

A Machine Learning Model for Normal and Extended Taxi-Out Time Prediction

Vienna Airport Case Study

Michael Probyn

Technische Universiteit Delft

A Machine Learning Model for Normal and Extended Taxi-Out Time Prediction

Vienna Airport Case Study

by

Michael Probyn

in partial fulfillment of the requirements for the degree of

Master of Science
in Aerospace Engineering

at the Delft University of Technology,
to be defended publicly on Thursday 20th August, 2020 at 09:00.

Student number: 4213599

Supervisors: Ir. P. C. Roling TU Delft, LR - C&O
Dr. Ir. F. F. Herrema EUROCONTROL

An electronic version of this thesis is available at <http://repository.tudelft.nl/>.

Cover image source: <https://quebec.huffingtonpost.ca/>

Preface

This Master Thesis marks the end of my studies at Delft University of Technology. The journey was long, and while I have thoroughly enjoyed my time here, I am ready for it to be over. Of course, I would not have been able to complete this journey without the help of many people.

First of all, I would like to thank my supervisor, Ir. Paul Roling, for his continuous guidance and support, and for taking the time to meet with me and share his valuable insights. I would also like to thank Dr. ir Floris Herrema from EUROCONTROL for his supervision at the start of my thesis, and for his continued support and ML related guidance throughout. Also, thank you for the data, and for helping me with its understanding.

Next, I would like to thank all of my friends that supported me along the way, without which I would not have made it this far. Thank you to all of my housemates, past and present, especially Thomas, Thibault and Joe who have been there since the start. The 'bouncer' has been endless and I hope for it to continue for many years to come. Also, thank you to Kostas for his help throughout the Masters, to Michael for the beers and the BBQs, and to my Banchory friends for their support from a distance. Special thanks to my girlfriend, Aurora, for keeping me sane during lockdown. Your support has been incredible and I cannot express how grateful I am for your help.

Lastly, I would like to thank my family who have always supported me. Thank you to my parents for their endless patience and wisdom, and again to my Dad for his help with my report. Also thank you to my brothers for the cs matches and helping me enjoy the last few months of my thesis.

*Michael Probyn
Delft, August 2020*

I

Article

A Machine Learning Model for Normal and Extended Taxi-out Time Prediction at Vienna International Airport

Michael Probyn

MSc Student, Air Transport and Operation, Delft University of Technology

Supervisors: Ir. P.C. Roling, Dr. Ir. F.F. Herrema*
*Delft University of Technology, *EUROCONTROL*

All major airport operators face a similar challenge, namely ensuring maximum throughput and maintaining high runway utilisation. A key part of this is accurately planning aircraft movements on the ground to avoid queueing and associated delays. A primary indicator of the operator performance in this area is the Taxi-Out Time. The research objective of this article is to review whether the application of machine learning can be used to model the departure process in such a way as to provide accurate prediction of TXOT taking into account a wide range of variables. A regression tree type machine learning model is developed using actual data from Vienna Airport and a selected set of significant predictor variables. The taxi-out times of the test set of flights are closely predicted with an RMSE of 2.03 minutes for normal taxi-out and 3.75 minutes for extended taxi-out.

Nomenclature

AOBT	Actual Off-Block Time	MSE	Mean Squared Error
ANSP	Air Navigation Service Provider	NNLM	Neural Network Levenburg-Marquardt
ATCO	Air Traffic Controller	PDF	Probability Distribution Function
ATOT	Actual Take-Off Time	QT	Queuing theory
CDF	Cumulative Distribution Function	RL	Reinforcement Learning
d_{max}	Maximum tree depth	RMSE	Root Mean Squared Error
FAA	Federal Aviation Administration	RT	Regression Tree
ICAO	International Civil Aviation Organization	RWY-STD	Runway-Stand
l_{min}	Minimum leaf size	TXOT	Taxi-Out Time
METAR	Meteorological Terminal Air Report	UTXOT	Unimpeded Taxi-Out Time
ML	Machine Learning	WMA	Wind measurement apparatus
MLP	Multilayer Perceptron		

I. Introduction

ALL major airport operators face a similar challenge, namely ensuring maximum throughput and maintaining high runway utilisation. A key part of this is accurately planning aircraft movements on the ground to avoid queueing and associated delays. This benefits the operator by enhancing their overall efficiency and hence their reputation with the airlines. It also benefits airlines by reducing their fuel costs and keeping their customers happy with faster transit times.

A primary indicator of the operator performance in this area is the Taxi-Out Time or TXOT. This study sets out to review whether the application of machine learning can be used to model the departure process in such a way as to provide accurate prediction of TXOT taking into account a wide range of variables. Enhancing the understanding of the influence of variables on the overall predicted TXOT is also seen as a

beneficial outcome.

This article covers a review of existing literature on modelling of the flight departure process in section II. A description of data preparation steps required is presented in section III. The approach taken to select and develop a machine-learning model and the further steps to optimise this are described in sections IV and V. Analysis of results is to be found in section VI and final recommendations are presented in section VII.

II. Related Work

This section presents relevant literature relating to modelling of the airport departure process, and more specifically the prediction of taxi-out time. The literature will be divided into two main groups: one which relates to statistical methods for modelling the prediction of TXOT including machine-learning techniques, and another which explores existing models that describe the departure process using queueing theory.

A. Statistical methods

Statistical methods have been shown to be effective in modelling the departure process where sufficient data is available to incorporate into the analysis. In particular, the application of machine-learning techniques is widely researched and many different models have been put forward as being suitable for taxi-out time prediction purposes.

It has been determined that methods which incorporate human knowledge in combination with analytical models, e.g. fuzzy rule-based models, tend to deliver the best results [1]. In a comparison of different machine-learning techniques [2] the regression tree method is found to be more efficient than neural network or reinforcement learning methods while delivering more robust prediction capabilities. RL techniques are shown to be useful where large uncertainties require near real-time data to update predictions [3].

In all these methods the importance of identifying the critical underlying factors or features with significant impact on taxi-out time variability has been highlighted. A recent study identifies the main factors as runway configuration, the airline/terminal, the downstream restrictions, and the take-off queue size, where the latter was deemed to be the most important [4]. Other studies include as factors the number of departures/ arrivals in the period leading up to and during taxi-out [5, 6]. Most studies agree that the amount of traffic in the system is the primary factor influencing variability in taxi-out times.

B. Queueing Theory

Queueing theory is the mathematical study of waiting in lines or queues, in which models are constructed so that queue length and waiting time can be predicted [7]. This approach is considered well suited to the airport departure process, which can be described simply as a combination of taxi-out travel time plus the waiting time at the runway entrance. As noted, the take-off queue size is a significant contributor to variability in the TXOT [4].

To provide greater accuracy most QT-based models described in the literature use unimpeded taxi-out time, additional taxi-out time, and queue time to model the airport departure process. The concept of unimpeded taxi-out time or UTXOT is fundamental to the working of these models although there is no common accepted definition. One approach promoted by EUROCONTROL for alignment of performance reporting uses a statistical analysis of taxi-out times against runway-stand combinations to identify standard times for unimpeded taxi-out [8]. The FAA uses a process which aims to build a numerical relationship between the number of aircraft on the ground and taxiing time through a linear regression model [9]. Other UTXOT concepts are often integral to the overall departure process modelling approach and, as such, are not separately described.

Having established the baseline for travel time to the runway queue the models reviewed incorporate various analytical or numerical methods to simulate overall departure performance. An early study [10]

split the process into service demand and departure delay where the queueing time was estimated using either exponential or Erlang-k models with or without server absence (runway availability). This work is subsequently enhanced by incorporating a more granular approach and evolving concepts of UTXOT which achieved better results when compared to a baseline deterministic approach [11–13]. Of the numerical models reviewed, one is a predictive TXOT model which incorporates gate departure delay [14], and others are models in which the travel time and runway queue service rates are modelled as stochastic variables [15] and [16]. The latter models the queue service time based on a varying Poisson distribution which changes according to congestion levels.

III. Data Preparation

Access to quality data is critical to enable accurate predictive modelling and validation of airport operations. Given the availability of a relevant, high quality data set covering an extended period of operations at Vienna International Airport, this airport was chosen as the basis for the TXOT model. For reference, the layout of the airport is shown in Figure 1.

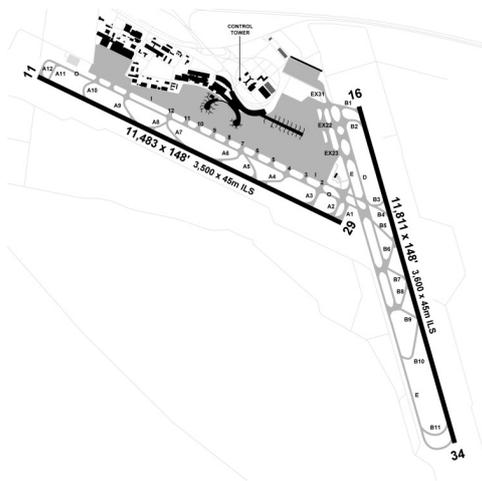


Table 1 Number of flights per runway

Runway	Departures		Arrivals	
	Number	%	Number	%
11	2938	2.5	15034	12.5
29	75554	64.5	16221	13.5
16	22814	19.5	33446	28.0
34	15911	13.5	55164	46.0
Total	117217	100	119865	100

Fig. 1 Layout of Vienna International Airport*

The data used was provided by Austrocontrol, the ANSP at Vienna Airport. It consists of one year of airport operations data from January 1st 2015 until December 31st 2015 compiled from multiple data sources. The data consists of all arrival and departure flights from all runways, a total of around 240,000 flights. A breakdown of these flights can be found in Figure 1.

The sources from which the data is extracted include the Advanced Surface Movement Guidance and Control System (A-SMGCS), the wind measurement apparatus (WMA), Meteorological Terminal Air Report (METAR) observations, and Radar track data. The A-SMGCS specifies time stamps such as AOBT and ATOT per aircraft as well as general flight information. The wind speed meter and METAR observations provide wind and weather information which includes information such as temperature, wind speed, and cloud ceiling. Finally, the radar track data specifies aircraft latitude, longitude, ground speed and flight level every second. However, only 6 months of radar track data was provided, and could therefore not be merged with the main data set. Instead, it was used for verification purposes achieved by plotting ground tracks of the 6 months of flights and observing routes and timestamps of aircraft.

In total 23 prediction variables could be extracted directly from the data provided by Vienna Airport. An analysis performed to identify additional variables which have a significant effect on TXOT revealed that

*<https://rzjets.net/airports/?code=LOWW>

several other prediction variables needed to be included. The analysis was based on the EUROCONTROL PRU methodology [8], which suggested the following additional variables have a significant influence on the TXOT: ‘congestion level’, ‘unimpeded taxi-out time per RWY-STD and STD group’, ‘saturation level’, and ‘number of departures in the last 20 mins’. To aid understanding, the interrelationship of these variables is illustrated for a selected RWY-STD combination in Figure 2. At low congestion levels TXOTs remain constant at the nominal UTXOT up to the saturation level, after which the TXOT trend increases due to congestion delays. Other related features extracted during the calculation of the aforementioned features include: ‘airport throughput in the last hour’, ‘1st estimation of UTXOT per RWY-STD and STD group’, and a ‘UTXOT indicator’ indicating whether or not a flight is unimpeded.

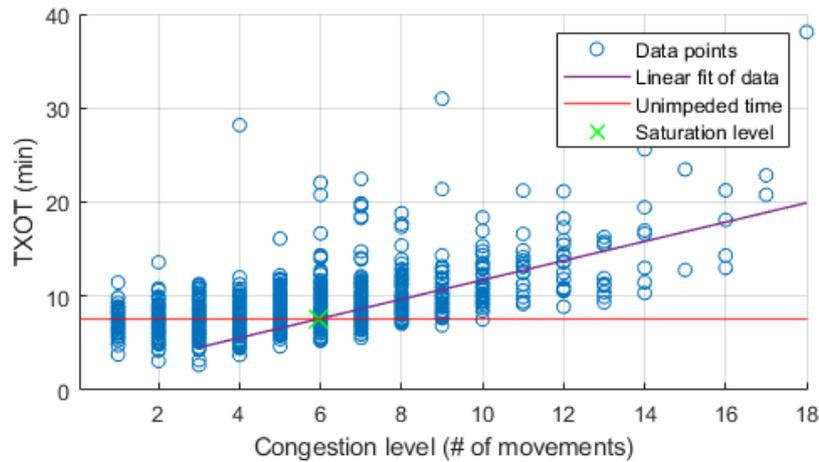


Fig. 2 TXOT vs. congestion level for RWY-STD combination C34-R29

The additional variables were calculated and merged with the main data set, resulting in a final table consisting of 35 variables including the target variable TXOT.

The next task in the data preparation stage is the identification of the subset of variables which most significantly influence the TXOT, or feature selection. A commonly used feature selection technique is the RreliefF regression modelling technique. This technique has been extensively researched [2, 17] and is therefore applied in this study.

The objective of feature selection is threefold: to improve the prediction performance of predictors, provide faster computational performance and more effective predictors, and provide a better understanding of the underlying process that generated the data. The technique was applied to the TXOTs of roughly 120,000 flights at Vienna airport, the results of which can be found in Figure 3.

The top features according to this method were found to be as follows: ‘congestion level’, ‘UTXOT’, ‘saturation level’, ‘1st estimation UTXOT per RWY-STD’, ‘1st estimation UTXOT per RWY-STD group’, ‘departure runway’, ‘ICAO weight category’, ‘aircraft weight’, ‘stand group’, ‘aircraft type’, ‘departures in preceding 20 mins’, ‘unimpeded flight’, ‘throughput in preceding hour’, ‘hour of flight’, and ‘AOBT’. These will be used in optimising the prediction model later in the process.

A. Normal vs Extended TXOTs

While several studies have been performed investigating different methods for normal TXOT, little has been done regarding the prediction of extended TXOTs. An analysis was performed using the MATLAB function ‘isoutlier’ to determine which flights could be considered to have ‘extended’ TXOTs. Based on this, and to ensure sufficient data points, it was decided to define an ‘extended TXOT’ flight as one with a TXOT

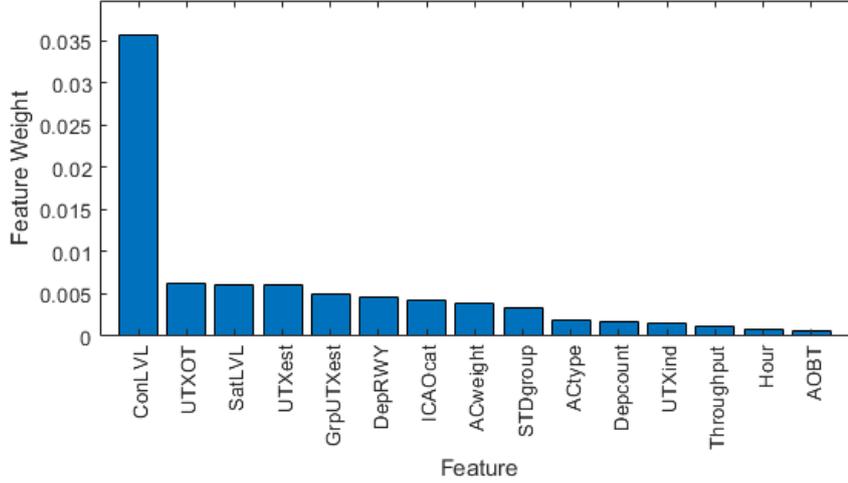


Fig. 3 Feature selection results using RReliefF algorithm

more than 2 standard deviations (σ) from the mean TXOT of all flights. Given this definition, 4050 flights at Vienna Airport in 2015 are considered to have ‘extended’ TXOTs. These flights comprise 3.46% of the total flights, and are shown in Figure 4, where the threshold TXOT is 17.3 mins.

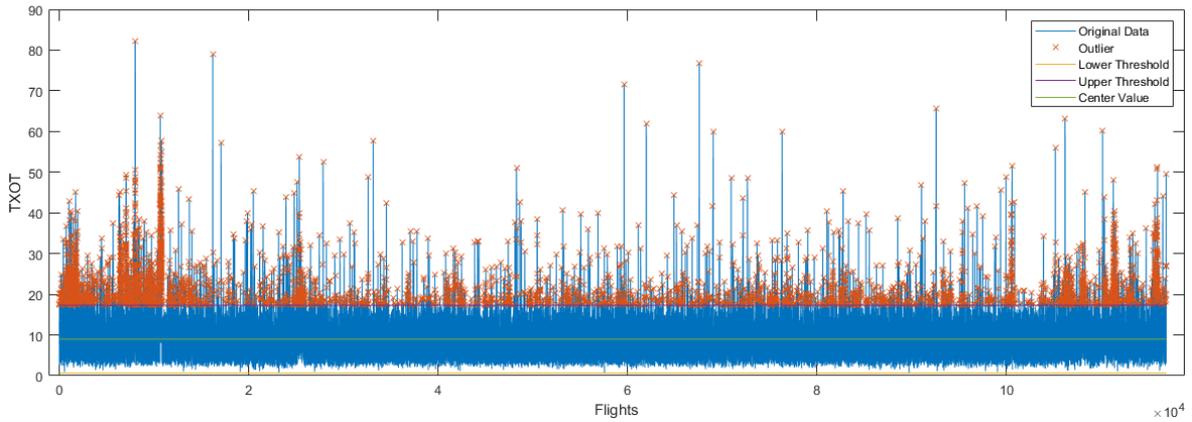


Fig. 4 Extended TXOT threshold at 2σ

IV. Machine learning model selection

In order to produce an accurate TXOT prediction model the Neural Network Levenburg-Marquardt and Regression Tree techniques were tested and assessed in relation to their performance indicators.

A. Neural Network Levenburg-Marquardt (NNLM)

The first method explored is the NNLM technique. In neural network fitting problems, the aim is to create a map between a data set of numeric inputs and a set of numeric targets. In this case, the inputs are the features extracted in section III, and the target is the TXOT.

For both the RT and NNLM techniques, the stability of the dataset is assessed. This process can be split into 3 tasks as follows:

- 1) Standardise the feature matrix ‘X’ based on the prediction variables. Also, separate the TXOT from feature matrix X to create the target variable matrix ‘Y’.
- 2) Split matrices ‘X’ and ‘Y’ into three subsets, namely, X_{train} and Y_{train} , $X_{validation}$ and $Y_{validation}$, and X_{test} and Y_{test} .
- 3) Analyse the different data subsets based on their splitting ratios.

In step 1, the variables are arranged in a logical order and prepared in such a way that the X and Y matrices can be used as an input for both ML methods.

In step 2, the default splitting ratios of 0.7, 0.15, and 0.15 were used for training, validation and testing respectively. For the NNLM technique, the training data set is presented to the network and the network is adjusted according to its error. The validation set is used to measure network generalisation, and to halt training when generalization stops improving. Finally, the test set is used to measure the overall predictive performance of model. This set has no effect on training and therefore provides an independent measure of network performance after training.

Lastly, in step 3, the stability of the datasets are proven. To achieve this, epoch and validation checks are performed, where the number of epochs represents the number of times the algorithm passes through the entire training set when training the model. A convergence check is then performed on the validation set, after which the model is evaluated on the test set.

Having prepared the datasets, the model was trained using all 116899 flights from all stands to all runways. Additionally, the default number of neurons, default splitting ratios, and all the prediction variables were used. The model was then retrained using only the top features found in section III. The results of both models are displayed in Table 2. The accompanying epoch check and error histogram of the model trained only with the top features are shown in figures 5 and 6.

Clearly, both models have very similar results. However, by excluding 14 variables, the model is trained faster and is more robust when inserting new data with a similar structure.

Table 2 Comparison NNLM results

Features	Data Set	MSE	R	Computation	%	%	Median	Distributions	Traceability
				time (s)	3 mins	5 mins			
All	Test	3.23	0.901	33	92.8	98.59	8.48	Yes	No
Top	Test	3.49	0.891	31	92.1	98.26	8.48	Yes	No

B. Regression Tree (RT)

Having prepared the feature matrix and tested the stability of the data, the data could directly be used in the regression tree. The purpose of a RT is to extract a set of if-then-else (what-if statements) split conditions in order to identify the main precursors that influence the TXOT.

The data was once again split into training, validation, and test data sets with a ratio of 0.7, 0.15, and 0.15 respectively. As with the NNLM, 2 models were trained using the training data set, one using all the prediction variables, the other using only the top variables. For both models the default parameters were used. The model performance of both models on the test set can be found in Table 3.

As can be seen, reducing the number of features only results in a minimal increase in MSE. However, the model trains faster, and is more robust when inserting new data with a similar structure.

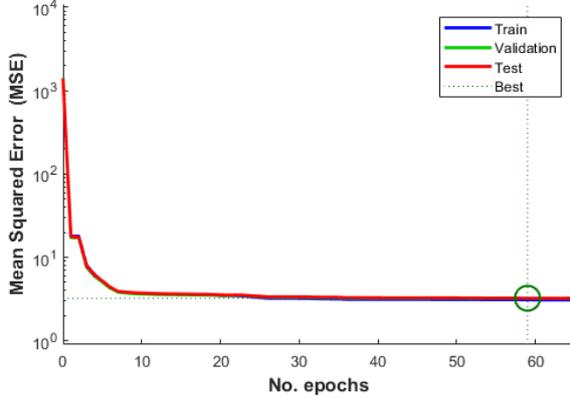


Fig. 5 MSE vs. number of epochs for the NNLM technique using all features

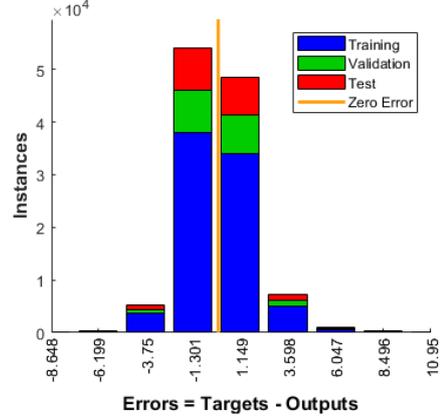


Fig. 6 Histogram of TXOT prediction error per flight

Table 3 RT results for models trained using all features and top features only

Features	Data set	MSE	R	Computation time (s)	% 3mins	% 5mins	Median	Distributions	Traceability
All	Test	3.27	0.902	5.8	93.43	98.47	8.44	Yes	Yes
Top	Test	3.4	0.897	5.1	93.17	98.25	8.46	Yes	Yes

C. Machine learning technique comparison

By comparing tables 3 and 2, it is observed that the performance of both techniques are very similar, with the RT performing marginally better for most of the performance indicators. The two exceptions to this are the computational time and the traceability. The computation speed of the RT is roughly 6 times faster, and most importantly, the results are traceable. Traceability refers to the ability to provide an explanation of how the results were derived. While the predictions of a RT can be explained by following decisions made by the tree at each split node, NNLM is more of a "black box" that delivers results without an explanation. As such, the RT method was chosen as the most suitable method for TXOT prediction and will be developed in section V.

V. Model optimisation

Having chosen the RT as the most suitable method for TXOT prediction, the model then had to be optimised. In order to create a tree that is not overcomplicated while still producing sufficiently accurate results, the optimal tuning parameters need to be found. In this case, the parameters that need to be tuned are the minimum leaf size, l_{min} , and the maximum tree depth, d_{max} . To assist with the explanation of these parameters, an example RT can be found in Figure 7.

A. Normal TXOT

The first parameter is the minimum leaf size, l_{min} , for which enough data points are required in each terminal node (leaf) to create a distribution while still producing accurate results. In general, a fine tree with many leaves is highly accurate on the training data, but has a less comparable accuracy on the validation and test sets. In contrast, a coarse tree with fewer leaves may not capture the important structure of the underlying process. As seen in Figure 8, the tree is trained multiple times using all the variables and varying leafsizes.

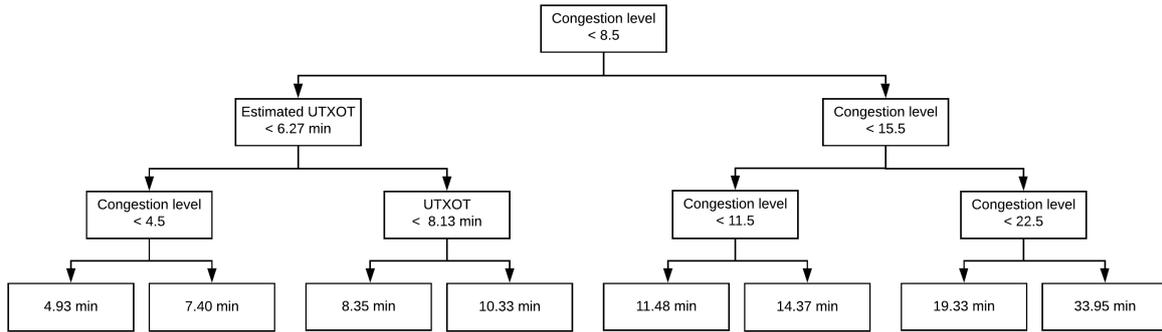


Fig. 7 Example RT with tree depth 3

The optimal leaf size can then be determined by assessing the corresponding cross-validated error.

As observed in Figure 9, the minimum cross-validated error occurs at a minimum leaf size of 28. However, this is the performance error on the training set. In order to avoid over-fitting, a slightly larger leaf size should be chosen. Additionally, a larger leaf size ensures a more accurate distribution can be fitted. A leaf size of 42 was therefore chosen.

The second parameter is the maximum tree depth, d_{max} . The maximum tree depth is used to restrict the number of layers of a tree. This is done to alter the interpretability of a tree, where a deeper tree (tree with more levels) is harder to interpret. This is an important factor when considering ease of use for ATCOs. Additionally, a very large tree with many leaves might overfit the data, whereas a small tree might not be able to capture the important structure of all the features. As seen in Figure 9, the optimal tree depth is determined by plotting the RMSE vs tree depth for the training and validation data sets and using all the variables.

While the predictive accuracy of the model on the training set may increase past a tree depth of 11, the accuracy on the validation set does not. Setting the tree depth any higher than 11 would cause the tree to ‘overfit’ overfit to the training data and be less robust when introduced to new data. Additionally, the greater the tree depth, the more difficult it is to interpret. The maximum tree depth was therefore set at 11 to ensure the interpretability of the tree while maintaining a sufficiently high predictive accuracy.

