

From our results, we have shown that the performance of the models in terms of their RMSE are similar. However, the key difference is the runtime of the model and how explainable it is. Out of all the models, the Decision Tree Regressor is the fastest at only 0.28 minutes. Furthermore, it is the most explainable model as it gives us explicit rules on how the outcome is achieved. This will give ATCs a better understanding of how the final outcome is achieved and might increase their overall confidence in a ROT prediction model. In addition to that, by applying permutation importance to the Decision Tree, we can conclude that the ICAO wake turbulence category is the biggest factor in the ROT prediction.

Since data from only one runway was used in this study, data from more runways can be used in future work. This can ensure that the performance of the machine learning model stay consistent among the different datasets, thus boosting the confidence in the model's prediction capabilities. In addition, since only runway 02L (northeast direction) and data for October is used in the study, weather data from different months can be used to determine if the prevailing winds for those periods (e.g., northerly to northeasterly winds in the period of December - March, southeasterly to southerly winds from June - September) will affect the ROT. If the prevailing winds are a major factor, different prediction models might be needed for different periods to obtain better prediction performance.

ACKNOWLEDGMENT

This research is supported by the National Research Foundation, Singapore, and the Civil Aviation Authority of Singapore, under the Aviation Transformation Programme. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore and the Civil Aviation Authority of Singapore.

REFERENCES

[1] Thomas L Spencer and Antonio A Trani. Predictive models of departure and arrival occupancy time and takeoff distance. *Journal of Air Transportation*, 27(2):1, 2019.

[2] Dario Martinez et al. A boosted tree framework for runway occupancy and exit prediction. In *Eighth SESAR Innovation Days*, 2018.

[3] Nicolas Pierre Meijers. *Data-driven predictive analytics of runway occupancy time for improved capacity at airports*. PhD thesis, Massachusetts Institute of Technology, 2019.

[4] Steven E Koenig. Analysis of runway occupancy times at major airports. Technical report, MITRE CORP MCLEAN VA METREK DIV, 1978.

[5] William E Weiss and JN Barrer. Analysis of runway occupancy time and separation data collected at la guardia, boston, and newark airports. Technical report, MITRE CORP MCLEAN VA, 1984.

[6] Zhi Jun Lim et al. Causal effects of landing parameters on runway occupancy time using causal machine learning models. In *2020 SSCI*, pages 2713–2722. IEEE, 2020.

[7] Tamas Kolos-Lakatos. *The influence of runway occupancy time and wake vortex separation requirements on runway throughput*. PhD thesis, Massachusetts Institute of Technology, 2013.

[8] EUROCONTROL. Technical note: Algorithm to describe weather conditions at european airports.

[9] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[10] Anna Veronika Dorogush et al. Catboost: gradient boosting with categorical features support. *arXiv preprint arXiv:1810.11363*, 2018.

[11] Matt W Gardner and SR Dorling. Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. *Atmospheric environment*, 32(14-15):2627–2636, 1998.

[12] Carolin Strobl, Anne-Laure Boulesteix, Achim Zeileis, and Torsten Hothorn. Bias in random forest variable importance measures: Illustrations, sources and a solution. *BMC bioinformatics*, 8(1):1–21, 2007.

[13] Kristin K Nicodemus. Letter to the editor: On the stability and ranking of predictors from random forest variable importance measures. *Briefings in bioinformatics*, 12(4):369–373, 2011.