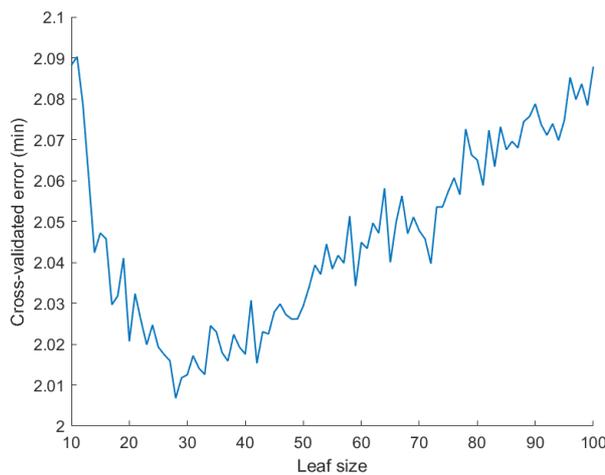


Fig. 8 Cross-validated error for varying minimum leaf size l_{min}

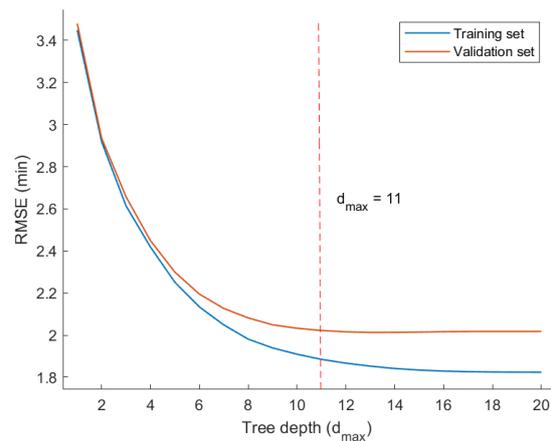


Fig. 9 RMSE vs. tree depth for training data set and validation data set

B. Extended TXOT

To determine the most significant factors influencing ‘extended’ TXOTs, a new RT was built using the data related to flights with extended TXOTs. Parameters were tuned in exactly the same way as for the normal TXOT RT. For this tree, the optimal l_{min} was determined to be 20, while the optimal d_{max} was found to be 9.

VI. Results

Having determined the optimal parameters for both the normal and extended TXOT models, the trees were trained using these parameters and all prediction variables. Using the `predictorImportance` function in MATLAB a new set of top features for each model could be extracted and the trees were then re-trained using these. Inspection of the outputs showed that further adjustment of the tuning parameters was not required at this stage. The prediction results and associated top features are presented below.

A. Normal TXOT

Having trained the normal RT using the optimal parameters and all variables, the top features and their respective importances were extracted as shown in Figure 10. A final tree was then trained using only these top variables.

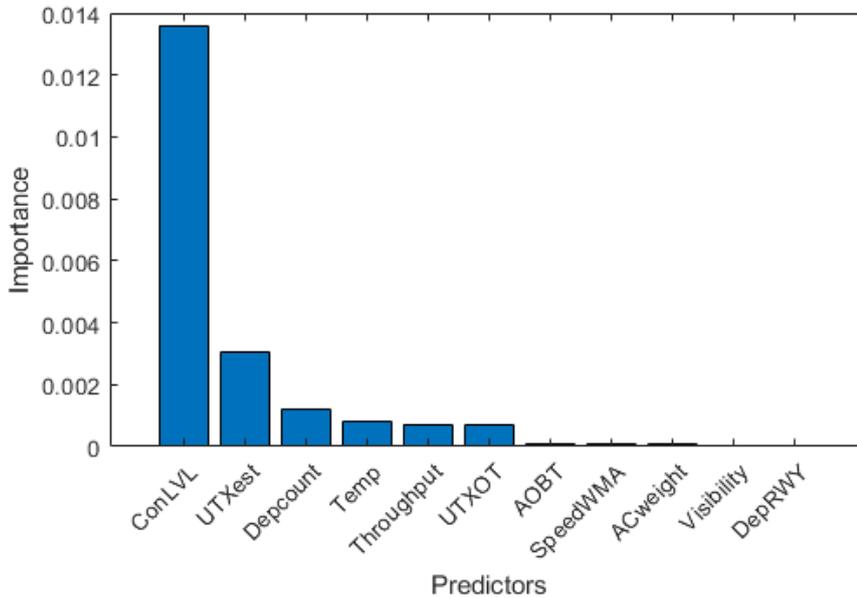


Fig. 10 Top features extracted from normal TXOT RT using predictorImportance function

Applying the model to the test data set delivered the results shown in Table 4. The results are displayed alongside the results of the RT trained with all variables for comparison.

From these results, it is clear that the use of top features is effective as there is a negligible decrease in predictive performance. Additionally, the model trains faster, and is more robust to new data.

The results of the RT in comparison to the actual TXOTs are visualised in figures 11 and 12. While TXOTs between roughly 4-6 minutes are sometimes overpredicted, resulting in the spike at around 8 minutes, for the most part the predictions match well. This indicates that the characteristics of the taxi-out process have been captured.

Finally, a parametric distribution is fitted to each terminal leaf. The probability distributions considered

Table 4 Results and comparison for normal TXOT RTs

Features	Leaf size	Tree depth	RMSE	R	Computation time (s)	% _c		Median
	l_{min}	d_{max}				3 mins	5 mins	
All	42	11	2.024	0.875	3.49	91.14	97.90	8.48
Top (RT)	42	11	2.026	0.875	3.14	91.32	97.88	8.42

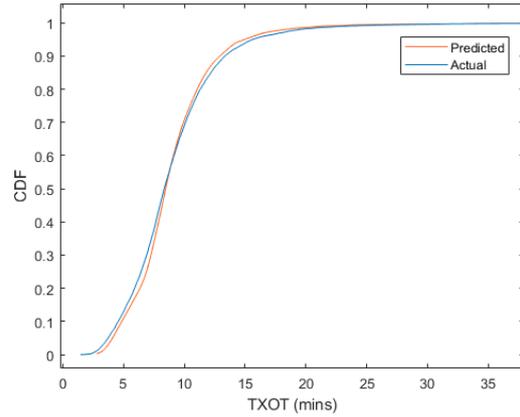
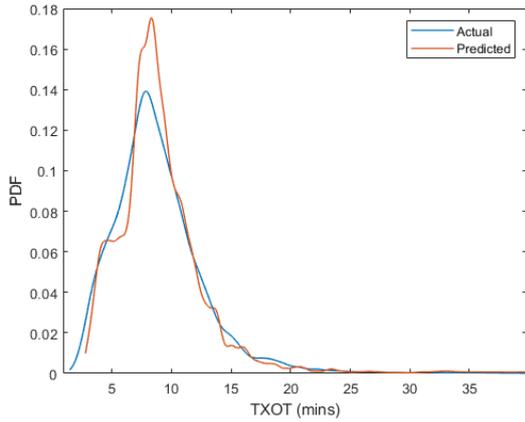


Fig. 11 PDF of actual TXOT vs predicted TXOT Fig. 12 CDF of actual TXOT vs predicted TXOT

include the ‘Birnbbaum-Saunders’, ‘Gamma’, ‘Gumbel’, ‘Logistic’, ‘Lognormal’, ‘Normal’, ‘tLocationScale’, and ‘Weibull’ distributions. The best fit was determined to be the ‘Gamma’ distribution. This distribution fitted to 60 example leaves (out of 668) is shown in Figure 13. Each distribution provides an indication of the confidence of a prediction. If there are more flights in a leaf and the distribution is less spread out, more confidence can be assigned to the prediction.

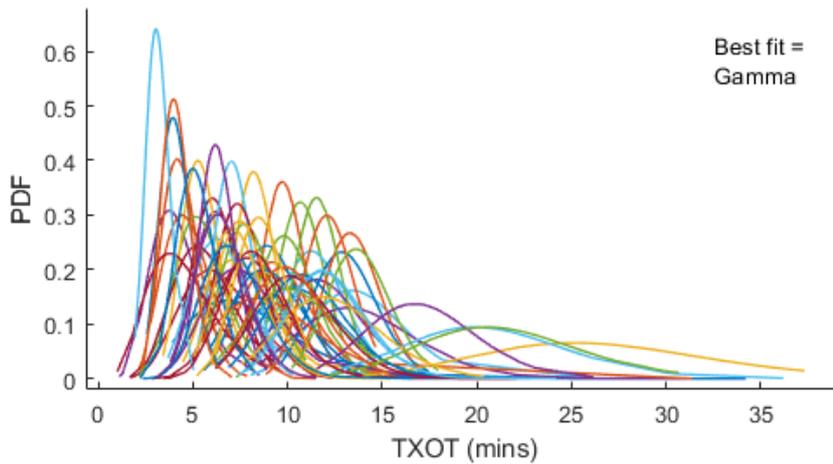


Fig. 13 Gamma distributions for 60 example leaves

B. Extended TXOT

Having trained the extended RT using the optimal parameters and all variables, the top features and their respective importances were extracted as seen in Figure 14. The tree was then re-trained using only these variables and applied to the test set, the results of which can be seen in Table 5. The results are displayed alongside the results of the RT trained with all variables for comparison.

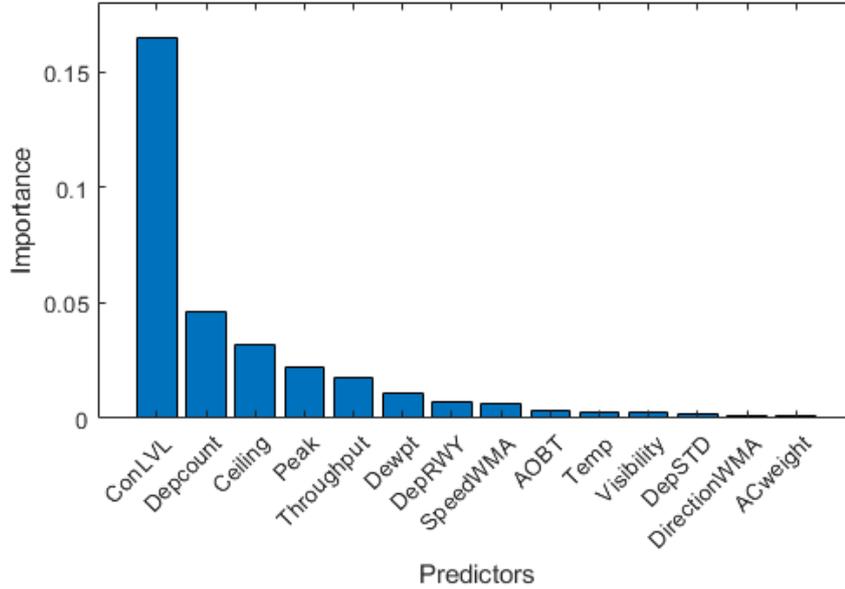


Fig. 14 Top features extracted from extended TXOT RT using predictorImportance function

Table 5 Results and comparison for extended TXOT RTs

Features	Leaf size	Tree depth	RMSE	R	%	%	%
	l_{min}	d_{max}			3 mins	5 mins	10 mins
All	20	9	3.75	0.789	68.21	84.93	97.52
Top	20	9	3.75	0.79	68.38	84.6	97.35

Clearly, using only the top features is effective. The optimised model shows a negligible change in predictive performance, and is more robust to new data input.

C. Example prediction

In order to help visualise the decisions made by the model to produce a prediction, and to provide a clear link to reality, a prediction for an example flight will be made. In Table 6 the data corresponding to the example flight, referred to as flight 'X', chosen at random, is displayed. In Figure 15, a part of the final RT used to make predictions for extended TXOTs is shown. The path highlighted in red shows the decisions that were made in order to achieve the TXOT prediction for flight 'X' and correspond to the highlighted cells in Table 6.

Starting at the root node, if the split condition is true (i.e. congestion level < 22.5) the lower path to the next node is followed. Conversely, if the split condition is false (i.e. congestion level > 22.5), the upper path

is followed. Tracing the conditions met using the data from Table 6, a terminal node with a TXOT of 24.3 minutes is reached. A TXOT prediction of 24.3 minutes is therefore made for flight ‘X’. Compared to the actual TXOT of 24.2 mins, a 7s difference is observed and is thus a very accurate prediction.

It is important to note that actual TXOT is almost 20 minutes longer than the UTXOT. To understand what causes this delay, the RT and respective data is analysed. Firstly, a relatively high congestion level, low throughput, and low departure count suggests longer queueing times. Additionally, a low cloud ceiling (100ft), coupled with a low dew point (-3°), temperature (-2°), and visibility (250m) indicates that it is snowing. As such, the aircraft will require de-icing resulting in a considerable departure delay.

Table 6 Available information for example flight ‘X’

Variable	Value	Variable	Value	Variable	Value	Variable	Value
Date	10/12/15	Dep RWY	R29	Uind	0	Day week	5
Flt. no	‘X’	AOBT	09:29:10	UTXOT (min)	5.242	Day month	10
AC Type	A320	ATOT	09:53:24	Wind dir. (°)	160	Day year	344
ICAO Cat	M	Con. lvl	18	Wind speed (kts)	4	Week month	2
MTOW (kg)	78000	Throughput	39	Visibility (km)	0.25	Week	50
Origin	LOWW	Dep. count	6	Ceiling (ft)	100	Month	12
Destination	-	U1	5.017	Temperature (°C)	-2	Peak	0
Dep STD	B82	U1 Group	5.083	Dew point (°C)	-3	TXOT	00:24:14
STD Group	AB	Sat. lvl	4.264	Hour	9		

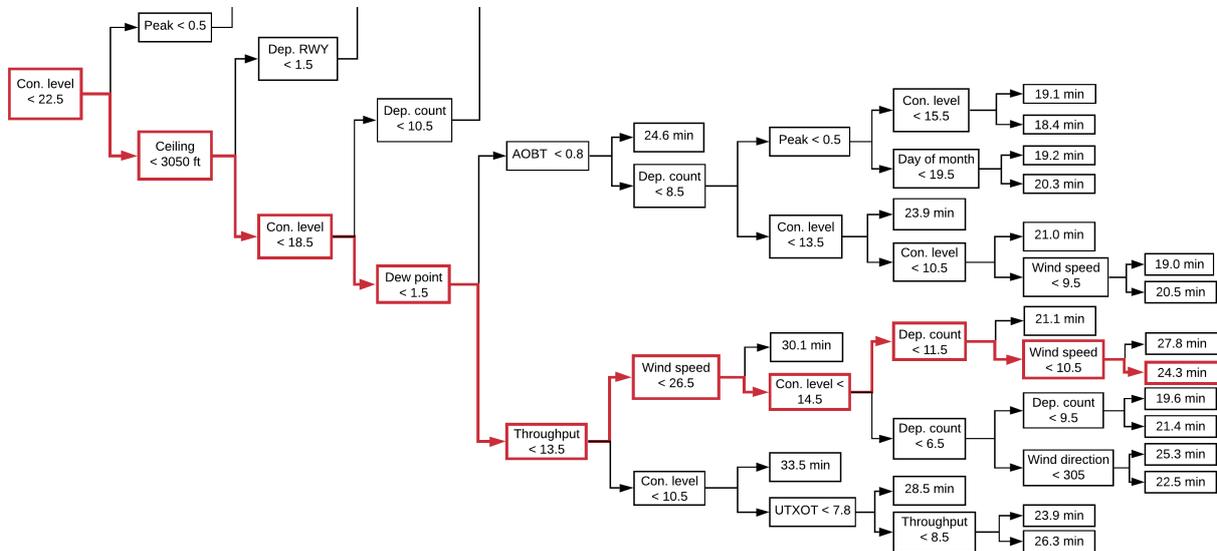


Fig. 15 Part of the RT for extended TXOT flights; the red path shows the decisions made to produce the TXOT prediction for example flight ‘X’

D. Top feature comparison

For comparison purposes the resultant top features identified for both normal and extended TXOTs are presented in Table 7. Congestion level is seen as the most influential factor in both cases. Other common features are number of prior departures and airport throughput which are both related to congestion level.

Table 7 Top feature comparison for normal and extended TXOTs

Rank	Normal TXOT	Extended TXOT
1.	Congestion level	Congestion level
2.	First estimation of the Unimpeded Taxi-Out Time per RWY-STD combination	Number of departures in the 20 minutes preceding the AOBT of the departing aircraft
3.	Number of departures in the 20 minutes preceding the AOBT of the departing aircraft	Cloud ceiling height
4.	Temperature	Airport peak hours
5.	Airport throughput in the hour preceding the AOBT of the departing aircraft	Airport throughput in the hour preceding the AOBT of the departing aircraft
6.	Unimpeded Taxi-Out Time (UTXOT)	Dew point
7.	Actual Off-Block Time (AOBT)	Departure runway

VII. Conclusion and Recommendations

It can be concluded that the optimised regression tree model developed can accurately predict the TXOT for the majority of departures from Vienna Airport based on 2015 full-year data. For normal TXOT prediction based on validation and testing using 17535 flights, the taxi-out times of the test set of flights are predicted with an RMSE of 2.03 minutes. The distribution of the predicted flights shows that flights with an actual TXOT of between 4-6 minutes are often overpredicted, making the model slightly conservative. The remainder of the flights are more accurately predicted. For extended TXOT prediction i.e. for flights more than 2 standard deviations from the mean of all TXOTs, based on validation and testing of 600+ flights, the test set taxi-out times are predicted with an RMSE of 3.75 minutes. The distribution of the predicted flights shows that the model often over-predicts the TXOT of flights with lower actual extended TXOTs (17.5 - 18.5 mins). While there is some noise in the rest of the predictions, they are closer to the actual TXOTs. This variability could be smoothed given more 'extended' TXOT flight data to train the model.

Use of the ML models highlighted the features which most significantly influence the prediction of both normal and extended TXOTs as summarised in Table 7. Based on these it can be concluded that, in line with the existing literature, the most significant factors affecting normal and extended TXOTs relate to congestion levels at the airport. It is interesting to note that in the case of extended TXOTs the weather plays a more significant role (e.g. cloud ceiling, dew point).

Recommendations for further research include deepening the understanding of outlier analysis and how e.g. a de-icing stand prediction variable could be included. Other features which could potentially be included for assessment include wake-vortex relationship between sequentially departing aircraft, stand/gate availability and private vs commercial operations. The prediction model could be refined further using different node-splitting algorithms for the regression tree. Additional testing at other, preferably busier, airports would enable further assessment of feature selection validity.

References

- [1] Ravizza, S., Chen, J., Atkin, J. A., Stewart, P., and Burke, E. K., "Aircraft taxi time prediction: Comparisons and insights," *Applied Soft Computing Journal*, Vol. 14, No. PART C, 2014, pp. 397–406. <https://doi.org/10.1016/j.asoc.2013.10.004>, URL <http://dx.doi.org/10.1016/j.asoc.2013.10.004>.
- [2] Herrema, F., Curran, R., Visser, H., Huet, D., and Lacote, R., "Taxi-Out Time Prediction Model at Charles de Gaulle Airport," *Journal of Aerospace Information Systems*, Vol. 15, No. 3, 2018, pp. 120–130. <https://doi.org/10.2514/1.i010502>.

- [3] Balakrishna, P., Ganesan, R., and Sherry, L., “Accuracy of reinforcement learning algorithms for predicting aircraft taxi-out times: A case-study of Tampa Bay departures,” *Transportation Research Part C: Emerging Technologies*, Vol. 18, No. 6, 2010, pp. 950–962. <https://doi.org/10.1016/j.trc.2010.03.003>, URL <http://dx.doi.org/10.1016/j.trc.2010.03.003>.
- [4] Idris, H., Clarke, J.-P., Bhuvu, R., and Kang, L., “Queuing Model for Taxi-Out Time Estimation,” *Air Traffic Control Quarterly*, Vol. 10, No. 1, 2017, pp. 1–22. <https://doi.org/10.2514/atcq.10.1.1>.
- [5] Clewlow, R. R., Simaiakis, I., and Balakrishnan, H., “Impact of arrivals on departure taxi operations at airports,” *AIAA Guidance, Navigation, and Control Conference*, , No. August, 2010, pp. 1–21. <https://doi.org/10.2514/6.2010-7698>.
- [6] Ravizza, S., Atkin, J. A., Maathuis, M. H., and Burke, E. K., “A combined statistical approach and ground movement model for improving taxi time estimations at airports,” *Journal of the Operational Research Society*, Vol. 64, No. 9, 2013, pp. 1347–1360. <https://doi.org/10.1057/jors.2012.123>.
- [7] Sundarapandian, V., *Probability, Statistics and Queuing Theory*, PHI Learning, 2009. URL <https://books.google.nl/books?id=9oUS6BBJkCYC>.
- [8] Capelleras, L., “Additional Taxi-Out Time Performance Indicator document,” Tech. rep., EUROCONTROL, 2015. URL <http://ansperformance.eu/references/library/pru-tx-out-pi.pdf>.
- [9] Zhang, Y., and Wang, Q., “Methods for determining unimpeded aircraft taxiing time and evaluating airport taxiing performance,” *Chinese Journal of Aeronautics*, 2017. <https://doi.org/10.1016/j.cja.2017.01.002>.
- [10] Hebert, J. E., and Dietz, D. C., “Modeling and analysis of an airport departure process,” *Journal of Aircraft*, 1997. <https://doi.org/10.2514/2.2133>.
- [11] Simaiakis, I., and Balakrishnan, H., “Queuing models of airport departure processes for emissions reduction,” *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2009.
- [12] Simaiakis, I., and Pyrgiotis, N., “An analytical queuing model of airport departure processes for taxi out time prediction,” *10th AIAA Aviation Technology, Integration and Operations Conference 2010, ATIO 2010*, 2010. <https://doi.org/10.2514/6.2010-9148>.
- [13] Simaiakis, I., and Balakrishnan, H., “A queuing model of the airport departure process,” *Transportation Science*, Vol. 50, No. 1, 2016, pp. 94–109. <https://doi.org/10.1287/trsc.2015.0603>.
- [14] Shumsky, R. A., “Dynamic Statistical Models for the prediction of aircraft take-off times,” *PhD Thesis*, 1995, pp. 1–214. URL <papers2://publication/uuid/95251702-60DD-48BC-AE91-042F332CC19B>.
- [15] Pujet, N., Delcaire, B., and Feron, E., *Input-output modeling and control of the departure process of congested airports*, American Institute of Aeronautics and Astronautics Inc, AIAA, 1999, pp. 1835–1852. <https://doi.org/10.2514/6.1999-4299>, URL <https://arc.aiaa.org/doi/abs/10.2514/6.1999-4299>.
- [16] Andersson, K., Carr, F., Feron, E., and Hall, W. D., “Analysis and Modeling of Ground Operations at Hub Airports,” *3rd USA/Europe Air Traffic Management R&D Seminar*, , No. June, 2000, pp. 13–16. <https://doi.org/10.1109/JSTSP.2013.2237882>.
- [17] Marko, R.-S., and Igor, K., “Theoretical and Empirical Analysis of ReliefF and RReliefF,” *Machine Learning*, Vol. 53, 2003, pp. 23–69. URL <http://lkm.fri.uni-lj.si/xaigor/slo/clanki/MLJ2003-FinalPaper.pdf>.

II

Report

Contents

I Article	iii
II Report	xxviii
List of Figures	xxiv
List of Tables	xxvii
Acronyms	xxviii
1 Introduction	1
2 Literature Study	2
2.1 Statistical Analysis	2
2.1.1 Machine learning techniques	2
2.1.2 Factors Influencing Taxi-Out Time	10
2.2 Queueing Theory	12
2.2.1 Unimpeded taxi-out time.	13
2.2.2 Analytical models.	15
2.2.3 Numerical models	19
2.3 Research questions	22
2.3.1 Research objective	22
2.3.2 Main research question.	22
2.3.3 Sub-questions	23
3 Methodology	24
3.1 Data Acquisition.	24
3.1.1 Key Statistics	26
3.1.2 Data Overview	26
3.2 Data Preparation	28
3.2.1 UTXOT Determination.	30
3.3 Prediction Variables	31
3.4 Feature Selection	32
3.4.1 RreliefF parameter determination	34
3.4.2 Selected features	35
3.5 TXOT Understanding	37
3.5.1 Additional prediction variables	38

CONTENTS

3.6	Neural-Network Levenburg-Marquardt (NNLM) feasibility	38
3.6.1	NNLM overview	39
3.6.2	Dataset stability.	39
3.6.3	Parameter selection.	40
3.6.4	NNLM results	40
3.6.5	NNLM results using feature selection variables	42
3.6.6	NNLM results comparison	44
3.7	Regression Tree (RT) feasibility	44
3.7.1	Validation scheme selection	44
3.7.2	RT results	45
3.8	ML technique comparison and selection.	46
3.9	Extended Taxi-out Times	48
3.10	Extended TXOT analysis	50
3.10.1	TXOTs greater than 50 mins	50
3.10.2	Extended TXOTs due to de-icing	56
4	Results normal TXOT	58
4.1	Normal TXOT analysis	58
4.1.1	Leaf Size Determination	58
4.1.2	Tree Depth determination	60
4.1.3	RT results for normal TXOT prediction.	60
4.1.4	Top feature extraction	61
4.1.5	Leaf size - top features	62
4.1.6	Tree depth - top features	63
4.1.7	Final tree results	64
4.2	Distribution fitting	66
4.2.1	Kolmogorov-Smirnov test	66
4.3	Top feature analysis.	70
5	Results extended TXOT	76
5.1	Regression tree setup	76
5.2	Top features	78
5.3	Final tree.	78
5.3.1	Example flight	80
5.4	Distributions.	81
5.5	Top feature analysis.	82
6	Verification and Validation	89
6.1	Verification.	89
6.2	Validation	91

CONTENTS

7 Conclusion and Recommendations	92
7.1 Conclusions	92
7.2 Recommendations	93
Bibliography	95
A ICAO Map - Vienna International Airport	97
B Gate Group Definitions	99

Summary

All major airport operators face a similar challenge, namely ensuring maximum throughput and maintaining high runway utilisation. A key part of this is accurately planning aircraft movements on the ground to avoid queueing and associated delays. This benefits the operator by enhancing their overall efficiency and hence their reputation with the airlines. It also benefits airlines by reducing their fuel costs and keeping their customers satisfied with faster transit times.

A primary indicator of the operator performance in this area is the Taxi-Out Time or TXOT. Studying literature revealed two main categories of methods used to model the departure process and predict TXOTs, namely, queueing theory or statistical analysis. One branch within statistical analysis that showed promise with respect to feasible TXOT prediction was machine learning. As such, this study sets out to review whether the application of machine learning can be used to model the departure process at Vienna International Airport in such a way as to provide accurate prediction of TXOT taking into account a wide range of variables. Enhancing the understanding of the influence of variables on the overall predicted TXOT is an additional beneficial outcome.

Using the data obtained from Vienna Airport, two separate machine learning techniques were tested, namely, the Neural Network Levenburg-Marquardt (NNLM) and Regression Tree (RT) techniques. A comparison of the techniques was performed where, based on several performance indicators, the RT was determined to be the most suitable technique for TXOT prediction. The RT is thus expanded upon, where the tuning parameters of RT model are optimised for normal TXOTs, and the most influential features extracted.

An additional study is performed to predict extended TXOTs (defined as flights with a TXOT more than 2 standard deviations from the mean) and extract the key related precursors. The regression tree technique is used here once again. The RT parameters are then re-tuned for extended TXOT prediction, and the most influential features extracted.

The taxi-out times of the test set of data are accurately predicted for normal taxi-out times, where an RMSE of 2.03 minutes is achieved with 91.32% of flights being predicted within 3 minutes. The features deemed to influence the TXOT most significantly are as follows: 'congestion level', 'unimpeded taxi-out time', 'no. of departures in the preceding 20 mins of the flight', 'temperature', 'airport throughput in the preceding hour of the flight', and 'actual off-block time'.

The taxi-out times of the test set of data for extended TXOTs are predicted with an RMSE of 3.75 minutes, with 84.6% of the flights being predicted within 5 minutes. The features which influence extended TXOTs most significantly are as follows: 'congestion level', 'no. of departures in the preceding 20 mins of the flight', 'cloud ceiling height', 'airport peak hours', 'airport throughput in the preceding hour of the flight', 'dew point', and 'departure runway'.

Summary

Recommendations are presented for further refinements to the models which would potentially increase reliability and accuracy, taking into account additional factors like de-icing requirements.

List of Figures

2.1	Example plot TXOT vs. congestion level [1]	3
2.2	Example regression tree with tree depth 4 [1]	7
2.3	RL functional block diagram [2]	10
2.4	Logical steps in Additional Taxi-out Time calculation [3]	13
2.5	Logical steps in Unimpeded Taxi-out Time calculation [3]	14
2.6	FAA APO method for determining unimpeded taxi-out time [4]	15
2.7	Departure process model as defined by Simaiakis et al. [5]	17
2.8	Definition of ground transit time (GTT) [6]	19
2.9	Structure of the departure process model [7]	20
3.1	Layout of Vienna International Airport	25
3.2	Departure ground track, ground speed, and flight level of an example flight departing from runway 29 - Vienna Airport	28
3.3	Flow chart of data preparation	29
3.4	Feature importance vs. no. observations	35
3.5	Feature importance vs. k-nearest neighbours	36
3.6	Normalized feature selection using RReliefF algorithm (see Table 3.3 for variable definitions)	36
3.7	TXOT breakdown	37
3.8	TXOT vs. congestion level for RWY-STD combination C34-R29	38
3.9	TXOT vs. congestion level for RWY-STD combination B92-R29	38
3.10	NNLM model overview [8]	39
3.11	NNLM model with chosen parameters	40
3.12	Epoch checks: best validation performance (MSE) is 3.235 at epoch 59	41
3.13	Error histogram of results	41
3.14	Regression plots of the training, validation, testing and combined data sets	42
3.15	Epoch checks for NNLM technique using top features only: best validation performance (MSE) is 3.625 at epoch 112	43
3.16	Error histogram of results using top features only	43
3.17	Regression plots of the training, validation, testing and combined data sets for NNLM model using top features only	43
3.18	Regression plot for testing data set using all features	46
3.19	Error histogram for tree trained with all features	46
3.20	Regression plot for testing data set using top features only	46
3.21	Error histogram for tree trained with top features only	46
3.22	Regression tree example	47
3.23	Threshold at 3 standard deviations (3σ) from the mean	49

LIST OF FIGURES

3.24	Threshold at 2 standard deviations (2σ) from the mean (17.3 mins)	49
3.25	Extended flight TXOTs and areas of interest	50
3.26	Extended TXOT outlier analysis	51
3.27	Ground track and speed profile for flight ‘A’	51
3.28	Ground track and speed profile for flight ‘B’	52
3.29	Ground track and speed profile for flight ‘C’	52
3.30	Ground track and speed profile for flight ‘D’	53
3.31	Ground track and speed profile for flight ‘E’	53
3.32	Ground track and speed profile for flight ‘E’	54
3.33	Ground track and speed profile for flight ‘F’	54
3.34	Ground track and speed profile for flight ‘G’	55
3.35	Ground track and speed profile for flight ‘H’	55
3.36	Ground track and speed profile for flight ‘I’ on 05-Nov-2015	56
3.37	De-icing standby and de-icing positions	57
4.1	Cross-validated error for varying number of folds k	59
4.2	Cross-validated error for varying minimum leaf size l_{min}	60
4.3	RMSE vs. tree depth for training data set and validation data set	60
4.4	Regression plot for testing data set using all features	61
4.5	Error histogram for tree trained with all features	61
4.6	Top features extracted from trained tree	62
4.7	Cross-validated error for varying minimum leafsize l_{min} using top features only	63
4.8	RMSE vs. tree depth for training data set and validation data set using top features only	63
4.9	RMSE vs. tree depth for validation data sets using top features and all features	64
4.10	Regression plot for final tree using top features only	65
4.11	Error histogram for final tree using top features only	65
4.12	PDF of actual TXOT vs predicted TXOT	65
4.13	CDF of actual TXOT vs predicted TXOT	65
4.14	Kolmogorov-Smirnov test	67
4.15	CDFs of empirical data and possible distributions for example node 1099	67
4.16	CDFs of empirical data and accepted distributions for example node 1099	67
4.17	Histogram of TXOTs and proposed distributions for example node 1099	68
4.18	Histogram of TXOTs fitted with a Gamma distribution for example node 1099	68
4.19	Gamma distributions for 60 of the 668 leaf nodes	69
4.20	Congestion level vs. TXOT and Congestion level histogram	70
4.21	1st Estimation UTXOT vs. TXOT and histogram of 1st estimation UTXOT	71
4.22	Departures in last 20 mins vs. TXOT and histogram of departures in last 20 mins	71
4.23	Temperature vs. TXOT and temperature histogram	72
4.24	Throughput in last hour vs. TXOT and histogram of throughput in last hour	72

LIST OF FIGURES

4.25	UTXOT vs. TXOT and UTXOT histogram	73
4.26	AOBT vs. TXOT and AOBT histogram	73
4.27	Wind speed vs. TXOT and wind speed histogram	74
4.28	Aircraft weight (MTOW) vs. TXOT and histogram of aircraft weight	74
5.1	Histogram of all 2015 flight TXOTs	76
5.2	Extended flight TXOTs	76
5.3	Leaf size vs. cross-validated error for	77
5.4	Caption	77
5.5	Regression plot for testing data set using all features	78
5.6	Error histogram for tree trained with all features	78
5.7	Top extended TXOT features	79
5.8	Regression plot for testing data set using top features only	79
5.9	Error histogram for tree trained with top features only	79
5.10	PDF of actual TXOT vs predicted TXOT	80
5.11	CDF of actual TXOT vs predicted TXOT	80
5.13	Ground track and speed profile for example flight 'X'	81
5.14	Birnbbaum-Saunders distributions for all 72 leaf nodes	82
5.12	Regression Tree for extended TXOT prediction with example prediction flight 'X'	85
5.15	Congestion level vs. TXOT and Congestion level histogram	86
5.16	Departure count vs. TXOT and departure count histogram	86
5.17	Cloud ceiling vs. TXOT and cloud ceiling histogram	86
5.18	Peak time boolean vs. TXOT and peak time histogram	87
5.19	Throughput in last hour vs. TXOT and histogram of throughput in last hour .	87
5.20	Dew point vs. TXOT and dew point histogram	87
5.21	Departure runway vs. TXOT and departure runway histogram	88
5.22	Wind speed vs. TXOT and wind speed histogram	88
5.23	AOBT vs. TXOT and AOBT histogram	88
A.1	ICAO Aerodrome Chart of Vienna Airport	98
B.1	Map of Vienna Airport: Gate groups - part 1. Obtained from [9]	100
B.2	Map of Vienna Airport: Gate groups - part 2. Obtained from [9]	101

List of Tables

2.1	Prediction and target variables as defined by Herrema et al. [1]	4
2.2	Feasible ML techniques assessed on performance indicators for runway 08L at CDG [1]	6
2.3	RMSE TXOT prediction within 3 and 5 min for 6 different models [1]	8
3.1	Number of flights per runway	26
3.2	Available data for Vienna International Airport	27
3.3	Prediction and target variables	33
3.4	Data set sizes	40
3.5	Results NNLM technique	41
3.6	Results NNLM technique using top features only	42
3.7	Comparison NNLM results	44
3.8	RT results for models trained using all features and top features only	45
3.9	Comparison ML techniques	47
3.10	Thresholds for ‘extended’ TXOTs	48
4.1	Results RT for normal TXOT using all prediction variables	60
4.2	Results RT for normal TXOT using only top prediction variables	64
4.3	RT result comparison	66
4.4	Results of Kolmogorov-Smirnov test for example node 109	68
4.5	Variance of distributions for example node 1099	68
4.6	% Best fits per distribution	69
5.1	Results RT for extended TXOT using all prediction variables	77
5.2	Results RT trained with top features only	78
5.3	Data for flight AUA601	81
5.4	% Best fits per distribution	82

List of Acronyms

AC	Aircraft	IFR	Instrument Flight Rule
ANSP	Air Navigation Service Provider	MDP	Markov Decision Process
AOBT	Actual Off-Block Time	METAR	Meteorological Terminal Air Report
APO	Aviation Policy and Planning Office	ML	Machine Learning
A-SMGCS	Advanced Surface Movement Guidance and Control System	MLP	Multilayer Perceptron
ASPM	Aviation System Performance Metrics	MSE	Mean Squared Error
ASQP	Airline Service Quality	MTOW	Maximum Takeoff Weight
ATA	Actual Time of Arrival	NNLM	Neural Network Levenburg-Marquardt
ATCO	Air Traffic Controller	QFU	Magnetic Orientation of Runway
ATOT	Actual Take-Off Time	RL	Reinforcement Learning
CDF	Cumulative Distribution Function	RMSE	Root Mean Squared Error
DSP	Departure Sequencing Program	RRF	RreliefF
EDCT	Expected Departure Clearance Time	RT	Regression Tree
EIBT	Estimated In-Block Time	RWY-STD	Runway-Stand
EOBT	Estimated Off-Block Time	TXOT	Taxi-Out Time
ETOT	Estimated Take-Off Time	UTXOT	Unimpeded Taxi-Out Time
FAA	Federal Aviation Administration	VFR	Visual Flight Rules
GS	Ground Stop	WMA	Wind speed meter
GTT	Ground Transit Time		
IATA	International Air Transport Association		
ICAO	International Civil Aviation Organization		

1

Introduction

All major airport operators face a similar challenge, namely ensuring maximum throughput and maintaining high runway utilisation. A key part of this is accurately planning aircraft movements on the ground to avoid queueing and associated delays. This benefits the operator by enhancing their overall efficiency and hence their reputation with the airlines. It also benefits airlines by reducing their fuel costs and keeping their customers satisfied with faster transit times.

A primary indicator of the operator performance in this area is the Taxi-Out Time or TXOT. This study sets out to review whether the application of machine learning can be used to model the departure process in such a way as to provide accurate prediction of TXOT taking into account a wide range of variables. Enhancing the understanding of the influence of variables on the overall predicted TXOT is an additional beneficial outcome.

This report covers a review of existing literature on modelling of the flight departure process, a description of the steps taken to select and develop a machine-learning approach based on data supplied by Vienna Airport, a detailed analysis of the results achieved and final conclusions and recommendations for further study.

2

Literature Study

The aim of this chapter is to present relevant literature relating to the airport departure process, and more specifically the prediction of taxi-out-time. The literature will be divided into two main groups, the first of which can be found in [section 2.1](#) and relates to statistical detection methods for the prediction of TXOT. The second group, found in [section 2.2](#), explores existing models that describe the departure process using queueing theory. Lastly, in [section 2.3](#), the research goals and questions of the thesis will be presented.

2.1. Statistical Analysis

In this section, methods based on the analysis of historical data will be explored. In [section 2.2](#), numerical and analytical models will be explored which attempt to describe the departure process by splitting the process into sections and modelling each section individually. Models based on statistical analysis, however, differ in the sense that they do not model the departure process in pieces, but rather as a whole. In the literature, two main categories of TXOT modelling using statistical analysis can be found. The first category is ‘Machine Learning’, in which historical data is analysed in order to predict taxi-out times. Additionally, ML techniques can be used to detect previously unobserved patterns which can be used to improve the accuracy of future predictions. The second category is ‘factors influencing taxi-out time’, in which authors attempt to model the process by determining which factors influence taxi-out time most significantly. An important step in machine learning is ‘feature selection’ in which only the most important factors which influence taxi-out time should be used to train the model. The two categories are therefore closely related.

2.1.1. Machine learning techniques

The first paper to be analysed was created by Herrema et. al [1]. The study presents a machine learning approach to predict taxi-out time (TXOT), used to cope with variability in aircraft behaviour at Paris-Charles De Gaulle (CDG) Airport. The methodology used to create the model comprises 5 steps which are as follows:

- TXOT computation
- TXOT understanding
- Data preparation
- Evaluation of feasible ML techniques

- Creation of a prototype model

The first step in the creation of the model was the computation of a TXOT indicator. This TXOT indicator provides a measure of the average outbound queuing time during congested periods. TXOT is defined as “the time elapsed between actual off-block time (AOBT), from a specific stand, and the actual take-off time (ATOT), on a specific runway” [1]. The TXOT includes factors such as queuing at the runway and additional time spans for specific procedures. Time spans linked to the actual progress of the operation were also included. To determine the TXOT indicator, plots of the TXOT vs. congestion level for all runway configurations at CDG were created, where the congestion level was defined as “the estimated number of movements at CDG within the estimated taxi-out transit of the respective flight” [1]. The definition for congestion level uses estimated values for number of movements and taxi-out transit time because the variables used to determine them were also estimated. The estimated variables used are as follows: estimated in-block time (EIBT), estimated off-block time (EOBT), and estimated take-off time (ETOT). An example of such a plot can be seen in Figure 2.1, where the red line represents the TXOT at low levels of congestion or the unimpeded TXOT (UTXOT). It can be seen that after a certain congestion level, the TXOT increases linearly.

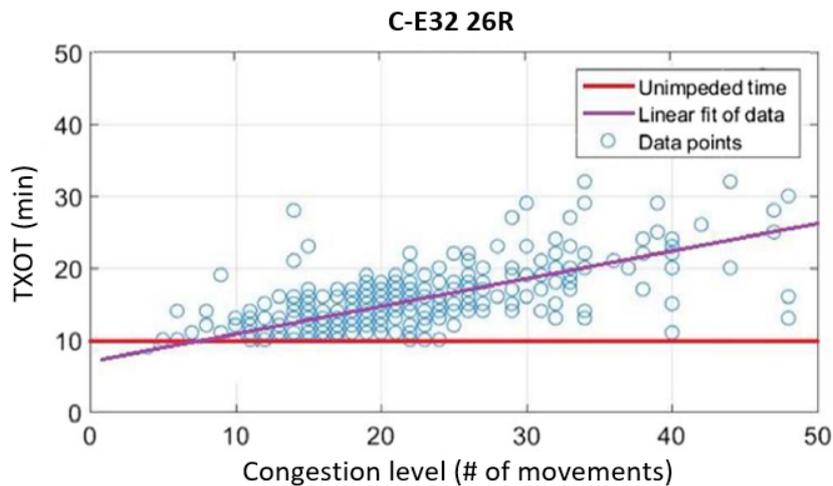


Figure 2.1: Example plot TXOT vs. congestion level [1]

The next step is the understanding of the TXOT. One task was to identify the factors that influenced the taxi-out time. Based on the data provided by CDG, 37 prediction variables were initially identified. An analysis was then conducted to extract an additional 5 prediction variables that affect TXOT behaviour. These variables were not included in the received historical operational taxi data and therefore had to be calculated. The identified prediction variables with the largest influence on TXOT also happen to be the additional variables and are as follows: congestion level, number of departures in last 20 mins, saturation level, unimpeded TXOT per runway stand and stand group, and stands gate availability. All 42 prediction variables related to the target variable TXOT can be found in Table 2.1.

The 3rd step in the methodology is the preparation of data. Given the initial raw aircraft operational taxi data, several activities were performed to set up the final dataset. This included the

Table 2.1: Prediction and target variables as defined by Herrema et al. [1]

Variables	Description
1. AOBT	Actual offblock time
2. ASAT	Actual startup time
3. ATOT	Actual takeoff time
4. Year	Year
5. DeIcingStand	Deicing stand
6. CTOT	Calculated takeoff time
7. Caractredevol	Commercial or private flight
8. CodeIATA	IATA code company
9. CodeAirportICAO	Airport destination ICAO code
10. CodeAirportIATA	Airport IATA code
11. Airline	Airline
12. DataPointSpeed	Point where speed is measured
13. DateReal	Actual date
14. DateSchema	Schema date
15. DepArr	Departure or arrival flight
16. EOBT/EIBT	Estimated offblock time/estimated in-block time
17. ETOT	Estimated takeoff time
18. DeIcingStatus	Deicing status
19. ScheduleBloc	AOBT or actual in-block time, depending on whether it is a departure or an arrival
20. SOBT/SIBT	Scheduled offblock time (SOBT) or scheduled in-block time (SIBT), depending on whether it is a departure or an arrival
21. ATCcallsign	Air traffic control (ATC) call sign
22. RegistrationCode	Registration code
23. Day	Day of the year
24. Month	Month of the year
25. FlightNumber	Flight number
26. DepartureStand	Departure stand
27. QFU	Aviation Q-code for magnetic heading of a runway
28. SOBT	Scheduled offblock time
29. TOBT	Target offblock time
30. TSAT	Target startup time
31. Terminal	Terminal departure
32. TimeATOT	Date and actual takeoff time
33. TimeSchema	Schema time
34. TimeReal	Actual time
35. AircraftType	Aircraft type
36. AircraftTypeICAO	Aircraft type ICAO
37. GateConnectionType	Connection type at the gate
38. CongestionLevel	The congestion level is the estimated number of movements (i.e., arrivals and departures) during the estimated taxi-out transit time (i.e., time between the estimated offblock and estimated takeoff) of the respective flight.
39. NumberOfDepartures	Number of departures in last 20 min observed when a flight is at ATOT on a specific runway.
40. UnimpededTXOT	There are two sets of unimpeded values. The first set is the unimpeded time per runway–stand combination, and the second set is the unimpeded time per stand group.
41. SaturationLevel	For explanation, see Sec. IIIA
42. StandsGate Availability	When a flight is at AOBT, the number of unused stands and gates are counted
43. TXOT	Taxi-out time

merging and cleaning of the taxi data which was obtained from the recorded runway scheduler data provided by CDG. The data covers 5 years of taxi-out and taxi-in records from 2011 – 2015 and comprises records of roughly 1,000,000 arrival and departure flights.

As part of the data preparation phase, feature selection also had to be performed. Feature selection is used to identify the most important of the features in [Table 2.1](#) for describing the variance in a dataset. Before a model is trained, feature selection methods are applied in a pre-processing step, of which RreliefF and Sequentialfs are two such techniques chosen by the author. As stated in [1], feature selection techniques aim to achieve the following:

- Improve the prediction performance of the predictors.
- Provide faster computational performance and more effective predictors.
- Provide a better understanding of the underlying process that generated the data.

Having applied these techniques and applying the “intersection” method on the results, the 10 most important features were found. These are as follows:

- Unimpeded TXOT
- Congestion level
- Saturation level
- No. departures in the last 20 min
- De-icing stand
- Month
- Actual time
- Departure stand
- Magnetic orientation of runway (QFU)
- Actual off-block time (AOBT)

Due to reasons explained by the feature selection objectives, only these variables were included in the developed ML model. In [subsection 2.1.2](#), more papers regarding the influence of features on TXOT are explored.

Step 4 involves the evaluation of feasible machine learning techniques. In order to produce an accurate TXOT ML model, the Multilayer perceptron (MLP), Regression tree (RT), Reinforcement learning (RL), and Neural Network Levenburg-Marquardt (NNLM) modelling techniques were applied and assessed based on several performance indicators.

As an example, the NNLM technique was first modelled. The model was trained with all the prediction variables listed in [Table 2.1](#). It was found that similar mean squared error (MSE) results were obtained using only the top 10 features as previously mentioned. Additionally, by excluding 32 variables, the model was trained 3 times faster, and was more robust when inserting new data with a similar structure. The ML TXOT prediction error results also show similar statistical TXOT prediction errors. The NNLM technique resulted in a root mean squared error (RMSE) of 1.97 min for 79% of the cases, and an RMSE of ≈ 5 min for 98% of the cases.

The same procedure performed for the NNLM technique was repeated for the RT, MLP, and RL techniques. As shown in [Table 2.2](#), a comparison of the minimum TXOT RMSE, the computational time, and amount of data needed to obtain the results was performed. In terms of RMSE, the RT and RL techniques were observed to perform best. However, the RT technique was selected as the

2.1. STATISTICAL ANALYSIS

most efficient method for TXOT prediction due to a lower computational time and the amount of data needed.

Table 2.2: Feasible ML techniques assessed on performance indicators for runway 08L at CDG [1]

Technique	Performance indicators				
	RMSE, min	Computational time, min	Feasible for real-time computations	Distributions associated with the individual TXOT predictions	Amount of TXOT data needed (flights)
Levenberg–Marquardt	01:58	01:06	Yes	Yes	70,000
Regression tree	01:36	01:20	Yes	Yes	70,000
MLP	01:42	01:50	Yes	Yes	110,000
Reinforcement learning	01:36	01:30	Yes	Yes	150,000

The purpose of creating a regression tree is to obtain a set of what-if statements which identify the precursors that have the greatest influence on TXOT. Starting at the root node, a series of questions are asked about the predictors. In each subsequent node, the variable and split point which achieves the minimum MSE between predictions and the actual TXOT is then selected by the tree. The process is then repeated until a stopping rule is applied and a terminal leaf is reached.

To ensure the model generates predictions that are both accurate and interpretable, the optimal tuning parameters for the tree had to be found. In [1] two parameters were used, namely, minimum leaf size l_{min} , and maximum tree depth d_{max} . In order to create a distribution for l_{min} , sufficient data points in each terminal node are required. This parameter is used to stop the splitting process when there are not enough occurrences in a leaf. Conversely, the tree may be difficult to interpret if it contains too many variables. To select the minimum leaf size cross validation was used. Regarding the 2nd tuning parameter, d_{max} , if a tree is very large and has many too many leaves it may overfit the data. A small tree, on the other hand, might not be able to capture the structure of all the variables or top 10 features. Therefore, the maximum tree depth was used to restrict the number of layers in the tree. To select the maximum tree depth, the set of parameters that resulted in the lowest MSE was used.

The model was first trained with all 42 variables and different settings of d_{max} and l_{min} . It was observed that as the tree depth increases, the MSE decreases, regardless of the leaf size. However, the MSE did not change significantly after a tree depth of 6. Regarding leaf size, it was observed that model performed better with a smaller leaf size. However, setting the leaf size to less than 4000 may have resulted in insufficient data point to fit a distribution. A value of 4000 for the leaf size was therefore chosen.

Next, the model was trained using only the top 10 features previously mentioned, where the tree was fitted to the entire dataset with d_{max} and l_{min} set to 6 and 4000 respectively. From this, a new set of predictors was obtained, of which the top top 10 features were selected as the final predictors. The tree was then re-trained with these 10 variables. The values of d_{max} and l_{min} were tested, and the cross-validation repeated. Once again, the tree with a leaf size of 4000 and tree depth 6 performed the best. Thus, the final model assume these values.

By training the tree, a mean and distribution could be extracted per decision node. This was needed to observe precursors and understand what was likely to happened for the TXOT. As stated in [1], the main factors that influence the TXOT were identified as follows:

2.1. STATISTICAL ANALYSIS

- Unimpeded Taxi-out – Does the flight experience unimpeded conditions? This is the most important predictor in the model. A flight is considered unimpeded if the observed TXOT remains sufficiently low for that RWY-STD.
- Congestion level - mainly influenced by the time of day and RWY-STD; estimated in real time using the EOBT, ETOT, and the EIBT within the estimated taxi-out transit time.
- Saturation level – predicted based on the top 10 feature and estimated congestion level.
- Time and Month – during 3 periods of the day, the probability of experiencing congested conditions is significantly higher than other time windows. Cold conditions or winter conditions can also negatively influence the TXOT.
- No. departures – assessed in the last 20 mins; measured before the ATOT and estimated using the ETOT.

In Figure 2.2 an example regression tree with tree depth 4 showing what-if statements can be seen. Starting at Node 1, if the statement is true, the path to the left will be followed, where a node on the next level is encountered; if the statement is false, the path to the right is followed. A parametric distribution was also fitted to each terminal leaf. Gumbel, Gamma, and F distributions were amongst the probability distributions considered. The following equation shows the Gumbel distribution, which provided the best fit over the terminal leaves:

$$f(x) = \frac{1}{\beta} e^{-((x-\mu)/\beta)} + (e^{-(x-\mu)/\beta}) \quad (2.1)$$

for $-\infty < x < \infty$, where $0 < \mu$ and $\beta < \infty$.

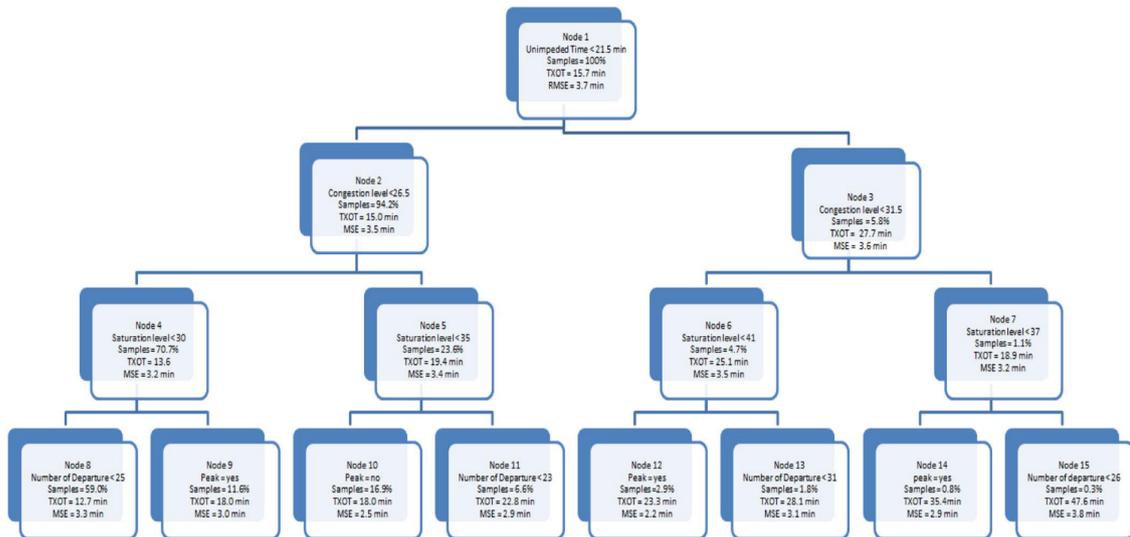


Figure 2.2: Example regression tree with tree depth 4 [1]

The final part of the methodology was the creation of the prototype model. The prototype model was developed using the RT method to forecast TXOT and CDG airport. It was built on 500,000 flights collected over 3 years of operational taxi data. An application was then developed to generate real-time predictions based on the proposed model. The aim was to "generate TXOT forecasts for

2.1. STATISTICAL ANALYSIS

Table 2.3: RMSE TXOT prediction within 3 and 5 min for 6 different models [1]

	Within 3 min	Within 5 min	Median TXOT, min
Charles de Gaulle Airport	94.2%	99.0%	17
Tampa International Airport	89.9–95.7%	— —	12
Stockholm-Arlanda Airport	96.1%	99.2%	11
John F. Kennedy International Airport	— —	20.7–100%	37
Detroit International Airport	89.9–97.1%	— —	15
Zurich Airport	95.6%	99.4%	14

each flight and of the number of aircraft assigned to a given runway per time window”. [1]

While there were no existing TXOT prediction functions available to validate the model, several function have been built to provide predictions at other airports. In Table 2.3, a table of baseline models was built to compare the CDG TXOT predictions against. The results of the case study include the final model with 10 predictors, a maximum tree depth of 6, and a minimum leaf size of 4000. The metric used to compare the TXOT predictions is the percentage of predictions that occur within 3 and 5 minutes. Table 2.3 shows the RMSE TXOT prediction results for six different models. However, as mentioned in [1], it should be noted that the following points are not taken into account in the models:

- The layout of the airport - particularly by not considering the factors associated to the distances and the turning angles.
- The time period and amount of data needed to learn.
- The computational time needed for a prediction.
- The instances at which the prediction is performed.

As can be seen in the Table 2.3, the RT model correctly predicted the TXOT of around 94% of the flights within 3 minutes and 99% of the flights within 5 min at CDG airport whilst considering both arrivals and departures simultaneously. While the Arlanda and Zurich airport cases performed slightly better with respect to the average RMSE, these models were much less complex in terms of operations. In [1] over 250,000 flights were used to train the model whereas the other models used data from only a day or a week of operation to train them.

While the RT model is complex and has captured the behaviour of many TXOT processes, several features have not been taken into account which could improve the accuracy of future prediction. These features include runway/taxiway repair, maintenance downtime, and weather events. Additionally, airport operational rules, regulations, and standards can vary significantly over time and should be taken into account for future predictions.

While the model can of course be improved, several advantages associated with the model exist. Firstly, the ML technique used to model the TXOT procedure was fast, accurate, and interpretable. In the future, similar models could help airport managers understand which feature affects TXOTs

per runway–stand most significantly. Secondly, the model was built based on a dataset of roughly 500,000 flights. 42 variables were also available for selection as certain predictors enabled new features to be built. Predictions could also be made in real-time due to the fast computational time. Additionally, the model provides TXOT predictions for each flight to a specific runway. These predictions may in the future help ATCOs make better decisions based on whether or not the aircraft will experience an additional TXOT. Finally, while the model was developed for TXOT prediction, the authors believe that the methodology proposed in their study can be easily applied to other runway processes such as predicting runway occupancy times.

Herrema et al. [1] provide a detailed overview of several different possible ML techniques for predicting TXOT. However, the techniques explored are not the only possibilities. Ravizza et al. [10] present a paper which tests different statistical regression approaches and various machine learning techniques to more accurately predict taxi-out times. The methods tested include least median squared linear regression, support vector regression, multiple linear regression, M5 model trees and two different fuzzy rule-based systems. Like the RT model mentioned in [1], fuzzy rule-based systems use if-then statements and are used to combine mathematical models with human knowledge. Like their previous paper [11], data from ARN and ZRH is used to make the predictions.

A comparison between the different models is performed. Metrics such as the prediction accuracy, mean-absolute error, root mean-squared error, and relative-absolute error are used to gauge the performance of the models. The prediction accuracy indicates what percentage of flights are correctly predicted within a certain timeframe. The comparison shows that ML fuzzy rule-based models provide the best results.

Another paper, written by Balakrishnan et. al [2], uses reinforcement learning (RL) algorithms to predict taxi-out times at Tampa International Airport (TPA). The paper suggests that the decision-making process made by air traffic controllers regarding departures and arrivals can be perceived as a stochastic control problem. Uncertainties in the process arise due to delays in taxi-time caused by congestion, weather and the probabilistic nature of the arrival and departure demands. Furthermore, the airport system can be modelled as a Markov chain, in which the system state is observed every minute. The combined system state process and the decision process form a Markov Decision Process (MDP), in which actions and rewards are added to the regular Markov chain.

In this case, the action is the predicted taxi-out value and the reward is the absolute error between the actual and the predicted taxi-out time. An analysis of data and literature resulted in the following state variables for taxi-time prediction: runway queue length, number of departing aircraft taxiing out at the same time, number of arrival aircraft in the taxiway system, average taxi-out time of the last 30 minutes, and the time of day. The average taxi-out time in the last 30 mins incorporates changes in taxi-times due to factors such as changing weather conditions and runway configuration.

In [Figure 2.3](#), the model block diagram shows the learning process of the RL algorithm. The goal of the algorithm is to maximise the utility reward function. The model was trained with three months of data, after which one week of operations was simulated.

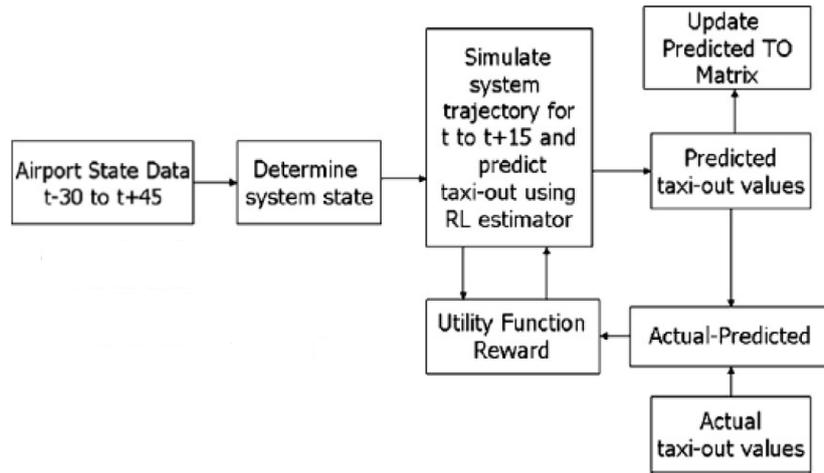


Figure 2.3: RL functional block diagram [2]

It was concluded that the RL algorithm is a suitable technique used to model the airport departure process. While this technique is more complex than others, the model can effectively capture trends in TXOT using real-time data from the last 30 minutes. Since it is often difficult to establish trends at airports without the use of recent data, this real-time data can be especially beneficial in airports which often experience large uncertainties.

2.1.2. Factors Influencing Taxi-Out Time

One paper that thoroughly analyses the influence of different factors on taxi-out was written by Idris et al. [12]. The method used was to identify the main factors that influence taxi-out time and build an estimation model that takes the most important ones into account. The main factors identified were the runway configuration, the airline/terminal, the downstream restrictions, and the take-off queue size, of which the take-off queue size was deemed to be the most important.

The first factor analysed was the runway configuration which is the assignment of runways to arrivals and departures. As such, it determines the flow pattern on the airport surface and in its surrounding space. The taxi-out time varies for different runway configurations for several reasons. These reasons include varying levels of interaction between the arrival and departure flows, different distances between gates and active departure runways, and different amounts of queuing and congestion due to the imbalance between the arrival/departure demand and the arrival/departure capacities of the runway configurations. In section 2.2, this complexity is acknowledged, which is why queueing models are often developed for one runway configuration only.

Another factor analysed is the distance between the gate from which the aircraft pushes back and the runway from which it takes off. In many cases specific gate information is not known and since the same gates or group of gates are often used by airlines, the airline information is used instead. A linear regression analysis is performed to determine the correlation between the airline and the taxi-out time in a specific runway configuration, resulting in an R^2 -value of 0.02. While this indicates that distance is a positive factor, it does not significantly account for the variability in the taxi-out time.

The influence of weather and downstream restrictions was also analysed. Weather reduces the

capacity of the airport system by impeding the flow through weather-related impact resources, such as runways and exit fires [12]. The reported weather forecast for each day and the reported meteorological conditions in terms of Visual Flight Rule (VFR) and Instrument Flight Rule (IFR) at the airport was used to determine a correlation between the weather and taxi-out time. However, a strong correlation was not found. The most indicative measure of the weather factor were the downstream restrictions. These are flow management programmes imposed on the departure traffic heading to weather-impacted destinations. Types of restrictions include Ground Stop (GS), Expected Departure Clearance Time (EDCT), Departure Sequencing Program (DSP), and In-Trail restriction. Idris et al. concluded that the taxi-out time and its variability increased significantly for aircraft affected by these restrictions.

As previously mentioned, given a certain runway configuration, Idris et al. [12] conclude that the departure demand and queue size is the most important causal factor for long taxi-out times. As expected, the paper shows that average taxi-out time of an aircraft increases when the number of aircraft in the taxiway system increases. However, the R^2 -value of the regression analysis between these two variables is 0.1927 meaning there is not a strong correlation between them, and that the number of departure aircraft on the airport surface does not accurately measure the size of the take-off queue that the aircraft faces. According to [12], this is primarily due to the passing between aircraft that takes place on the airport surface. These passings may occur if, for example, an aircraft is an emergency and must be expedited, has an assigned take-off time, or if some of the existing aircraft are suspended. They may also be allowed due to the different distances between gates and departure runways and due to the sequencing strategies of the air traffic controllers.

Other influencing factors analysed included arrival demand, measured by the number of arrivals, and the aircraft type. Surprisingly, neither of these were significant factors in affecting taxi-out times, with their R^2 -values being 0.01 and 0.02, respectively. Interestingly, in a paper written by Clewlow et al. [13] which analyses the impact of arrivals on departure taxi operations, the opposite conclusion is made. In this paper, the number of arrival aircraft has an R^2 -value of 0.71 which accounts for 71% of the variability in taxi-out times, nearly as much as the variability caused by the size of the departure queue. Other important factors include runway configuration, weather, and originating terminal. It is important to note that in [12], the departure runway is modelled as a departure runway only, whereas the runways in [13] are modelled as mixed-mode operations. This is likely the main cause of the discrepancy.

The paper presented by Clewlow et al. [13] uses the same data set as that of [12]. As such, the results of [12] can be verified. Clewlow et al. [13] suggest that a factor which significantly impacts the TXOT prediction is the definition of the number of arrivals. In [12], the number of arrivals is defined as being the number of aircraft taxiing-in when aircraft i is pushed back from the gate. In this case, an R^2 -value of 0.02 is obtained. This poor R^2 -value is obtained since this definition disregards arrivals that take place after push-back and before take-off of aircraft i . In [12], an alternate definition for the number of arrival aircraft was proposed, namely: the number of aircraft that arrive at their gate while aircraft i is taxiing out. Using this definition, R^2 -values of 0.677 and 0.747 were achieved at Boston Logan International Airport (BOS) and John F. Kennedy

Airport (JFK), respectively. While more representative of real life, implementing this definition into the model requires additional computational time due to its complexity.

Clewell et al. [13] also perform an analysis of the definition of the number of departures, in which they assess which definition influences the taxi-out time most significantly. Given the definition of the number of departures is the number of take-offs that occur between pushback and take-off of aircraft i , R^2 -values of 0.638 and 0.759 are obtained for BOS and JFK respectively. In order to implement this definition, however, knowledge of the order of which aircraft take-off is assumed. For example, if an aircraft j has an earlier pushback and a later take-off than aircraft i , it is assumed the aircraft do not interfere. In reality, this is not the case.

Lastly, Ravizza et. al [11] use multiple linear regressions to identify the most relevant factors affecting taxi-in, and taxi-out time. This paper differs from those explored thus far as the model is created based on European airports, rather than US airports. The airports concerned are Stockholm-Arlanda Airport and Zurich Airport, of which one day of data is analysed from each. The paper focuses on predicting the average taxi speed, from which the taxi-time can be derived.

Regarding factors influencing the taxi-out time, the first factor to be analysed was the taxi distance. To calculate the taxi distance, the airport layout was modelled and the shortest path was assumed to be the distance travelled. It was observed that if an aircraft had a longer taxi distance the average taxi speed was higher. The impact of the total number of turns performed by an aircraft was then analysed. This factor had a significant impact on the prediction performance of the model since aircraft need to slow down to make a turn, resulting in longer taxi-times. Lastly, in agreement with [12] and [13], the amount of traffic was also found to be of great importance, where only 13% of the variability was not explained by the model. The model is unique due to the detailed information used for the airport-layout. While the model performs well using a single day of data, it is unclear how it would perform over a longer time-span.

2.2. Queueing Theory

In this section, papers that model the airport departure process using queueing theory are explored. Queueing theory is the mathematical study of waiting in lines or queues, in which models are constructed so that queue length and waiting time can be predicted [14]. With respect to the airport departure process, queueing theory can be used to determine the extra taxi-out time due to the waiting time at the runway entrance. In [12], the authors found that the main factor influencing the taxi-out time was the take-off queue size. The uncertainty can therefore largely be explained using queueing theory.

To model the departure process using queueing theory, it is useful to divide the process into smaller sections. The simplest division that can be made is by dividing the process into a travel time and a queue time. To obtain a more accurate estimation, the travel time can be further split into unimpeded taxi-out time and additional taxi-out time. Currently, most literature uses unimpeded taxi-out time, additional taxi-out time, and queue time to model the airport departure process. Each paper, however, uses a different method. The purpose of this chapter is therefore to highlight these

differences.

In subsection 2.2.1, different methods used to calculate unimpeded taxi-out time are explored. Next, methods used to describe the complete departure process are described. This includes various analytical models, found in subsection 2.2.2, as well as several numerical models, found in subsection 2.2.3.

2.2.1. Unimpeded taxi-out time

To achieve an improved estimation of the travel time, a distinction should be made between the unimpeded taxi-out time and additional taxi-time. For this reason, many papers feature the calculation of an unimpeded, or nominal taxi-out time. However, literature suggests there are several ways to calculate this. To complicate matters, their definitions of unimpeded taxi-out time are not always the same. In this section, these definitions and calculation methods will be explored.

One method used to calculate unimpeded taxi-out time was developed by EUROCONTROL [3]. The primary purpose of the paper is to describe the conceptual, logical, and mathematical model of an additional taxi-out time performance indicator. This additional taxi-out time indicator is needed to provide a reliable measure of the average outbound queuing time during airport congestion. Knowing this, the efficiency of the departure process, and thus the operational costs associated with sequencing can be assessed. As shown in Figure 2.4, in order to produce this performance indicator, the unimpeded taxi-out time first had to be calculated.

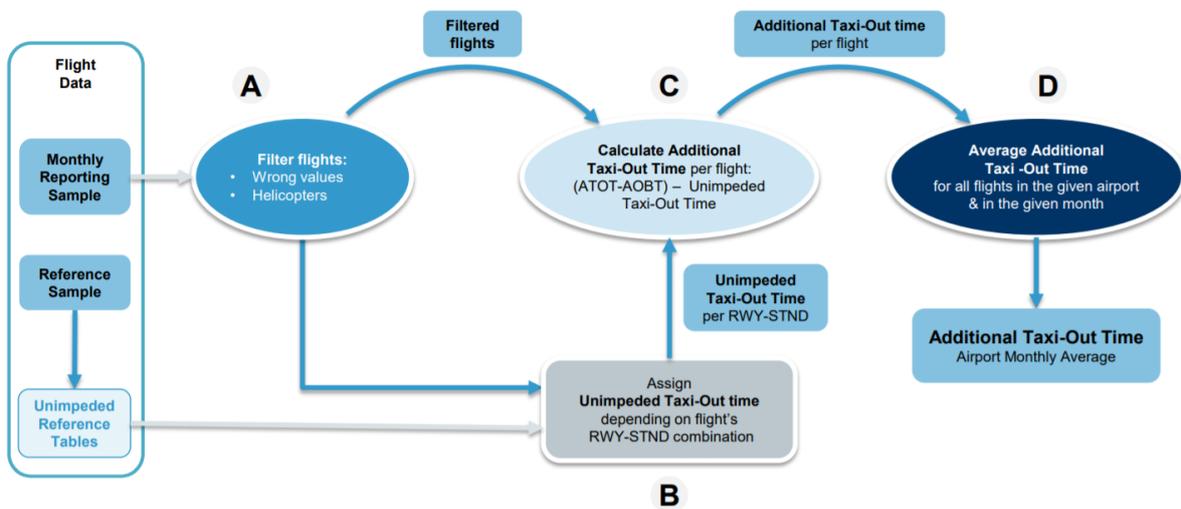


Figure 2.4: Logical steps in Additional Taxi-out Time calculation [3]

According to EUROCONTROL [3], the unimpeded taxi-out time (UTXOT) corresponds to the taxi-out time that an aircraft of a given Departure runway – Departure Stand (RWY-STD) combination would spend if no additional sequencing time was added, i.e. if the operation was unimpeded. In order to calculate the UTXOT, the steps shown in Figure 2.5 were followed and are described below.

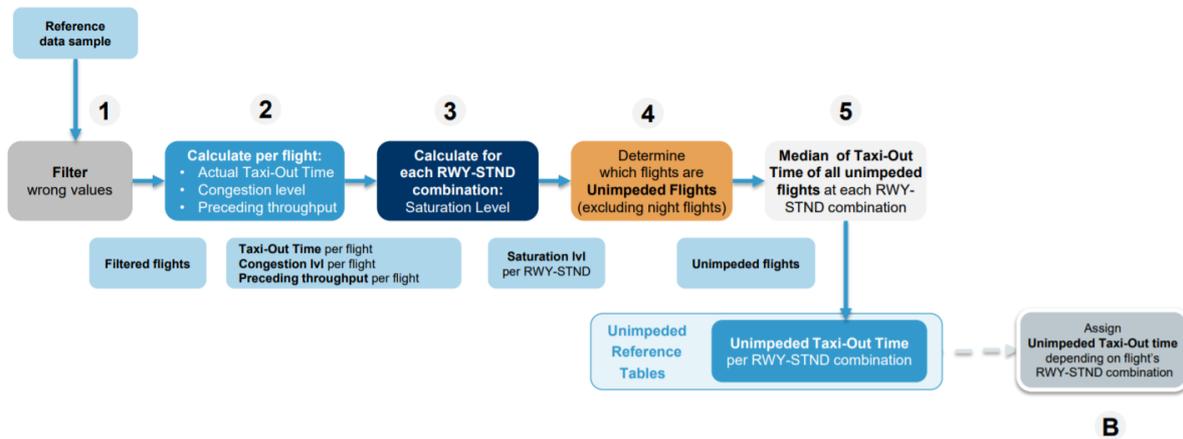


Figure 2.5: Logical steps in Unimpeded Taxi-out Time calculation [3]

The first step is to filter the outliers in the data. The sample used for calculations was one year of data, in which only flights with a taxi-out time of less than 300 minutes were considered. Next, the actual taxi-out time and the congestion level for each flight were calculated. The congestion level depends on the runway throughput as well as the number of arriving and departing flights in the last hour. The 3rd step is to calculate the saturation level for each RWY-STD combination. The saturation provides an indication of the maximum number of aircraft per hour that can be served in non-congested periods. Next, the unimpeded flights are determined, where a flight is considered unimpeded if the congestion level is lower than the congestion limit times the saturation level for its RWY-STD combination. For major hubs the congestion limit is 0.6, and for all other airports the congestion limit is 0.5. Finally, given that there are at least 10 flights in the sample, the unimpeded taxi-out time is calculated by taking the median taxi-out time of all unimpeded flights for each RWY-STD combination. The UTXOT is not calculated for groupings with less than 10 unimpeded flights.

A second method of estimating unimpeded taxi-out time was established by the FAA Aviation Policy and Planning Office (APO) [4]. The process aims to build a numerical relationship between the number of aircraft on the ground and taxiing time through a linear regression model. The model uses variables for taxi-in and taxi-out queue length and is based on 2 linear equations for the taxi-in and taxi-out times. Other model inputs are derived from the Aviation System Performance Metrics (ASPM) database. The steps taken to calculate the unimpeded taxi-out time are displayed in Figure 2.6.

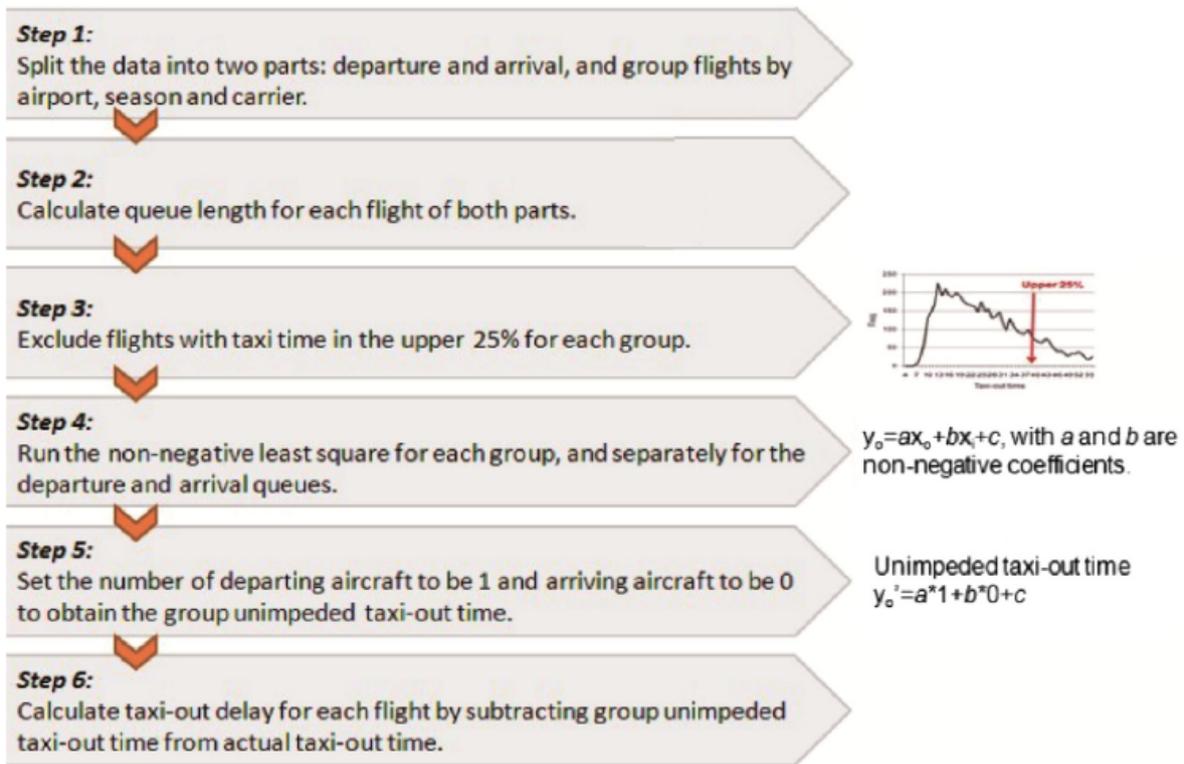


Figure 2.6: FAA APO method for determining unimpeded taxi-out time [4]

This method, however, has several limitations. Firstly, the parameters recorded in this method are a gate-out time and a wheel-off time. Therefore, the taxi-out time measured is the travel time of an aircraft from gate-out time to wheel-off, rather than from pushback to take-off. These values are therefore used as surrogates as an aircraft may spend a considerable time within the apron even after a gate-out message is triggered. Additionally, this method only applies to airline service quality performance system (ASQP) carriers – other airlines at airports are assigned an average taxi-out value. Finally, this method does not include other contributing factors such as runway configuration, gate location, or weather conditions. Another simple method of calculating unimpeded taxi-out time is the 20th percentile method (P20) [4]. Here, a cumulative distribution function of taxi-out times for each group of flights, grouped by airline, season, and runway configuration is constructed. The unimpeded taxi-out time is then determined by using the actual taxi-out time value at the 20th percentile.

Several other methods of calculating unimpeded taxi-out time also exist in literature. However, as with [3], these methods are often performed as a part of the process for modelling the entire departure process. As such, the remaining methods for determining unimpeded taxi-out times will be explored in subsection 2.2.2 and subsection 2.2.3, where entire departure models are explained.

2.2.2. Analytical models

In this section, several analytical models describing the entire departure process will be explored. One paper, written by Hebert et. al [15], aims to model the departure process at LaGuardia Airport based on data collected for two days in June 1994. Several definitions are useful in formulating an

appropriate queueing model. As these definitions are often different to those in the EU, it is important to understand them.

- Service completion i.e. departure: occurs when aircraft completes take-off and clears the runway environment sufficiently for another aircraft to be granted take-off clearance.
- Service demand time i.e. travel time: time at which the aircraft enters the departure queue after leaving the passenger gate (pushback).
- Departure delay i.e. queue time: difference between service demand time and the initiation of the service (clearance for take-off).
- Roll-out time i.e. taxi-out time: total time between pushback and take-off clearance (sum of taxi time and departure delay.)

The departure process is divided into two steps, namely, service demand and departure delay. Since service demand time is not recorded in the data set, push back times (plus nominal taxi time) can offer a reasonable surrogate. To obtain an estimate for nominal taxi time, it was assumed that a departure queue does not exist in lull periods where roll-out times appear stable. Here, the average time between pushback and take-off clearance represents the approximate taxi time. Assuming any delay caused by taxi-way congestion is relatively insignificant, this taxi time can be used to translate pushback times to service demand times. Since the number of pushbacks varies throughout the day, the service demand time is modelled by a non-homogeneous Poisson process. The intensity function varies per hour and is equal to the mean number of pushbacks in said hour.

To model the departure delay (queue time), three different models were formulated and evaluated. The first model is an exponential model, in which the service times are represented by iid random variables with an exponential distribution. While the model shows a reasonable fit on both days, it often under or overestimates the service time.

The second model is an Erlang-k model in which all service times are represented by iid Erlang-k random variables. Given that μ is the mean service rate, each service rate is determined as the sum of k exponentially distributed stages with mean completion rates $k\mu$. In this case, 2 stages are used, and is therefore an Erlang-2 model. While the Erlang-2 model does under or overestimate the service time multiple times throughout the day, it appears superior in matching the observed roll-out times when compared to the exponential model.

The third model is an Erlang-k model with server absences. When analysing the time between two consecutive take-offs, a separation of 1, 2, or 3 minutes was normally observed. However, in certain cases, a separation of between 4 and 10 minutes was observed. These cases indicate a server absence, in which the runway is not available due to arrivals or other external factors. In the model, a server absence is experienced with a probability of $p = 0.2$. While the inclusion of server absence may better represent reality, the results do not indicate an improvement in comparison to the Erlang-2 model. An explanation for this is that during the day the probability of runway availability is assumed constant, while in reality it depends highly on the number of arrivals and other influencing factors.

While the model appears to function well on the days tested, it is unclear how it would perform on other days as the model was built using the data from the same two days. Since very few studies

regarding departure delay and queue length prediction had been performed at the time of writing, it was difficult to validate the model. Additionally, the runway service rate used was derived from data of these days whereas this information would not normally be available at the start of the day. The model did however provide promise for future investigation due to its flexibility. For example, if the LGA data set reflected all scheduled departures, the service time distribution would likely be more symmetric and have a lower variance. This distribution could then be captured by increasing the number of stages in the Erlang model. This flexibility suggests that the model could also be applied to other airports.

A more recent and detailed analytical model was created by Simaiakis et al. [5]. Like many other models, this model divides the departure process into the travel time and the queue time. An illustration of this can be seen in Figure 2.7. The travel time consists of the unimpeded taxi-out time plus a linear term which represents the delay due to ramp and taxiway interactions with other aircraft on the way to the departure queue. Logically, this delay increases as the number of departing aircraft increases. For simplicity, the unimpeded taxi-out time distribution is determined for each airline at the airport for given visibility conditions and runway configuration. The data points in these distributions consist only of points where less than five aircraft are present in the taxiway system, such that interactions are minimal. A log-normal distribution is then fitted with the empirical distributions.

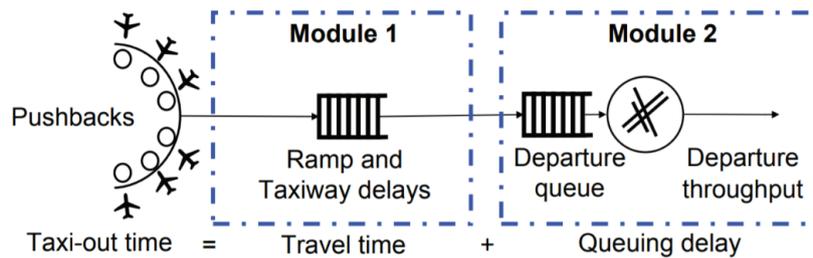


Figure 2.7: Departure process model as defined by Simaiakis et al. [5]

Module 2 describes the queueing delay estimation. Here, the service rate is defined as the number of departing aircraft that can take off from the runway(s) modelled per 15-minute interval. The service rate is assumed to follow a time-dependent Erlang distribution with parameters k and $k\mu$. Using a regression tree, an empirical distribution can be determined by looking at the route availability and arrival throughput for every 15-minute interval. This distribution can then be used to estimate the parameters k and $k\mu$ for the Erlang distribution. The regression tree enables interactions with the arrival flow to be modelled in a simplified manner. The take-off time is then calculated by adding the expected travel time and the expected queue time to the actual push-back time.

Using data from the most frequently used runway configuration at Newark Liberty International (EWR) airport in 2011, the model results were obtained. Estimations of the unimpeded taxi-out times and throughput distributions were also determined using the 2011 data, with the input to the model being the pushback schedule in 2011. To validate the model, the results were compared to a deterministic model created using the same data. Compared to the actual times, the deterministic

model underestimates the mean taxi-out time in both in periods of high and low congestion. While the stochastic model created by Simaiakis [5] also underestimates the mean, it performs much better overall, especially in periods of low congestion. In the deterministic model, average taxi-out times are underestimated by over a minute, whereas the stochastic model only does so by 30s.

Additionally, to gauge the model's predictive ability, the model was used to estimate taxi-out times in the years 2007 and 2010 while using model parameters from 2011. Regarding the average taxi-out time as a function of the number of aircraft taxiing out, the predictions in 2010 are as accurate as in 2011. However, the analysis of the averages of a single day shows that the taxi-out time is more difficult to predict on days with a continuously high demand. While the estimate of the throughput is quite accurate, every error also propagates to every taxi-out time as the queue never appears to be empty. Once again, in 2010, the model produces good results when compared to the actual data, especially on days with low demand. For 2007, however, the results show greater deviation from the actual data. This is due to the information on route availability not being available, leading to less accurate distributions.

Given the time stamps between pushback and take-off are unknown, the model provides an effective strategy to discretise the departure process. However, the model does not include uncertainty in the push-back schedule, meaning the predictive properties of the model are only realistic for roughly 15-minute intervals. In order to accurately capture the departure behaviour of aircraft up to two days in advance, the uncertainty in push-back must be included.

An earlier version of this analytical model was also developed by Simaiakis [16]. This version also separates the TXOT into three parts: the unimpeded taxi time, the taxiway system interactions, and the departure queue. However, the models differ in the assumptions made. In [16], the UTXOT is estimated by plotting the taxi-out time against the take-off queue length. A linear regression is then applied and the Y-intercept is taken as the UTXOT. The regression only takes into account data points consisting of periods when less than 8 aircraft are present in the queue. This method differs from Simaiakis' more recent papers, where the UTXOT is determined from distributions obtained in low-traffic scenarios. However, the main difference between the papers occurs in the determination of the departure demand rate. In the earlier version, the inter-arrival times at the runway are assumed to be random, and the departure demand rate is therefore modelled as a non-stationary Poisson process. The latest version, however, assumes a deterministic flow for the departure demand rate. While both papers model the runway queue and services rate with an Erland distribution, the new version is much more complex due to the regression tree created using arrival data.

Given the push-back time as an input, all the analytical models provide effective ways to model the departure process. A more detailed estimation of the TXOT is provided by Simiakis et al. [5][16], in which the travel time is separated into unimpeded taxi-out time and additional travel time due to taxiway system interactions. Hebert et al. [15] focus mainly on the determination of the service rate of the departure queue. While these models focus on different aspects of the departure process, all show promise for future development.

2.2.3. Numerical models

In addition to analytical models, several numerical models exist through which the departure process can be simulated. In this section these models will be explored. The first model considered is a PhD thesis created by Shumsky [6] in 1995 which aims to predict take-off times. As opposed to predicting taxi-out time only, where the pushback (or gate departure) delay is not included, since the aim here is to predict take-off time, the pushback delay is included. The departure process as modelled by Shumsky can be seen in Figure 2.8. The gate departure delay and the taxi-out times are considered independent variables for which two separate models are created.

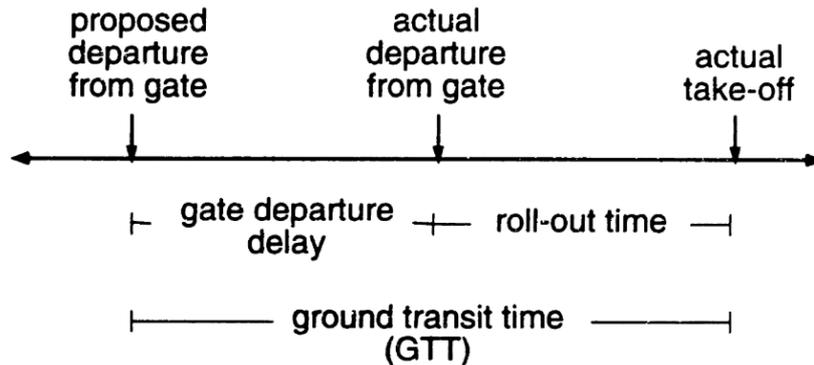


Figure 2.8: Definition of ground transit time (GTT) [6]

The first section aims to predict gate departure delay, in which two separate methods for estimating this delay are described. The first method involves analysing the influence of the weather and runway configuration on pushback delay. It concludes that, while the factors are statistically relevant, the largest cause of pushback delay is not explained by either of these factors. The second method is based on delay propagation in aircraft schedules. A prediction for the pushback delay is made by analysing the arrival time of the previous flight and the minimum turnaround time. This information is especially relevant for flights with delays of over 30 minutes as these delays are not random and should therefore not be included in the pushback distribution. By doing so, however, the time horizon of the simulation is limited to one to three hours before departure since this is the earliest the information can be obtained. Since the output of the gate departure delay model is deterministic it does not include stochastic uncertainty.

The second section aims to predict the taxi-, or roll-out time. This is done by simulating the aircraft flow from the gate to the departure runway and is based on an aircraft flow model. The number of pushbacks in a certain time period determines the rate of flow onto the taxiway system. Using historical data the travel time to the queue could be determined and was assumed constant. The service rate of the runway is then modelled using a cumulative exponential capacity estimate and is limited by the airport capacity. The model was created based on Boston Logan International Airport, where all the active runways are modelled as a single server with a capacity equal to that of all runways combined.

In order to verify the model, several empirical tests were performed comparing the forecasted number of pushbacks with the actual number of pushbacks in a 10 minute period, with the forecast being produced 30 min in advance. An analysis of the results show that, given perfectly accurate

predictions of push-back times, the number of aircraft on the airfield in the next 10 minutes can be predicted with an RMSE of 1.4 aircraft. Given a forecast window of 1 hour, the RMSE increases to 2.2 aircraft.

This model proves to be extensive and produces accurate and reliable results. However, the stochastic nature of the departure process is not captured, and the interactions between departing and arriving aircraft are not included. While it does include the possibility to update the inputs using real-time information, these uncertainties cannot be ignored.

The next paper considered, written by Pujet et al. [7], models the departure process at Boston Logan International airport from terminal to take-off as an input-output system. The structure of this model can be seen in Figure 2.9. Since no data is available on push-back requests or push-back clearances, this model uses the actual push-back time as input. As such, only the last two blocks are included in the model.

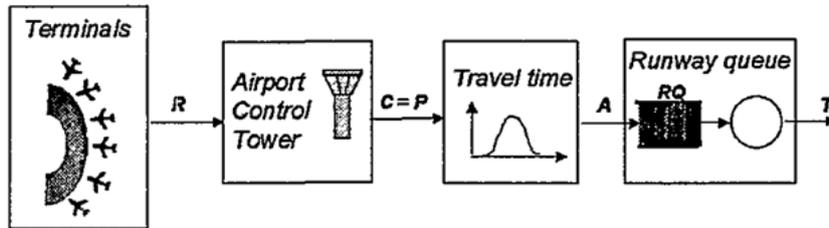


Figure 2.9: Structure of the departure process model [7]

The travel time is estimated using data from off-peak hours when very few aircraft are present in the taxiway system. A gaussian distribution is then fitted through the travel time data for every runway configuration and airline pair. It should be noted that the take-off roll and the initial climb are included in the travel time estimated by this distribution.

By adding the travel times to the actual pushback times of different aircraft, the number arrivals in the departure queue in a certain time period can be determined. The queue is simulated by using a balance equation of aircraft arrivals in the queue and aircraft leaving the queue. As described in subsection 2.2.2, a method like that used in [15] is used to determine the take-off rate. In periods where many aircraft are occupying the taxiway system, it is assumed that the queue is never empty. The inter-departure times in these periods must be analysed. To simulate the runway availability the server absence concept previously described is applied, meaning that the runway is available in each time period with a probability p . The probability and capacity are chosen such that the probability distribution matches the actual data.

A simulation was performed and the model outputs were compared with actual data. The taxi-out times predicted were similar to those observed, thus verifying the model. From the results it can be observed that mean taxi-out time and its variance increases with the amount of traffic. The model provides a good fit when light to medium traffic is observed. In heavy traffic, however, the model fit is less accurate. This can be explained by the fact that periods with heavy traffic occur less frequently than those with light or medium traffic.

The model is validated using departure demand data from 1997. Using this data, the model

provides reasonably accurate estimates of taxi-out times in most runway configurations. However, in some runway configurations the model overestimates the runway capacity, resulting in the average taxi-out time being underestimated. The reason for this is that these configurations are most likely used in bad weather scenarios where the runway capacity is lower in general.

This paper demonstrates that a simple model can provide useful insights into the departure process. To become more useful, however, the model must become more detailed. Details such as gate departure delay should be included, and, given more initial data, actual take-off time rather than the time at the end of the initial climb.

Andersson et al. [17] built upon the model created by Pujet et al. [7] by modelling all ground operations at an airport. This includes the arrival, turnaround, and departure processes. The three processes are linked but are modelled separately. Once again, the departure process is separated into travel time and queue time. However, the model differs from [7] through the assumptions made.

The first difference is regarding the unimpeded taxi-out time. In [7], unimpeded taxi-time is determined by observing the number of aircraft in the taxiway system at time t . In [17], the unimpeded taxi-time is determined by observing the number of take-offs while an aircraft is taxiing out. In this way, data from aircraft experiencing long taxi-out times due to factors other than surrounding aircraft are not included in the distributions as the number of aircraft taking off is still high. The unimpeded taxi-out distributions are approximated by fitting the empirical results with Gaussian or log normal distributions.

Another difference occurs in the way that the runway queue service rate is modelled. In this case, it is modelled using Poisson distributions for each level of departure congestion. As the congestion level increases, the rate of the Poisson distribution increases until a maximum throughput is reached. Different distributions are also obtained for each runway configuration, periods of good weather, and periods of bad weather. In [7], the runway service rate distribution included a server absence which was a clear indicator of runway availability. In this model [17], a server absence is not included. However, the Poisson distributions still provide information about runway availability since they are fitted through actual data.

Data from Hartsfield-Jackson Atlanta International Airport in 1998 and Dallas/Fort Worth International Airport in 1997 was used to calibrate the departure model. Validation with data from different times was not performed. However, both sets of calibration results matched the experimental data accurately which provides some indication of the validity of the model.

The final numerical model considered is an earlier model developed by Simaiakis et. Al [18]. Besides the determination of the queue service rate, the model is essentially the same as their later model described in subsection 2.2.2 [5]. While the model in [5] assumes an Erlang distribution, the service time in [18] is assumed to be a random variable with three possible outcomes. An analysis of the inter-departure times at Boston Logan International Airport in 2007 reveals that in most cases there is either a one or two minute take-off interval between aircraft. These are therefore the first two outcomes which the random variable can assume. The third possible outcome is the next minute increment which satisfies the condition that the sum of the probabilities of the random variables must

equal 1. In this case, this outcome is the 5-minute service time.

The paper also includes an in-depth explanation of the modelling of the ramp and taxiway interactions, which is represented by $\tau_{taxiway}$ in the model. This term is difficult to estimate since there are no specific operating conditions in which this is the dominant term. This term is therefore ignored at first, meaning only the unimpeded taxi-out time and queue time are considered. Analysing the results of the model excluding this term shows that compared to reality, the model overestimates the take-off rate in periods of medium traffic. This was expected as the model overestimates the rate at which aircraft arrive at the runway since no ramp delay is included. The linear term $\alpha R(t)$, equivalent to $\tau_{taxiway}$, is therefore added to counteract this phenomenon, where $R(t)$ is the number of aircraft currently in the taxiway system, and α is a parameter that depends on the airport and runway configuration. The value of α is chosen so as to yield the optimal fit between the actual and the modelled distributions, leading to a more accurate prediction of taxi-out time.

The model is then validated using data from 2008 at Boston Logan International Airport, using model parameters established using data from 2007. The model accurately predicts the taxi-out time in two out of three configurations in both 2007 and 2008. While no explanation is provided for the taxi-out times being overestimated in this configuration, it is likely due to this being a lesser used configuration at the airport.

All the numerical models provide legitimate ways to discretise the airport departure process. In [6], Shumsky creates a predictive TXOT model which includes gate departure delay. The model also provides a real-time update capability, and should therefore be used for short term analysis. However, the model fails to address the stochastic uncertainty. The paper by Pujet et al. [7] and Andersson et al. [17] both create models in which the departure process is split into a travel time and a queue time, where the travel time and runway queue service rates are modelled as stochastic variables. Lastly, Simaiakis et al. [18] create a TXOT prediction model by assuming a stochastic runway service rate and a deterministic flow entering the runway queue.

2.3. Research questions

In this section, the research objective, research question, and corresponding sub-questions are presented. These have been formulated based on the review of existing literature which has identified areas for possible further work.

2.3.1. Research objective

The objective of the research that is to be performed in the follow up thesis is as follows:

“To gain a better understanding of the underlying nature of the departure process by creating a machine learning model capable of providing reliable predictions for normal and extended taxi-out times.”

2.3.2. Main research question

To achieve the research objective, the following main research question should be answered:

“Which taxi-out features most significantly influence the accurate prediction of normal and extended taxi-out times?”

2.3.3. Sub-questions

In answering the main research question, several sub-questions must also be answered. These sub-questions relate to the steps required to create a model and are as follows:

- Which machine learning techniques are feasible for predicting taxi-out-time?
- Which metrics can be used to compare the performance of different techniques?
- Of the feasible machine learning techniques, and based on the performance metrics, which technique performs the best?
- Are the results of the model sufficiently accurate?
- Which features can be used to model the taxi-out process?
- Which features have not yet been tested?
- Which features have the most influence on normal TXOT?
- Which features have the most influence on extended TXOT?

3

Methodology

In this chapter, the methodology used to produce reliable predictability models for both normal and extended TXOTs is presented. The study is performed at Austria's largest airport, Vienna International Airport. As shown in [Figure 3.1](#), the airport consists of two runways, namely, runway 11/29 and runway 16/34. The runway configuration most frequently used is with segregated operations (departures only) on runway 29, and mixed-mode operations (both arrivals and departures) on runway 34. In order to create a comprehensive model, the study is performed using all possible runway configurations.

The main reason for selecting Vienna Airport for the study was due to the availability of high quality detail data. Given that this data is key to creating and validating a functioning model this chapter presents an in-depth overview of the available data in [section 3.1](#), including relevant statistics and an explanation of the data. In [section 3.2](#), all the steps are described which are required to prepare the final dataset to be used in the model, which is summarised in [section 3.3](#). This is followed by a section on feature selection, identifying significant variables with impact on TXOT and a more general exploration of how congestion levels have a major impact on TXOT in [section 3.5](#). In [section 3.6](#) and [section 3.7](#) two possible machine learning models are explored and a technical comparison is presented in [section 3.8](#) which leads to the selected model for further investigation. In [section 3.9](#), an analysis is presented on how the ML model could be applied to extended taxi-out times. Finally, in [section 3.10](#), an analysis is performed in order to determine the causes of certain extended TXOTs not captured by the model.

3.1. Data Acquisition

Access to quality data is critical to enable accurate predictive modelling and validation of airport operations. Analysis of recent work identified the availability of a high quality set of relevant data covering an extended period of operations at Vienna International Airport and it was decided to use this as the basis for the TXOT model. Key statistics relating to the operation are presented in [subsection 3.1.1](#) and a more detailed overview of the content of the dataset is to be found in [subsection 3.1.2](#).

¹<https://rzjets.net/airports/?code=LOWW>

3.1. DATA ACQUISITION

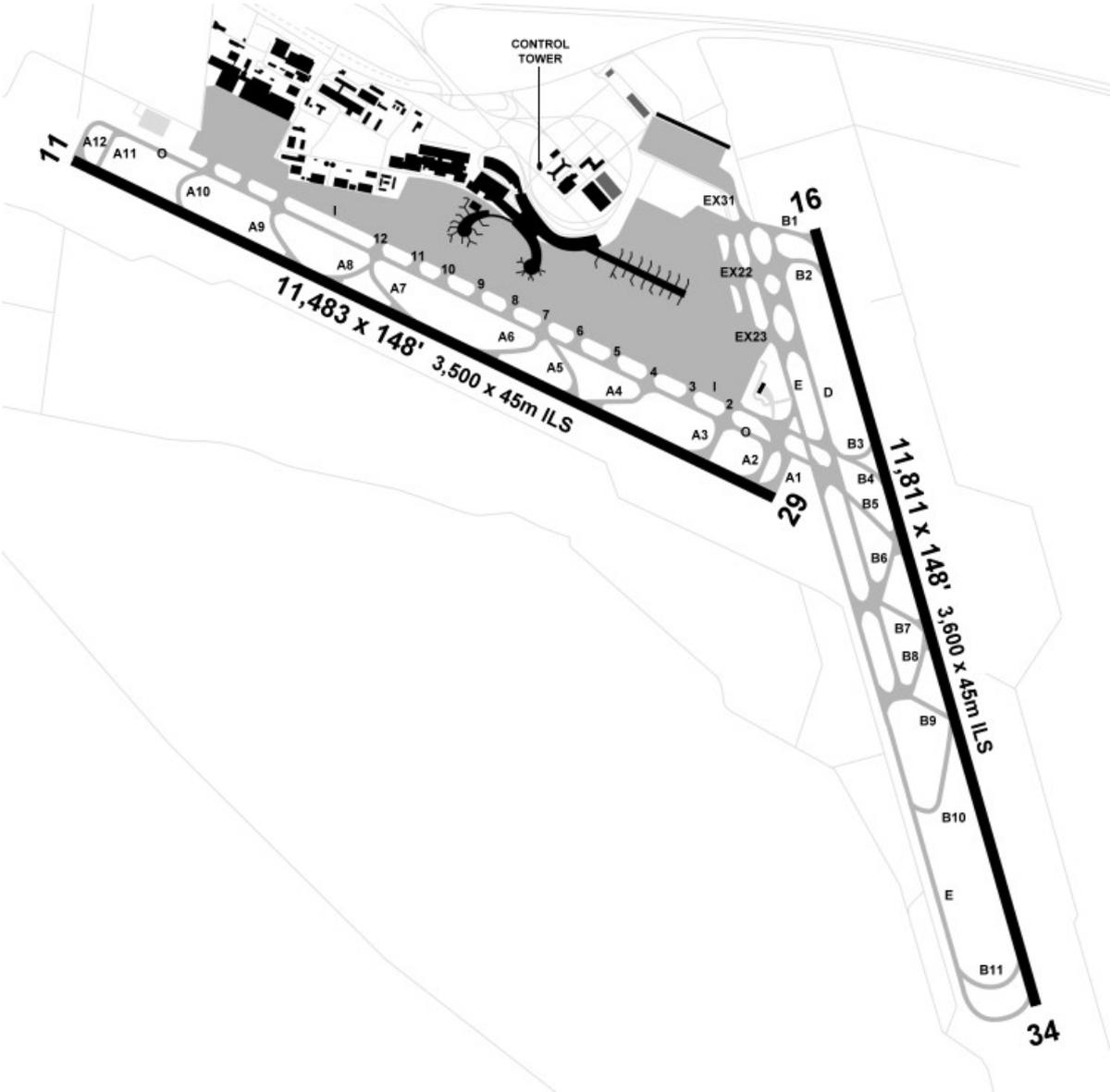


Figure 3.1: Layout of Vienna International Airport¹

Runway	Departures		Arrivals	
	Number	Percentage	Number	Percentage
11	2938	2.5%	15034	12.5%
29	75554	64.5%	16221	13.5%
16	22814	19.5%	33446	28.0%
34	15911	13.5%	55164	46.0%
Total	117217	100%	119865	100%

Table 3.1: Number of flights per runway

3.1.1. Key Statistics

The data used in the case study was provided by Austrocontrol, the ANSP at Vienna Airport. It consists of one year of airport operations data from January 1st 2015 until December 31st 2015 compiled from multiple data sources. The data consists of all arrival and departure flights from all runways, a total of around 240,000 flights. A breakdown of these flights can be found in [Table 3.1](#).

From the summary it can be seen that the total number of incoming and outgoing flights is almost equal. This provides an initial measure of the reliability of the data, as inbound and outbound flow should be balanced. The difference is negligible and can be attributed to errors in processing the data.

In relation to runway configuration, it is clear that the most utilised runway for departures is runway 29, while for arrivals it is runway 34. Also note-worthy is the significant number of departures from runway 34 which is most frequently used for mixed-mode operations. For convenience, this most prevalent runway configuration will be referred to as D29M34.

Another observation to make is that the least commonly used runway for both departures and arrivals is runway 11. The reason behind the low departure count is that the departing aircraft would have to cross runway 16/34, thus impeding any potential arrivals/departures on that runway. The same applies to arriving aircraft at runway 11, where in case an aircraft has an unsuccessful landing and requires a go-around, this aircraft would have to cross runway 16/34. Lastly, the low arrival count on runway 29 is also caused by arrivals having to cross runway 16/34. Due to this runway crossing limitation, ATCOs will usually only use these runway configurations during off-peak hours or in adverse weather conditions.

3.1.2. Data Overview

The data received for Vienna Airport was collected from multiple sources. In [Table 3.2](#) an overview of the data sources and the corresponding data provided by each is shown. The first set of data is obtained from the Advanced Surface Movement Guidance and Control System (A-SMGCS). This provides important information such as the Actual Off-Block Time (AOBT), Actual Take-Off Time (ATOT), and Actual Time of Arrival (ATA), as well as the departure gate and departure runway for outbound flights, and the arrival runway for inbound flights. This is an important source as the availability of the AOBT and ATOT allows for the actual TXOT to be calculated which is needed for comparisons with the predicted TXOTs.

The next data set is obtained from the wind speed meter (WMA), which provides wind speed and

3.1. DATA ACQUISITION

Data Type	Parameter	Description
A-SMGCS	Flight Date	Date of Flight
	Event Time	Time of flight event occurrence
	Event	Event type e.g. AOBT, ATOT
	Flight No.	ICAO flight number
	Flight No.	IATA flight number
	Origin	ICAO flight origin
	Destination	ICAO flight destination
	AC Type	Aircraft type
WMA	Gate/Runway	Arrival gate/Departure runway
	Date	Date of WMA reading
	Time	Time of WMA reading
	Wind Direction	Wind direction at time of WMA reading
METAR	Wind Speed	Wind speed at time of WMA reading
	Date	Date of METAR observation
	Time	Time of METAR observation
	Cloud Ceiling	Cloud ceiling altitude (ft)
	Day/Time (dHHMM)	Day/Time of METAR observation
	Airport Code	ICAO airport code
	Obs. Time	Time of observation (mmHHMM)
	Wind Vector	Wind direction/speed (XXXdeg YYkts)
	Visibility (m)	Visibility in meters
	Clouds	coverage/altitude (ft)
	Temp (deg) / Dew Point (deg)	Temperature / Dew Point
	SNOWTAM	SNOWTAM in format RXX/TEDDBB
Radar Track	Trend	Trend for future weather
	Flight No.	ICAO Flight No.
	AC Type	Aircraft type/model
	Weight Class	ICAO Weight Class
	Time	Time of recording (HH:MM:SS)
	Latitude	Latitude at time of recording
	Longitude	Longitude at time of recording
	Flight Level	Flight Level per 100ft
Ground Speed (kts)	Ground Speed in kts	
Visibility	AC On Runway	Boolean; 1 if aircraft is on the runway, 0 if not
	Date	Date (yyyy-mm-dd)
	Time	Time (HH:mm:ss)
	Visibility	Visibility in km
	Cloud Type 1 (okt alt type)	oktas/altitude(ft)/cloud type
	Cloud Type 2	oktas/altitude(ft)/cloud type

Table 3.2: Available data for Vienna International Airport

3.2. DATA PREPARATION

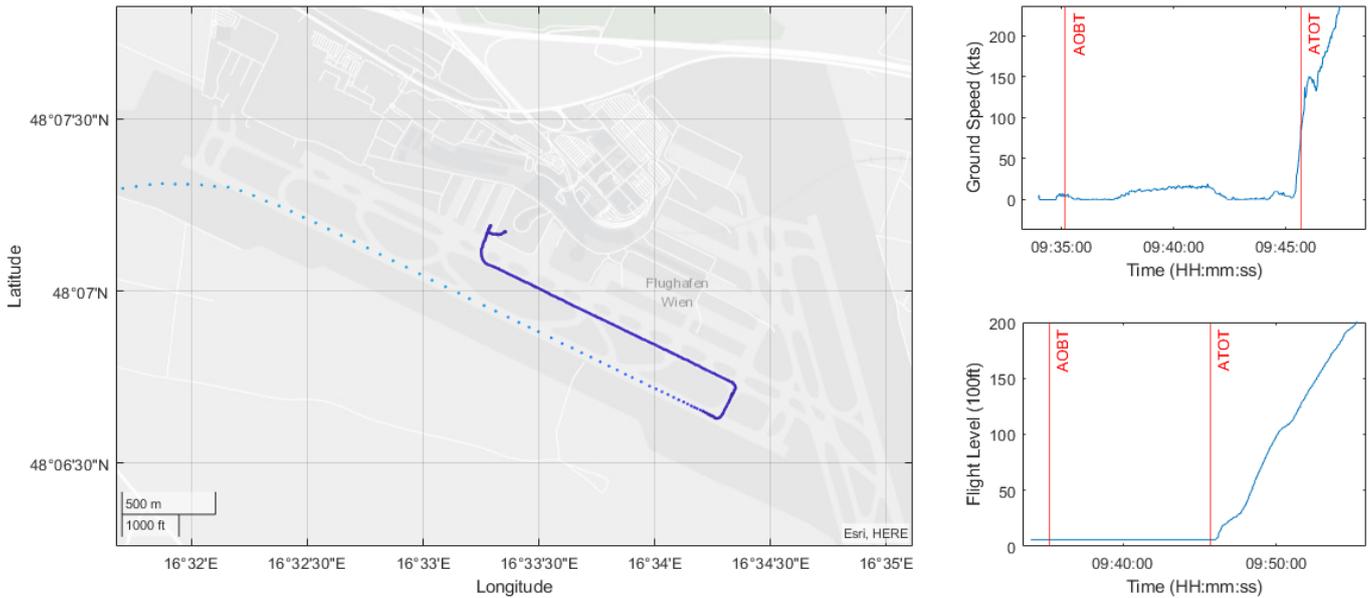


Figure 3.2: Departure ground track, ground speed, and flight level of an example flight departing from runway 29 - Vienna Airport

wind direction observations as well as the date and time of the observations. This data set is useful to examine possible influence on TXOT as a result of wind speed/direction combinations. The third main data source is the Meteorological Terminal Air Report (METAR) which provides a multitude of information including temperature, dew point, visibility and cloud ceiling. Several of these were used as variables in the model to determine their influence, if any, on TXOT.

The fourth set of data is the RADAR track data. The data provides a complete flight profile per flight which includes measurements of the latitude, longitude, groundspeed, and flight level of the aircraft every second. For departure flights, the flight profile is measured from the gate location until 30NM from the airport. The data also includes the weight class of the aircraft, which is used as a feature for TXOT prediction. As seen in [Figure 3.2](#), this data provides useful insight in visualising departures. Common routes taken from specific gates to runways can be observed and any irregularities in TXOTs can be investigated. The data was also used to verify the AOBT and ATOT timestamps provided by the A-SMGCS. While all the other data sources provided a full year of data, only 6 months of Radar Track data (25/06/15 - 31/12/15) for departures was available. Therefore, not all timestamps could be verified and not all the outliers could be fully analysed. For this same reason, other than the aircraft weight class, no other features were extracted from the Radar track data to be used in the ML models.

3.2. Data Preparation

In this section, the required preparation of the data is explained. This phase covers all activities to set up the final dataset from the raw data provided, as well as the merging and the cleaning of the data. All processing steps were performed in MATLAB. A flow chart of the key tasks is illustrated in [Figure 3.3](#).

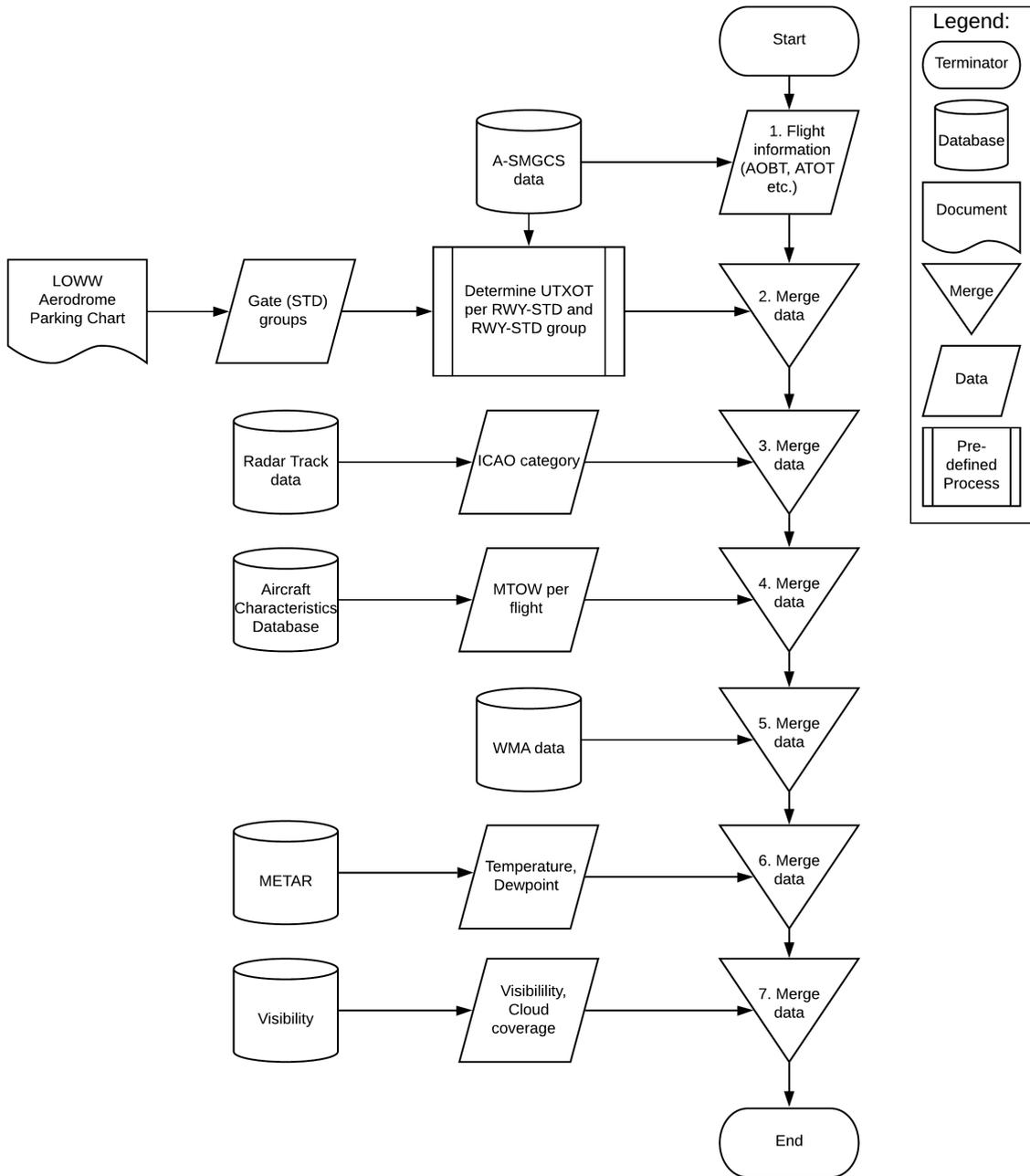


Figure 3.3: Flow chart of data preparation

The first step is to extract general flight information from the A-SMGCS data. This includes the date, flight number, aircraft type, departure gate, departure runway, AOBT and ATOT. The additional time data, such as the hour of the flight, day of the week, day of the year etc. are also calculated here.

In the second step, the unimpeded taxi-out time (UTXOT) is calculated per runway-stand combination and runway-stand group combination, and this is merged with the flight information data. As identified in the literature review, a standardised process exists for calculation of UTXOTs using information from the A-SMGCS as input. This process was adopted for the purposes of this study and is described in more detail in [subsection 3.2.1](#). The additional features considered to be of interest in evaluating TXOT impact which are also calculated as part of this step are as follows: congestion level, throughput, departure count, 1st estimation UTXOT per runway-stand combination, 1st estimation UTXOT per runway-stand group combination, and saturation level per runway-stand combination. Additionally, in order to determine the UTXOT per RWY-STD group, the gate groups first had to be determined. This information was extracted from the Vienna Airport Aerodrome Parking Chart².

In step 3, the ICAO wake turbulence category per flight is extracted from the Radar Track data and merged with the rest of the combined data so far. In step 4, the MTOW per aircraft was merged into the dataset using the Aircraft Characteristics Database³ obtained from the federal aviation administration (FAA) as source. Lastly, in steps 5-7, the useful data from the WMA, METAR, and Visibility datasets was extracted and combined with the rest of the data to create the final dataset. This final dataset consists of 35 data elements, as presented in [Table 3.3](#) and forms the basis for training and validation of the machine-learning prediction model.

3.2.1. UTXOT Determination

Step 2 of the data preparation flow chart shown in [Figure 3.3](#) entails the calculation of the UTXOT per RWY-STD and RWY-STD-group and is based on the method outlined in the Additional Taxi-Out Time Performance Indicator document [3]. In order to determine whether a taxi-out can be considered unimpeded several other factors have to be taken into consideration like the airport throughput, congestion level and saturation level. A flowchart representing this process was shown in [Figure 2.5](#) whose steps are outlined as follows:

1. Perform a dataset quality check, clean data:
 - Check consistency of AOBTs and ATOTs from A-SMGCS with Radar Track data, as described in [subsection 3.1.2](#).
 - Reject flights with a negative TXOT; flights with a negative TXOT are impossible and are therefore removed from the data set. Negative TXOTs can occur due to faults in the A-SMGCS where false timestamps are recorded.
 - Reject flights with an actual TXOT of more than 90 minutes.
 - Reject helicopter flights. Helicopters do not follow the same departure procedure as airplanes and are thus removed from the dataset.
 - Remove incomplete records.

²https://eaip.austrocontrol.at/lo/200716/ad_2_loww.htm - Accessed 28/06/2020

³https://www.faa.gov/airports/engineering/aircraft_char_database/ - Accessed 28/06/2020

2. Perform the following for each departure flight:

- Calculate the actual TXOT using the following equation:

$$ATXOT = ATOT - AOBT \quad (3.1)$$

- Determine the congestion level by counting the number of take-offs and landings in the time interval between the AOBT and ATOT of the respective flight.
- Determine the airport throughput observed in the hour preceding the AOBT of the flight. The airport throughput is defined as the total number take-offs and landings at the airport.
- Determine the runway-stand (RWY-STD) combination of the flight.

3. Determine the saturation level for each departure flight:

- Estimate the peak airport throughput (R) using the 90th percentile of the airport throughput in the preceding hour of all flights.
- Next, calculate a first estimation of the UTXOT per RWY-STD combination (U_1) by using the 20th percentile of the actual TXOTs of the flights belonging to that RWY-STD combination.
- Repeat the previous step to calculate the first estimation of the UTXOT per runway-stand-group combination (U_{group}).
- Compute the saturation level (L) per RWY-STD and RWY-STD-group using the following equations:

$$L = U_1 * \frac{R}{60} \quad (3.2)$$

$$L_{group} = U_{group} * \frac{R}{60} \quad (3.3)$$

4. Identify unimpeded flights:

- Flights are considered unimpeded if they meet the following condition:

$$congestionlevel \leq 0.5 * saturationlevel \quad (3.4)$$

5. Computation of UTXOT:

- If the number of unimpeded flights belonging to a specific RWY-STD combination is greater than or equal to 10, the UTXOT for that couple is defined as the median TXOT of the unimpeded flights.
- If the number of unimpeded flights belonging to a specific RWY-STD combination is less than 10, no UTXOT is computed for that couple. Instead, the median TXOT of all unimpeded flights belonging to the RWY-STD-group is used as the UTXOT.

Once these steps have been completed, the process in [Figure 3.3](#) can be continued.

3.3. Prediction Variables

The purpose of the data preparation outlined above is to produce a streamlined set of variables which provide sufficient granularity to model the actual data and help to understand where specific features

may have a significant influence on the target outcome, i.e. TXOT. It is also important that data which has limited or no additional value is eliminated to improve processing times. In this section, the required features which have been extracted from the raw data are presented alongside the additional variables which have been subsequently determined, as described in [subsection 3.2.1](#), that were not provided in the raw operational data. Note that the terms *features* and *prediction variables* refer to the same concept and will be used interchangeably throughout. The overview of the final variables to be used in the model are presented in [Table 3.3](#).

3.4. Feature Selection

As highlighted in [chapter 2](#), another step that should be performed before a machine learning model is trained is feature selection. This task is performed to identify which features best describe the variance in a dataset, or specifically, which features have the most influence on TXOT. Feature selection techniques have 3 main objectives, namely:

1. To improve the prediction performance of the predictors.
2. To provide faster computational performance and more effective predictors.
3. To provide a better understanding of the underlying process that generated the data.

The chosen feature selection method is the RRelief technique which was discussed in [subsection 2.1.1](#). The technique is a proven and reliable one as demonstrated by Herrema et. al [1], [19] whose algorithm is described below.

RRelief penalises predictors that give different values to neighbors with the same response values, and rewards predictors that give different values to neighbors with different response values. RRelief also uses intermediate weights to compute the final predictor weights.

Given two nearest neighbors, the algorithm assumes the following:

- W_{dy} is the weight of having different values for the response y .
- W_{dj} is the weight of having different values for the predictor F_j .
- $W_{dy^d_j}$ is the weight of having different response values and different values for the predictor F_j .

RRelief first sets the weights W_{dy} , W_{dj} , $W_{dy^d_j}$, and W_j equal to 0. Then, the algorithm iteratively selects a random observation x_r , finds the k -nearest observations to x_r , and updates, for each nearest neighbor x_q , all the intermediate weights as follows:

$$\begin{aligned} W_{dy}^i &= W_{dy}^{i-1} + \Delta_y(x_r, x_q) \cdot d_{rq} \\ W_{dj}^i &= W_{dj}^{i-1} + \Delta_j(x_r, x_q) \cdot d_{rq} \\ W_{dy^d_j}^i &= W_{dy^d_j}^{i-1} + \Delta_y(x_r, x_q) \cdot \Delta_j(x_r, x_q) \cdot d_{rq} \end{aligned} \tag{3.5}$$

where the variables are denoted as follows:

- i is the iteration step number and m is the number of iterations specified by ‘updates’. By default, RRelief uses all observations and thus sets m to the total number of observations (flights).

3.4. FEATURE SELECTION

	Variable	Description
1.	Date	Date of Flight
2.	Fltno	ICAO flight number
3.	ACtype	ICAO aircraft type
4.	ICAOcat	ICAO wake turbulence category (H/M/L)
5.	ACweight	MTOW of aircraft (kg)
6.	Origin	ICAO airport code of flight origin
7.	Destination	ICAO airport code of flight destination
8.	DepSTD	Departure gate (/stand)
9.	STDgroup	Gate (stand) group
10.	DepRWY	Departure runway
11.	AOBT	Actual Off-Block Time (hh:mm:ss)
12.	ATOT	Actual Take-Off Time (hh:mm:ss)
13.	ConLVL	Congestion level; number of movements (take-offs and landings) in the time interval between the aircraft's AOBT and ATOT
14.	Throughput	Airport throughput;
15.	Depcount	Departure count; number of departure in the 20 mins preceding the aircraft's AOBT
16.	UTXest	1st estimation of the unimpeded taxi-out time of the runway-stand combination corresponding to the flight (mins)
17.	GrpUTXest	1st estimation of the unimpeded taxi-out time of the runway-stand-group combination corresponding to the flight (mins)
18.	SatLVL	Saturation level; the greater the congestion level compared to the saturation level, the greater the likelihood of a flight being delayed due to traffic ahead
19.	UTXind	1 if flight is unimpeded, 0 if not
20.	UTXOT	Unimpeded Taxi-Out Time (mins)
21.	DirectionWMA	Wind direction (deg)
22.	SpeedWMA	Wind speed (kts)
23.	Visibility	Visibility in kilometers (km)
24.	Ceiling	Cloud ceiling altitude (ft)
25.	Temp	Temperature (deg)
26.	Dewpt	Dew point (deg); temperature to which the air must be cooled to become saturated with water vapor.
27.	Hour	Hour of the flight
28.	Dayweek	Day of the week
29.	Daymonth	Day of the month
30.	Dayyear	Day of the year
31.	Weekmonth	Week of the month
32.	Weekyear	Week of the year
33.	Month	Month of the year
34.	Peak	1 if the aircraft is departing in peak hours, 0 if not
35.	TXOT	Taxi-Out Time (mins); this is the target variable

Table 3.3: Prediction and target variables

- $\Delta_y(x_r, x_q)$ is the difference in the value of the continuous response y between observations x_r and x_q and is calculated using [Equation 3.6](#).

$$\Delta_y(x_r, x_q) = \frac{|y_r - y_q|}{\max(y) - \min(y)} \quad (3.6)$$

where y_r is the value of the response for observation x_r , and y_q is the value of the response for observation x_q .

- $\Delta_j(x_r, x_q)$ is the difference in the value of the predictor F_j between observations x_r and x_q . For continuous F_j , $\Delta_j(x_r, x_q)$ is calculated using [Equation 3.7](#).

$$\Delta_j(x_r, x_q) = \frac{|x_{rj} - x_{qj}|}{\max(F_j) - \min(F_j)} \quad (3.7)$$

where x_{rj} denotes the value of the j th predictor for observation x_r , and x_{qj} denotes the value of the j th predictor for observation x_q .

- d_{rq} is a distance function, calculated as shown in [Equation 3.8](#).

$$d_{rq} = \frac{\tilde{d}_{rq}}{\sum_{l=1}^k \tilde{d}_{rl}} \quad (3.8)$$

where the distance is subject to a scaling equation, [Equation 3.9](#)

$$\tilde{d}_{rq} = e^{-(\text{rank}(r,q) / \text{sigma})^2} \quad (3.9)$$

and where $\text{rank}(r, q)$ is the position of the q th observation among the nearest neighbors of the r th observation, sorted by distance. k is the number of nearest neighbors, specified by k . The scaling can be changed by specifying ‘sigma’, whose default is 50 for regression.

Finally, RReliefF calculates the predictor weights W_j after fully updating all the intermediate weights using [Equation 3.10](#).

$$W_j = \frac{W_{dy \wedge dj}}{W_{dy}} - \frac{W_{dj} - W_{dy \wedge dj}}{m - W_{dy}} \quad (3.10)$$

3.4.1. RreliefF parameter determination

The RReliefF algorithm requires 3 inputs as follows:

- X - predictor data: a numeric matrix where each columns corresponds to one of the prediction variables 1-34 from [Table 3.3](#), and each row corresponds to an observation (flight).
- y - response data: a numeric vector representing the TXOT per observation.
- k - number of nearest neighbours: specified as a positive integer scalar.

While the predictor data and response data had already been prepared, the value of the number of nearest neighbours to be used was not yet known. An analysis was therefore performed to determine its value.

The value of k was determined iteratively, where the value of k was set to 10 for the first trial using all observations. However, due to the computational time required for 1 iteration of the RreliefF

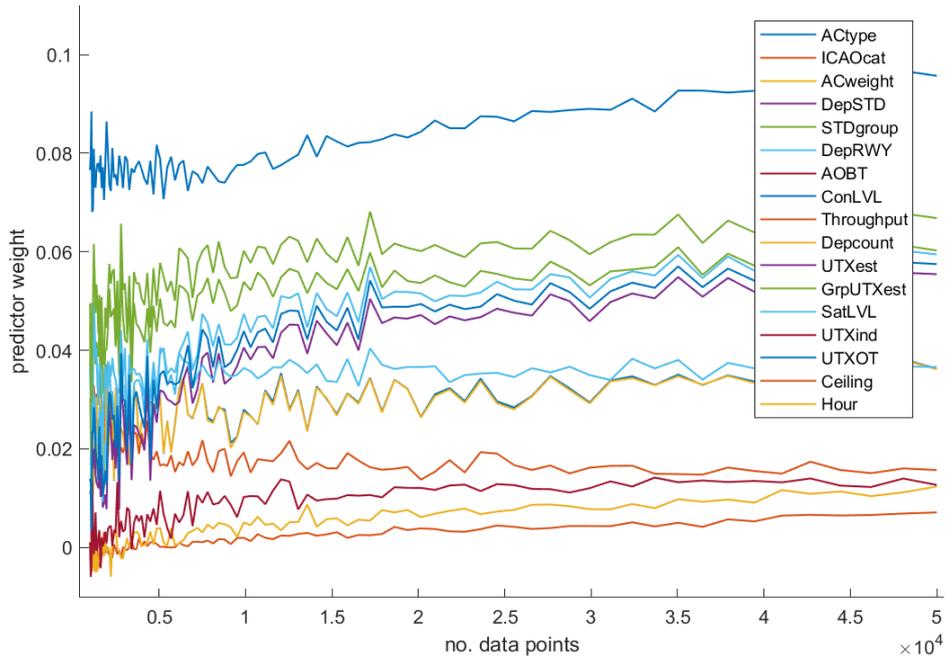


Figure 3.4: Feature importance vs. no. observations

algorithm using all observations, running the algorithm for a large range of k -values would become too time consuming. A separate analysis was therefore performed to determine the number of observations required for the algorithm to produce a stable output. The results of this analysis, where the number of observations was varied from 1,000 - 50,000, can be seen in Figure 3.4. Here, a conservative k -value estimate of 150 was used.

From the figure, it can be observed that the output of the RReliefF algorithm becomes sufficiently stable at roughly 25,000 observations. Using 25,000 data points significantly reduces the computational time of the algorithm, meaning the k -value analysis becomes more feasible. k -values ranging from 1-150 were tested using 25,000 observations per iteration. The resulting graph can be seen in Figure 3.5.

From the figure, it is clear that the results of the algorithm are stable by a k -value of 100. While any larger value for k would result in the same algorithm output, increasing k also increases computational time. The final k -value to be used in the algorithm was therefore set at 100.

3.4.2. Selected features

The RReliefF algorithm was applied using all 114,000 observations and a k -value of 100. The resulting feature importances can be seen in Figure 3.6.

The most important features, as determined by the RReliefF technique are as follows: ‘congestion level’, ‘UTXOT’, ‘saturation level’, ‘1st estimation UTXOT per RWY-STD’, ‘1st estimation UTXOT per RWY-STD group’, ‘departure runway’, ‘ICAO weight category’, ‘aircraft weight’, ‘stand group’, ‘aircraft type’, ‘departures in preceding 20 mins’, ‘unimpeded flight’, ‘throughput in preceding hour’, ‘hour of flight’, and ‘AOBT’.

These features are therefore selected for use in the ML models developed in sections 3.6 and 3.7.

3.4. FEATURE SELECTION

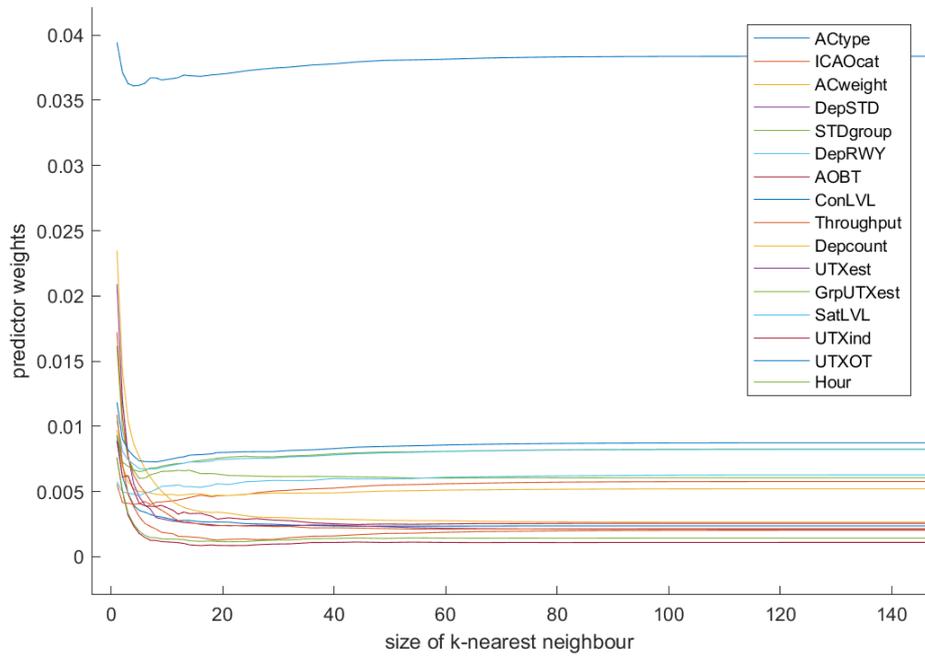


Figure 3.5: Feature importance vs. k-nearest neighbours

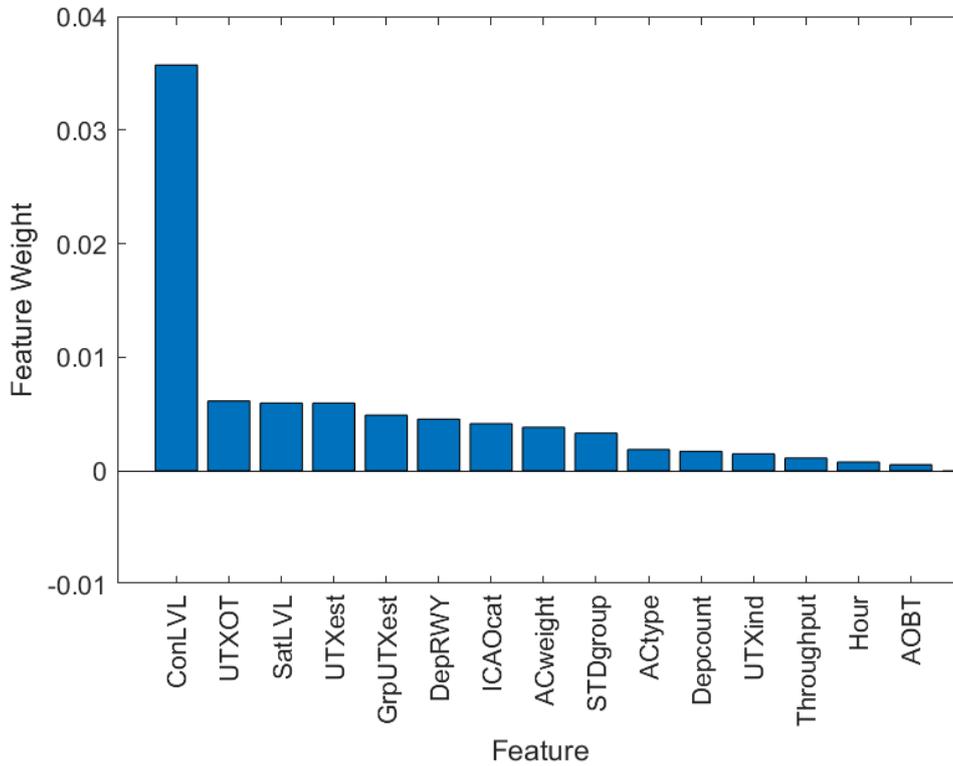


Figure 3.6: Normalized feature selection using RReliefF algorithm (see Table 3.3 for variable definitions)

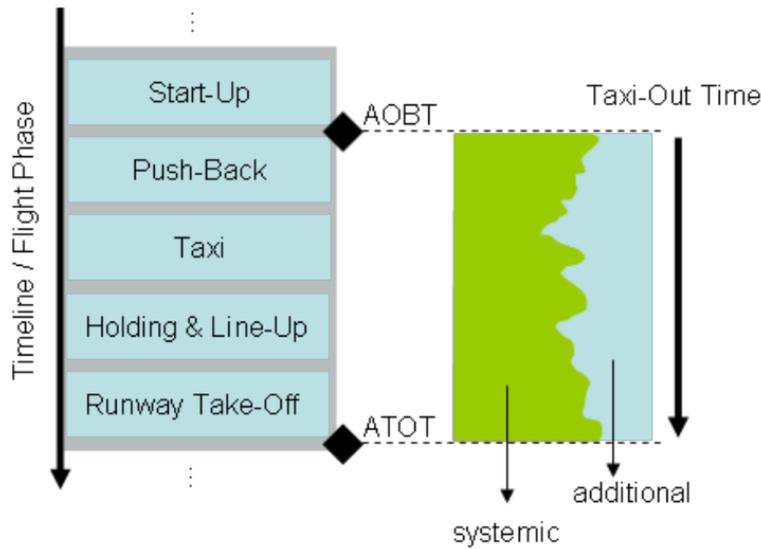


Figure 3.7: TXOT breakdown

By restricting the number of variables used when training ML models to only those which appear to have a significant influence on the target outcome the overall prediction process should be less complex and therefore faster with limited loss of accuracy.

3.5. TXOT Understanding

As suggested by Herrema et al., one of the purposes of calculating the TXOT is to "provide a reliable measure of the average outbound queueing time during times that the airport is congested" [1]. The TXOT is therefore an important indicator of airport operational efficiency and in this section an analysis is presented of the variation of TXOT at Vienna Airport for different levels of congestion.

The TXOT for a flight is defined as the elapsed time between the actual off-block time AOBT and the actual take-off time ATOT (as per Equation 3.1). The TXOT envelope includes both systemic durations and additional time aspects linked to the actual progress of the operations. Examples of systemic durations include time spans for certain procedures and queueing at the runway to ensure flight demand. A visual representation of this can be seen in Figure 3.7.

In order to understand the TXOT process, the TXOT vs. congestion level for all 466 RWY-STD combinations at Vienna airport were plotted. Here, the congestion level per aircraft is defined as the number of movements (take-offs and landings) that occur between the AOBT and ATOT of the aircraft. Since the model aims to make predictions for all runway configurations, all movements at the airport in this time envelope were considered. In figures 3.8 and 3.9, example plots of TXOT vs. congestion level for different runway-stand combinations can be seen.

It can be observed that in general for low levels of congestion, the TXOT remains fairly constant. This TXOT corresponds to the UTXOT and is represented by the horizontal red line in figures 3.8 and 3.9. The concept of UTXOT was discussed in detail in subsection 2.2.1. Additionally, once the congestion level reaches a certain level, the TXOT begins to increase linearly. This linear increase is represented by the purple line in figures 3.8 and 3.9. The point after which the TXOT starts to increase

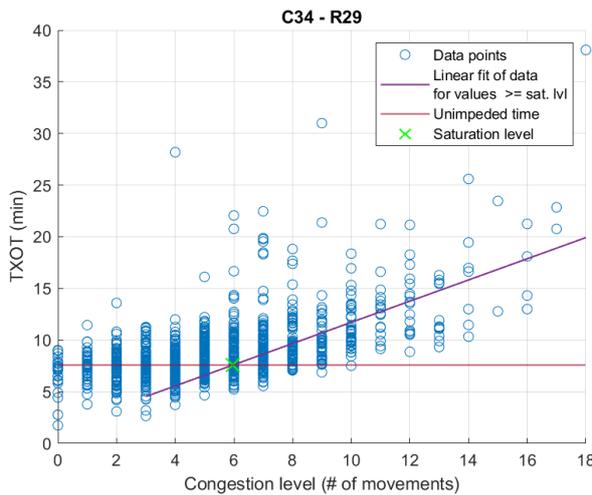


Figure 3.8: TXOT vs. congestion level for RWY-STD combination C34-R29

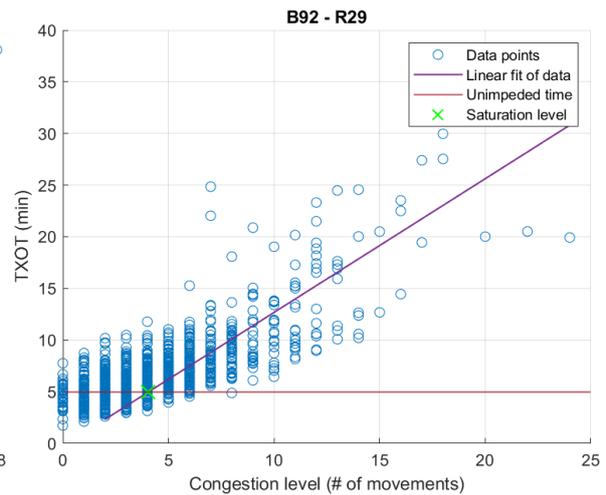


Figure 3.9: TXOT vs. congestion level for RWY-STD combination B92-R29

linearly is known as the saturation level, the calculation of which was described in [section 3.2](#). This value also corresponds to the intersection of the red and purple lines, highlighted by the green cross in figures [3.8](#) and [3.9](#).

While in general it is clear that the trend is linear, it can be seen in [Figure 3.9](#) that the linear fit of the data does not match the trend of the data. For this case and others, the skew is due to outliers. Outliers and delayed flights will be analysed in depth in [section 3.10](#).

3.5.1. Additional prediction variables

Congestion level is not the only variable of interest. Further insights can be gained by including other variables which appear to have a clear effect on TXOT behaviour. Based on the analysis performed as part of the data preparation, [subsection 3.4.2](#), and in line with the EUROCONTROL PRU methodology [\[3\]](#), additional variables which have a significant influence on the TXOT are identified as follows: ‘congestion level’, ‘unimpeded taxi-out time per RWY-STD and STD group’, ‘saturation level’, and ‘number of departures in last 20 min’. As described in [section 3.2](#), these variables were not included in the data received from Vienna Airport and therefore first had to be calculated. They are included with the other prediction variables in the final dataset and can be viewed as entries 13-20 in [Table 3.3](#).

For periods of low traffic, the TXOT is captured using statistical analysis and is referred to as UTXOT. The UTXOT indicator can be used to make more accurate predictions of actual TXOT. It is first calculated for a comparable grouping of flights characterized by the same combination of RWY-STD, followed by flights of the same RWY-STD group combination.

3.6. Neural-Network Levenburg-Marquardt (NNLM) feasibility

In [chapter 2](#), several machine learning models applicable to TXOT prediction were explored. As presented by Herrema et al. [\[1\]](#), 2 such feasible ML methods are the Regression Tree (RT) and Neural-Network Levenburg-Marquardt (NNLM) techniques. While other feasible methods were considered,

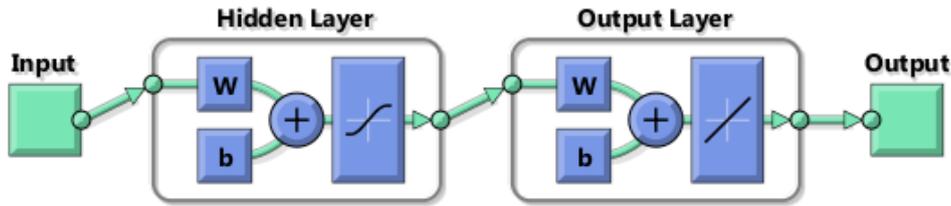


Figure 3.10: NNLM model overview [8]

for reasons made clear in [section 3.7](#), only these 2 techniques were explored in depth. In this section, the set up and results of the NNLM technique will be discussed. In [section 3.7](#), the feasibility of the regression tree will be explored. The models will then be assessed in relation to their performance indicators and other relevant factors. The most appropriate technique will then be chosen, and the model will be expanded upon.

3.6.1. NNLM overview

In neural network fitting problems, the aim is to create a map between a data set of numeric inputs and a set of numeric targets. In this case, the set of numeric inputs is the feature matrix 'X' (variables 1-34 in [Table 3.3](#)), and the numeric target set is the TXOT matrix 'Y' (variable 35 in [Table 3.3](#)). An overview of the process of the neural network is presented in [Figure 3.10](#).

In this case, the network is a 2-layer feed forward network with sigmoid hidden neurons and linear output neurons. Given consistent data and enough neurons in its hidden layer, the network can fit multi-dimensional mapping problems arbitrarily well. The network will be trained using the Levenburg-Marquardt backpropagation algorithm 'trainlm' in MATLAB. The steps taken to create this model are presented in the following subsections.

3.6.2. Dataset stability

For both the RT and NNLM techniques, the stability of the dataset must be assessed. This process can be split into 3 tasks as follows:

1. Standardise the feature matrix 'X' based on the prediction variables in [Table 3.3](#). Also, separate the TXOT from feature matrix X to create the target variable matrix 'Y'.
2. Split matrices 'X' and 'Y' into three subsets, namely, X_{train} and Y_{train} , $X_{validation}$ and $Y_{validation}$, and X_{test} and Y_{test} .
3. Analyse the different data subsets based on their default splitting ratios.

Step 1 is straight forward and was mostly covered in [section 3.2](#). The variables are rearranged in a logical order and prepared in such a way that the X and Y matrices can be used as an input for multiple ML methods.

In step 2, the matrices are split into training, validation, and testing data sets. The default ratios of 0.7, 0.15, and 0.15 were used for training, validation and testing respectively. This is equivalent to 81829, 17535, and 17535 flights per data set. The data set sizes are presented in [Table 3.4](#) for future reference. For the NNLM technique, the training data set is presented to the network and the

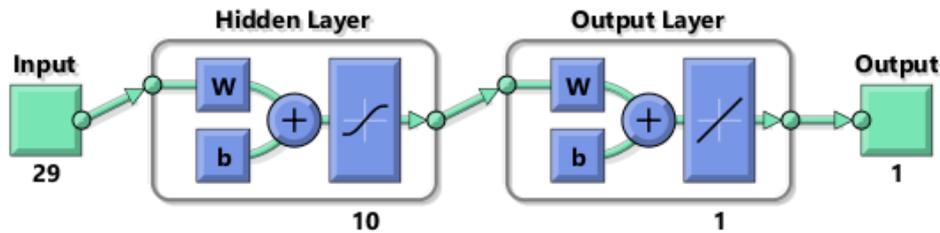


Figure 3.11: NNLM model with chosen parameters

network is adjusted according to its error. The validation set is used to measure network generalization, and to halt training when generalization stops improving. The testing set is used to measure the overall predictive performance of model. This set has no effect on training and therefore provides an independent measure of network performance during and after training [8].

Data Set	Percent	No. flights
Training	70	81829
Validation	15	17535
Testing	15	17535
Total	100	116899

Table 3.4: Data set sizes

The 3rd step involves proving the stability of the datasets. This is accomplished by performing epoch and validation checks, where the number of epochs represents the number of times the algorithm passes through the entire training set when training the model. A convergence check is then performed on the validation set, after which the model is evaluated on the test set.

3.6.3. Parameter selection

Other than the ratios of datasets, the only parameter that can be adjusted is the number of neurons in the fitting network's hidden layer. The default value for this parameter is 10 and is the value used for the 1st iteration of the model. The number of neurons should be changed if the network does not perform well after training. A visual representation of the model with the number of input prediction variables and selected number of hidden neurons can be seen in [Figure 3.11](#).

3.6.4. NNLM results

The model was trained using 116899 flights for all RWY-STD combinations, using all the prediction variables and the default number of neurons. The performance indicators chosen to assess the performance of the model are mean squared error (MSE) and the regression R value. The MSE is the average squared difference between outputs and targets, where lower values are better and 0 means no error. Regression R values measure the correlation between outputs and targets. An R value of 1 indicates a close relationship, while 0 indicates no correlation. Additional performance metrics include the computation time, percentage of predictions within 3 minutes and 5 minutes, traceability, and whether or not there are distributions associated with the individual TXOT predictions. Trace-

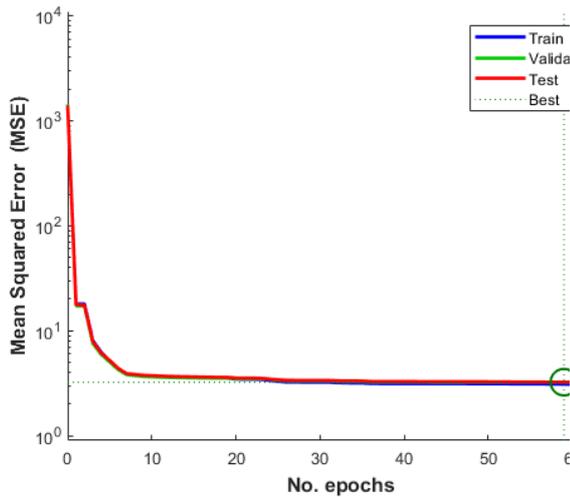


Figure 3.12: Epoch checks: best validation performance (MSE) is 3.235 at epoch 59

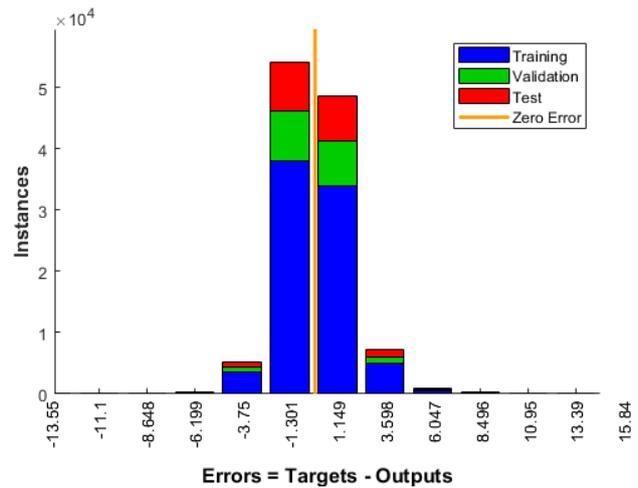


Figure 3.13: Error histogram of results

ability refers to the ability to provide an explanation of how the results were derived. The results of the model and corresponding performance indicators of the data sets can be seen in Table 3.5. The performance displayed is based on the test data set.

Data Set	MSE	R	Computation time (s)	% 3 mins	% 5 mins	Median	Distributions	Traceability
Testing	3.232	0.901	33	92.8	98.59	8.48	Yes	No

Table 3.5: Results NNLM technique

Additionally, the following plots were generated in order to visualise the process and results of the algorithm. In Figure 3.12, the epoch checks are observed, where the best model performance occurs at the 59th epoch producing a validation MSE of 3.235. As can be seen, the epoch checks for the different data sets are almost indistinguishable, therefore confirming the stability of the datasets.

Figure 3.13 corresponds to the results of the 59th epoch, and shows the number of instances (flight predictions) of a prediction error in a certain range. As can be seen, the majority of TXOT predictions have small errors of either -1.301 or 1.149 min for all data sets.

Lastly, in Figure 3.14, the regression plots of different data sets are presented. Here, each point corresponds to a flight, the x-axis represents the target (actual TXOT), and the y-axis represents the output (predicted TXOT). The dotted line represents a perfect prediction accuracy, meaning a flight will find itself on this line if the predicted TXOT is the exact same as the actual TXOT. Finally, the coloured line represents the line of best fit of the points. The closer this line is to the dotted line (higher R value), the better the prediction accuracy of the model.

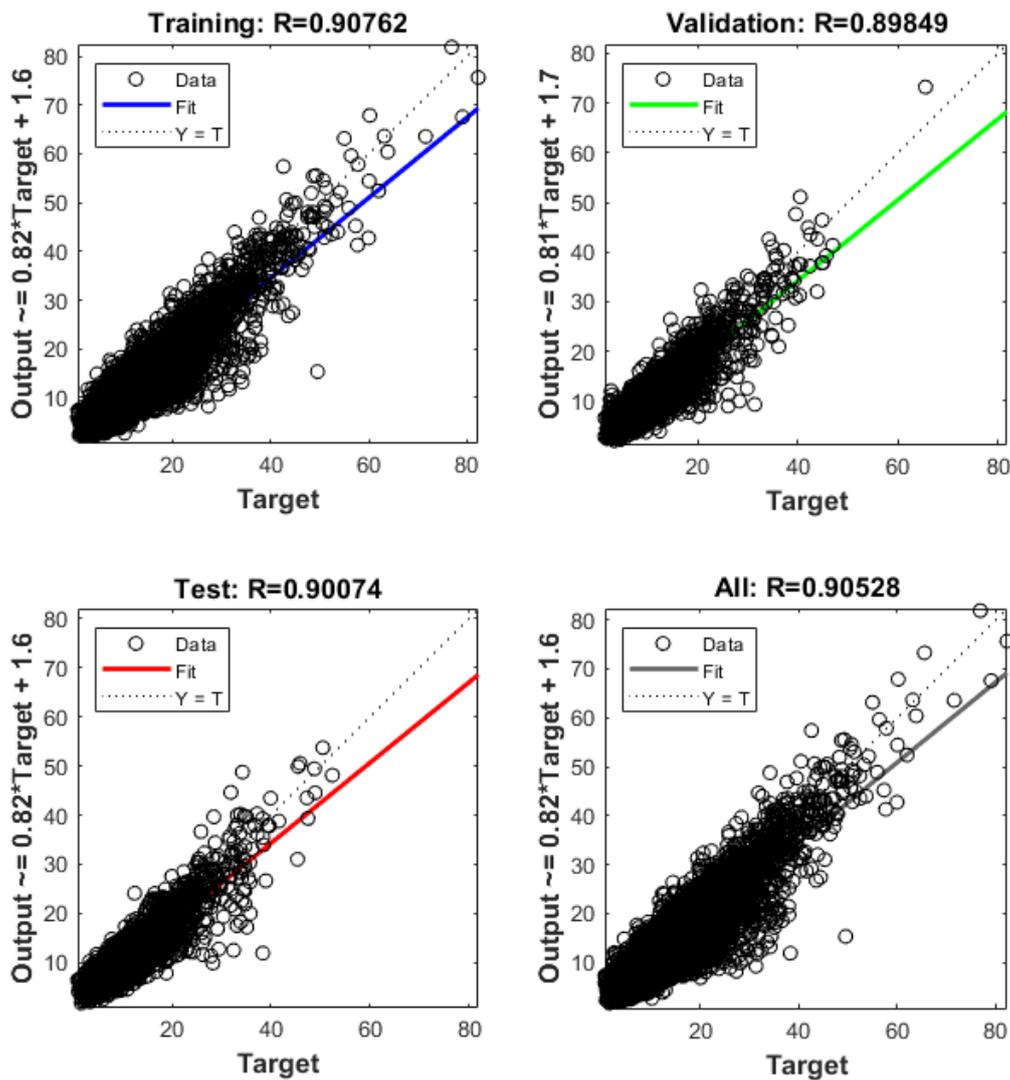


Figure 3.14: Regression plots of the training, validation, testing and combined data sets

3.6.5. NNLM results using feature selection variables

The model is re-trained using only the top features found in section 3.4. The same procedure outlined in subsections 3.6.1-3.6.4 was used. The reasons for using only the top features to train the model were also expressed in section 3.4. The results of this model are displayed in Table 3.6. The epoch checks and error histogram for the model using only the top features are found in figures 3.15 and 3.16 respectively. Lastly, in Figure 3.17, the regression plots of the different data sets are presented.

Data Set	MSE	R	Computation time (min)	% 3mins	% 5mins	Median	Distributions	Traceability
Testing	3.49	0.891	00:30	92.1	98.26	8.48	Yes	No

Table 3.6: Results NNLM technique using top features only

3.6. NEURAL-NETWORK LEVENBURG-MARQUARDT (NNLM) FEASIBILITY

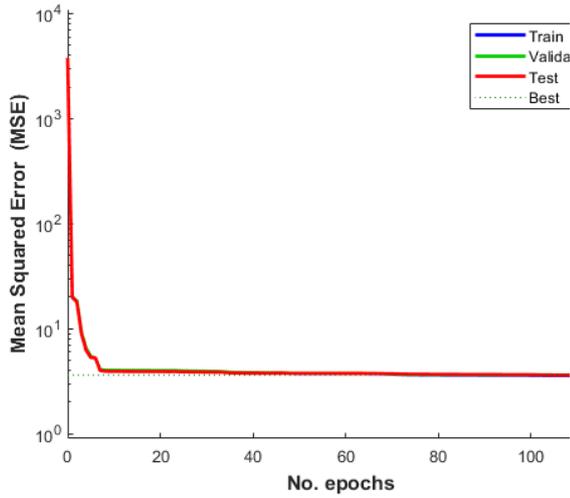


Figure 3.15: Epoch checks for NNLM technique using top features only: best validation performance (MSE) is 3.625 at epoch 112

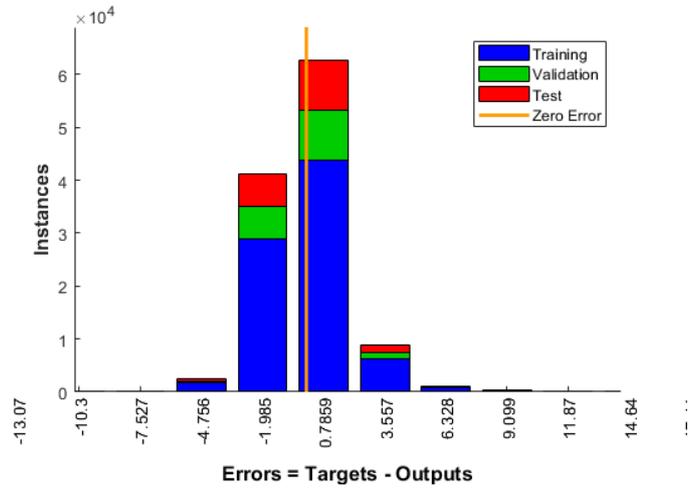


Figure 3.16: Error histogram of results using top features only

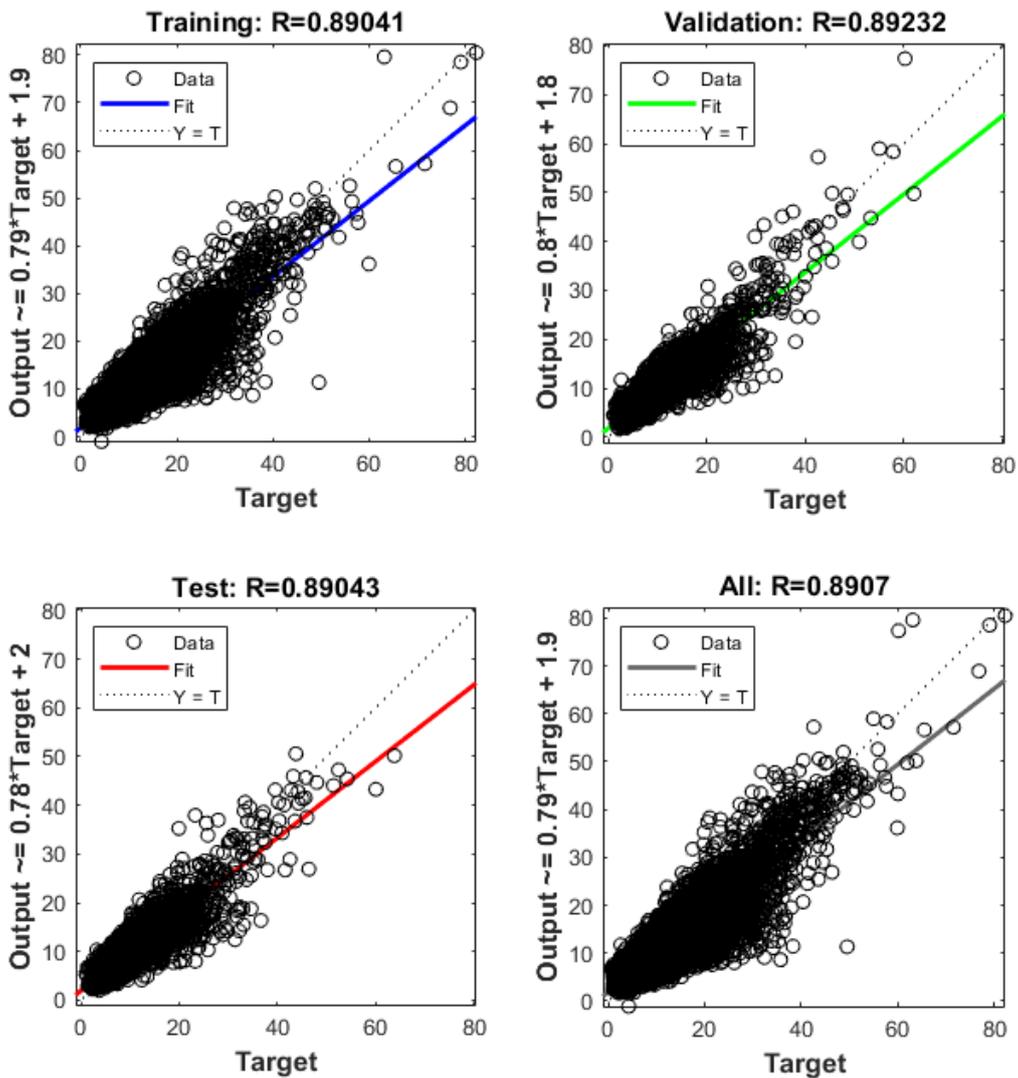


Figure 3.17: Regression plots of the training, validation, testing and combined data sets for NNLM model using top features only

3.6.6. NNLM results comparison

A comparison of the model trained using all variables and the model trained using only the top variables is performed. The results obtained for both models is shown in [Table 3.7](#).

As can be seen, both models have very similar results. However, by excluding 14 variables, the model is trained faster and is more robust when inserting new data with a similar structure.

Features	Data Set	MSE	R	Computation	%	%	Median	Distributions	Traceability
				time (s)	3 mins	5 mins			
All	Test	3.23	0.901	33	92.8	98.59	8.48	Yes	No
Top	Test	3.49	0.891	31	92.1	98.26	8.48	Yes	No

Table 3.7: Comparison NNLM results

3.7. Regression Tree (RT) feasibility

In this section, the initial set up and results of the regression tree technique will be discussed. Having prepared the feature matrix and tested the stability of the data in [subsection 3.6.2](#), the data could directly be used in the regression tree. The results of the tree will be presented in [subsection 3.7.2](#). Before the model could be trained however, a validation scheme had to be chosen. The concept of validation methods for RTs is discussed in [subsection 3.7.1](#).

3.7.1. Validation scheme selection

In order to examine the predictive accuracy of a trained tree, a validation method had to be chosen. Using validation helps to estimate model performance on new data, and is an important step to protect against overfitting. A model that is too flexible and suffers from overfitting has a worse validation accuracy. Options for the validation method include cross-validation, holdout validation, or no validation. The options are discussed below:

- **Cross-Validation:** Cross-validation is a good technique to test a model on its predictive performance. While a model may minimize the Mean Squared Error on the training data, it can be optimistic in its predictive error. The partitions used in cross-validation help to simulate an independent data set and get a better assessment of a model's predictive performance. The method requires multiple fits, but makes efficient use of all the data, so it works well given the data set is not too large. Cross validation is performed in MATLAB using the 'crossval' function. It requires a positive integer 'k' as input and performs the following:
 1. Partitions the data into 'k' disjoint sets or folds
 2. For each fold:
 - (a) Trains a model using the out-of-fold observations
 - (b) Assesses model performance using in-fold data
 3. Calculates the average test error over all folds
- **Holdout Validation:** Select a percentage of the data to use as a validation set. The app trains a model on the training set and assesses its performance with the validation set. The model

3.7. REGRESSION TREE (RT) FEASIBILITY

used for validation is based on only a portion of the data, so holdout validation is appropriate only for large data sets. The final model is trained using the full data set.

- No Validation: if no validation method is chosen, the model will have no protection against overfitting. The app uses all the data for training and computes the error rate on the same data. Without any test data, you get an unrealistic estimate of the model's performance on new data. That is, the training sample accuracy is likely to be unrealistically high, and the predictive accuracy is likely to be lower. While there may be specific cases where no validation scheme needs to be chosen, in most cases one should be selected to avoid overfitting the data.

For this tree, the default validation method was chosen, namely, cross validation with 5 folds (k=5).

3.7.2. RT results

The data was once again split into training, validation, and testing data sets with a ratio of 0.7, 0.15, and 0.15 respectively. As with the neural network, 2 models were trained using the training data set, one using all the prediction variables, the other using only the top variables. For both models the default parameters were used. The parameters that can be adjusted are minimum leaf size and maximum tree depth, whose concepts will be discussed in [section 4.1](#). The default minimum leaf size for a 'coarse' tree is 36, and there is no maximum tree depth. However, the resulting tree depth after having trained the model is 26. The results for the testing data set of these models can be found in [Table 3.8](#). The corresponding result plots for the model trained with all variables can be found in [figures 3.18](#) and [3.19](#). The plots for the model trained with only the top variables are shown in [figures 3.20](#) and [3.21](#).

Features	Data set	MSE	R	Computation time (s)	% 3mins	% 5mins	Median	Distributions	Traceability
All	Test	3.27	0.902	5.8	93.43	98.47	8.44	Yes	Yes
Top	Test	3.4	0.897	5.1	93.17	98.25	8.46	Yes	Yes

Table 3.8: RT results for models trained using all features and top features only

3.8. ML TECHNIQUE COMPARISON AND SELECTION

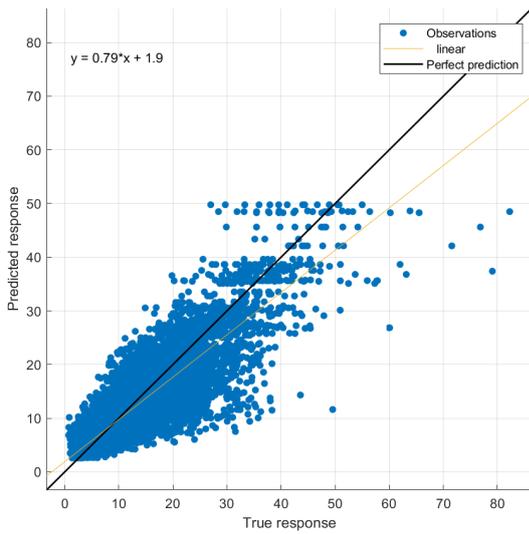


Figure 3.18: Regression plot for testing data set using all features

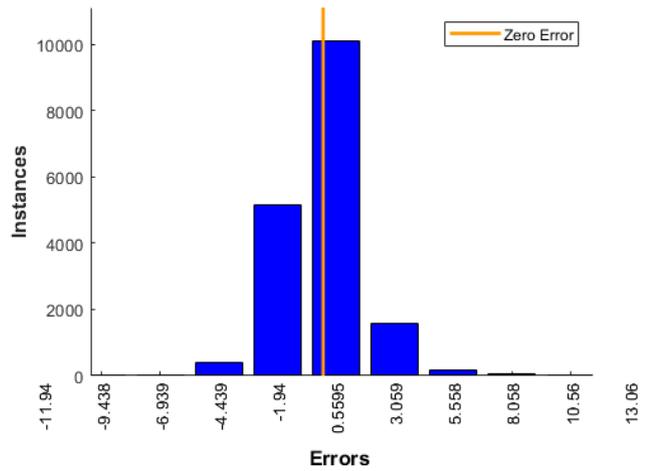


Figure 3.19: Error histogram for tree trained with all features

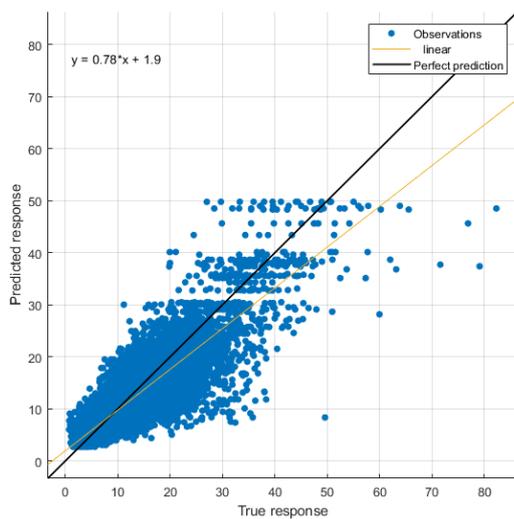


Figure 3.20: Regression plot for testing data set using top features only

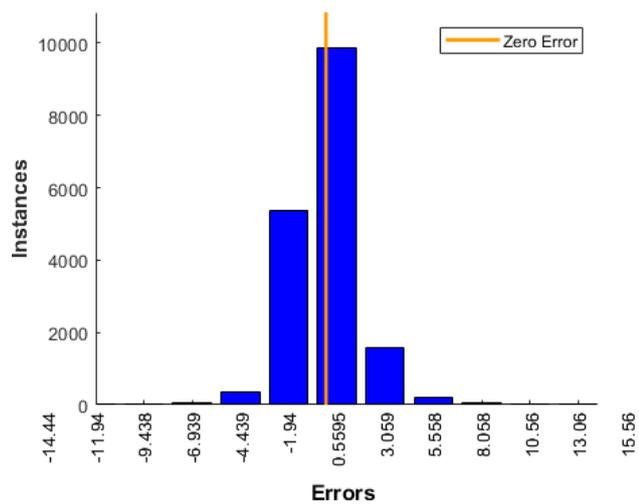


Figure 3.21: Error histogram for tree trained with top features only

Clearly, reducing the number of features only results in a minimal increase in MSE. However, the model trains faster, and is more robust when inserting new data with a similar structure.

3.8. ML technique comparison and selection

A comparison of the results obtained in sections 3.6 and 3.7 is shown in Table 3.9.

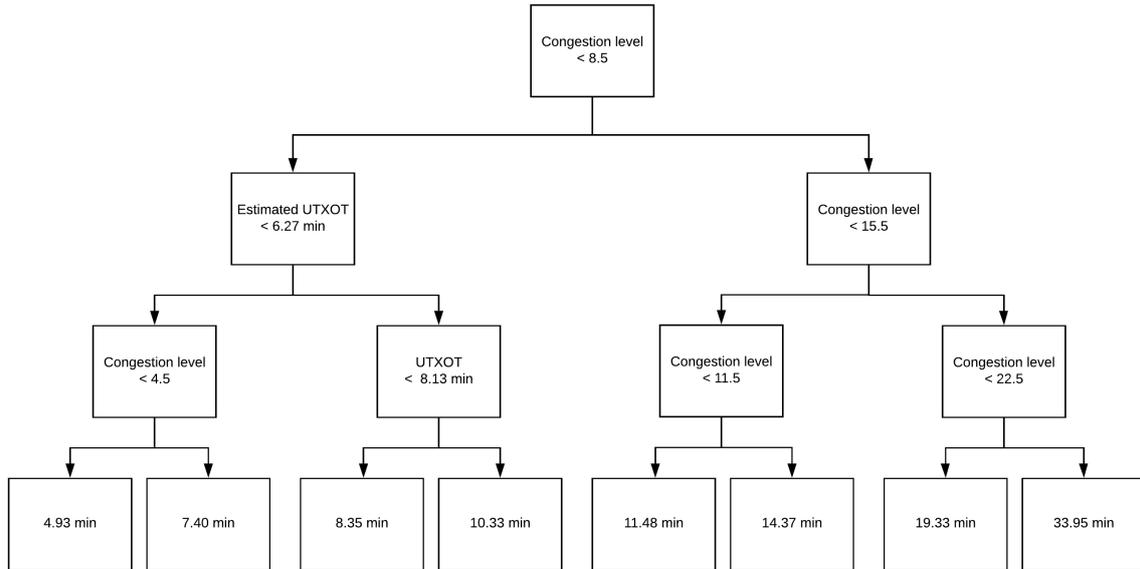


Figure 3.22: Regression tree example

Technique	Features	MSE	R	Computation time (s)	% 3 mins	% 5 mins	Median	Distributions	Traceability
NNLM	All	3.23	0.901	33	92.8	98.59	8.48	Yes	No
NNLM	Top	3.49	0.891	31	92.1	98.26	8.48	Yes	No
RT	All	3.27	0.902	5.8	93.43	98.47	8.44	Yes	Yes
RT	Top	3.4	0.897	5.1	93.17	98.25	8.46	Yes	Yes

Table 3.9: Comparison ML techniques

As can be seen, the performance for both techniques is very similar, with the regression tree performing marginally better for most of the performance indicators. The two exceptions to this are the computational time and the traceability. The computation of the RT is roughly 6 times faster, and most importantly, the results are traceable.

One of the goals of this study is to create a useable predictability model for ATCOs. In order for a model to be useable by ATCOs, there must be a way to explain how a prediction is made. When looking at a decision tree, an example of which is shown [Figure 3.22](#), it is easy to see which decisions were made by the regression tree to reach a certain prediction. An initial variable divides the data into two categories, each of which is further split into two until a stopping rule is applied. As such, the ATCO can clearly provide an explanation for the result based on the decisions made in the tree; the result is "traceable". Additionally, this information is very useful in the understanding of the underlying nature of the taxi-out process as the variables used in making the decisions can be observed. This is in alignment with another of the goals of this study, namely, to find which variables most significantly influence TXOT.

In contrast, a neural network is more of a "black box" that delivers results without an explanation of how the results were derived. Thus, it is difficult or impossible to explain how decisions were made based on the output of the network. For example, if an ATCO wanted to challenge a prediction made

by the network they could not do so as the decisions made by the network are not "traceable".

As mentioned at the start of [section 3.6](#), other ML techniques such as Reinforcement Learning (RL) and Multilayer Perceptron (MLP) were also considered. However, these techniques also have the "traceability" issue. The only technique which provides interpretable and traceable results is the regression tree. As such, these other techniques were not explored in depth. For these reasons, the Regression Tree was chosen as the most suitable technique for TXOT prediction. A further analysis of this technique will be presented along with the results in [chapter 4](#).

3.9. Extended Taxi-out Times

While several studies aiming to predict normal TXOTs and observe key related precursors have been performed, little research has been performed regarding the prediction of abnormal or extended taxi-times and their key related precursors. A study was therefore performed which aims to predict longer taxi-out times, and observe the features which influence these flights most significantly.

After having performed the feasibility tests of 2 ML techniques in [sections 3.6](#) and [3.7](#), and having compared these techniques in [section 3.8](#), it was determined that the RT technique was most suitable for predicting TXOTs. The reasons for selecting the RT technique for normal TXOT prediction can also be applied to extended TXOT prediction. It was therefore decided that a RT would also be used to predict extended TXOTs and observe the most influential features. Additionally, much of the data and algorithm had already been prepared in [section 3.7](#), and therefore only had to be adapted for new data inputs. The model creation and parameter determination, along with the results will be presented in [chapter 4](#).

First, however, an analysis was performed to determine which flights should be considered to 'extended' TXOTs. To determine this, the MATLAB function 'isoutlier' was used. Here, two options were considered:

1. Consider TXOTs to be 'extended' if the TXOT is more than 3 standard deviations (3σ) from the mean TXOT of all flights.
2. Consider TXOTs to be 'extended' if the TXOT is more than 2 standard deviations (2σ) from the mean TXOT of all flights.

In [Table 3.10](#), the statistics relating to the different thresholds are observed. Given a threshold of 3σ from the mean, 1678 flights can be considered to have 'extended' TXOTs. These flights comprise 1.44% of the total flights, and can be seen in [Figure 3.23](#), where the threshold is set at 21.48 mins. Given a threshold of 2σ from the mean, 4050 flights can be considered to have 'extended' TXOTs. These flights comprise 3.46% of the total flights, and can be seen in [Figure 3.24](#), where the threshold is 17.3 mins.

Threshold (σ)	Threshold (min)	No. flights	Percentage
3σ	21.48	1678	1.44
2σ	17.3	4050	3.46

Table 3.10: Thresholds for 'extended' TXOTs

3.9. EXTENDED TAXI-OUT TIMES

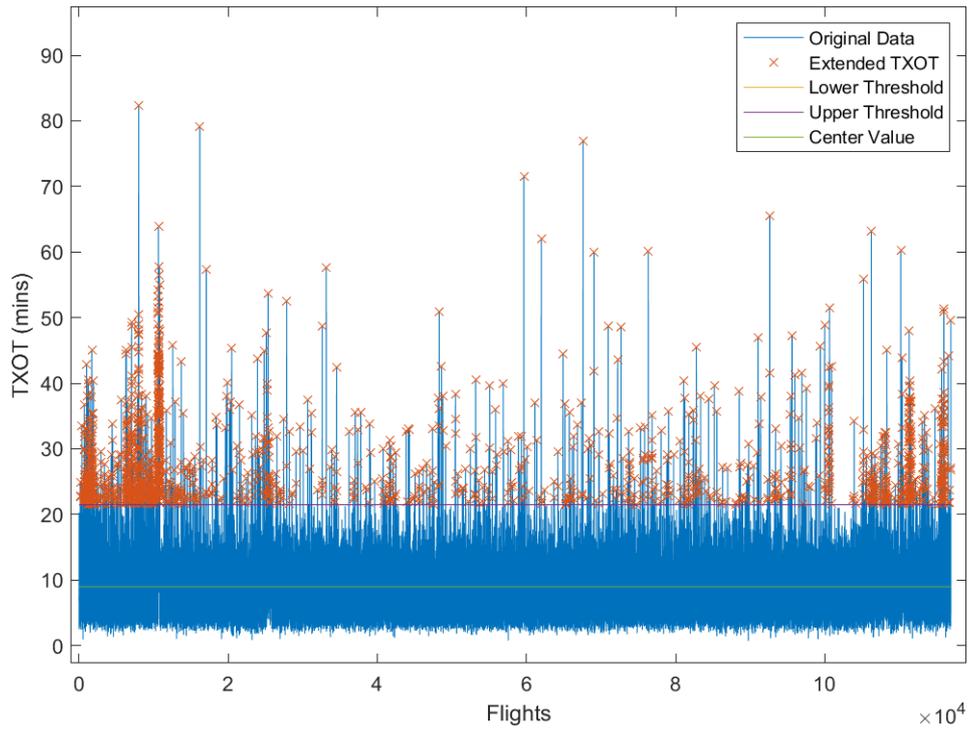


Figure 3.23: Threshold at 3 standard deviations (3σ) from the mean

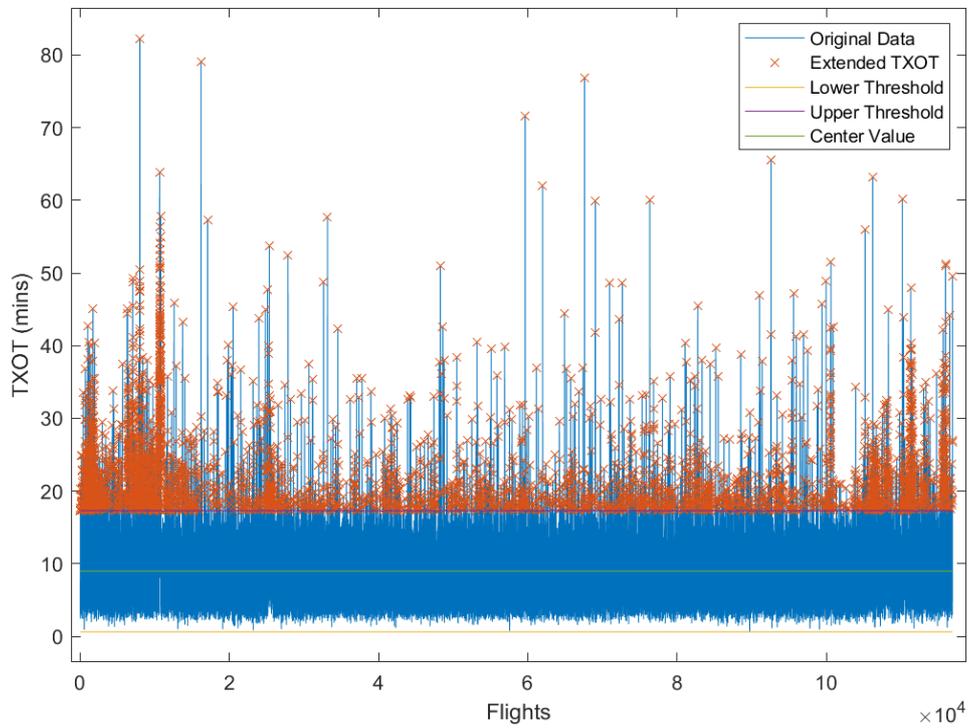


Figure 3.24: Threshold at 2 standard deviations (2σ) from the mean (17.3 mins)

3.10. EXTENDED TXOT ANALYSIS

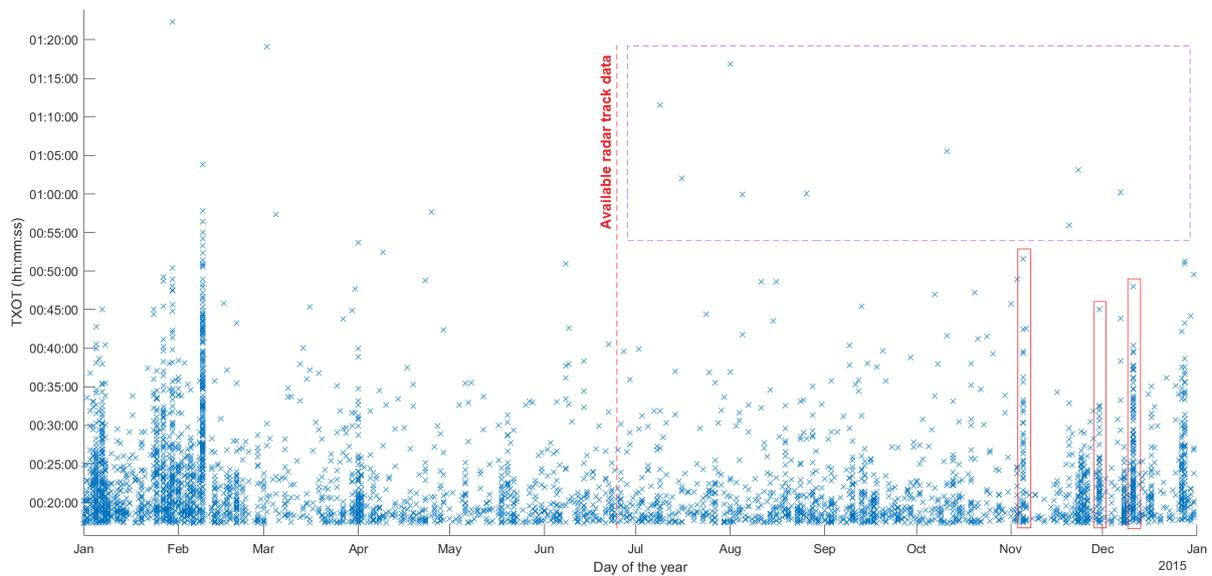


Figure 3.25: Extended flight TXOTs and areas of interest

A decision was made to set the threshold at 2σ based on the fact that the 1678 flights that would be obtained using a threshold of 3σ is not enough to produce meaningful results using a regression tree. In general, the more data that is used to train a regression tree, the more accurate it becomes. Using the 4050 flights after the 2σ threshold was therefore considered more appropriate. Visual inspection of Figure 3.24 shows that the bulk of the data lies below the threshold and reinforces the selection at this level.

3.10. Extended TXOT analysis

An analysis was performed in order to determine the causes of certain extended TXOTs that could not be captured in the RT model. To analyse these TXOTs, the Radar Track data was used to visualise the ground tracks of these aircraft. However, since Radar track data was only available from June 25th 2015 onwards, only these flights could be analysed. In Figure 3.25, the TXOTs of all 'extended' TXOT flights are displayed, where the dashed red line represents the date after which the Radar Track data is available. Within the extended TXOT flights two specific areas of interest were analysed, namely, flights with a TXOT greater than 50 mins (purple box in Figure 3.25), and days in which there are a large number of extended TXOTs (red boxes in Figure 3.25).

3.10.1. TXOTs greater than 50 mins

In Figure 3.26, flights with TXOTs greater than 50 minutes within the purple box are observed. Each of the flights are numbered and are analysed individually by plotting their ground track and speed profile. Each of the data points correspond to real life flights, however, due to confidentiality reasons, the flight number cannot be displayed.

3.10. EXTENDED TXOT ANALYSIS

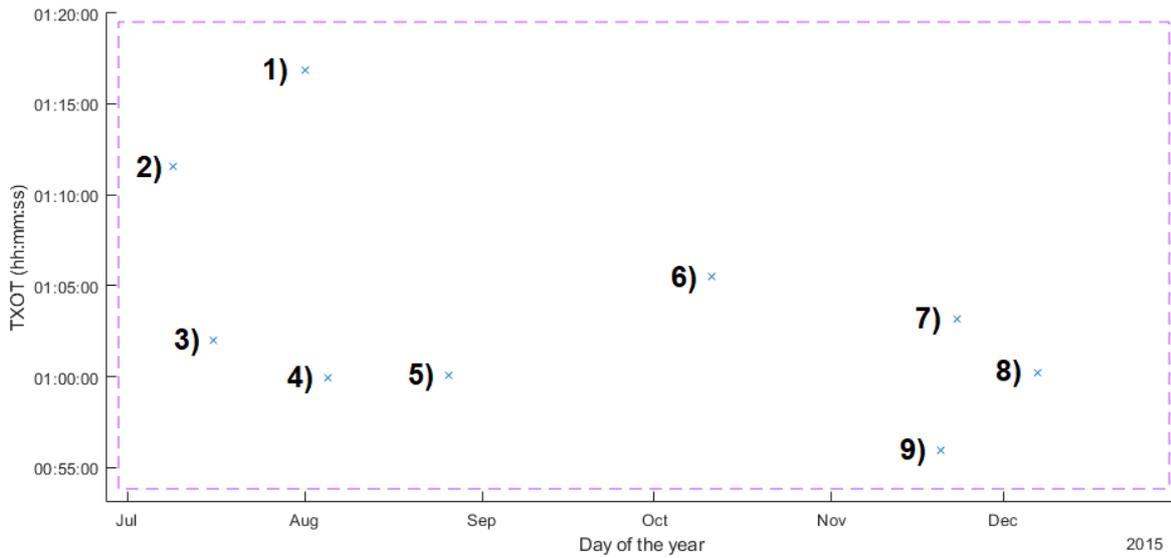


Figure 3.26: Extended TXOT outlier analysis

1. 01-Aug-2015 - Flight 'A'

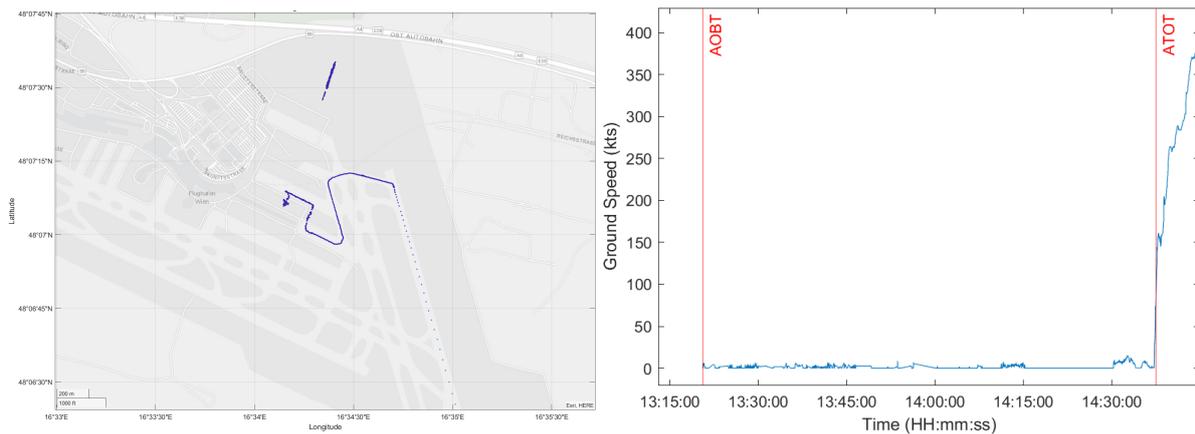


Figure 3.27: Ground track and speed profile for flight 'A'

From the ground track in [Figure 3.27](#), movement detached from the main ground track can be observed in the north. This movement occurs outside of areas in which aircraft can move and is therefore clearly an error in the radar data. This initial false movement triggers a false AOBT indication resulting in a false TXOT of 76 minutes. Based on ground speed profile, it can be seen that the actual AOBT occurs at 14:30 and the ATOT at 14:37 resulting in much more realistic TXOT of 7mins. For simplicity this data entry was removed from the final data set.

2. 09-Jul-2015 - Flight 'B'

3.10. EXTENDED TXOT ANALYSIS

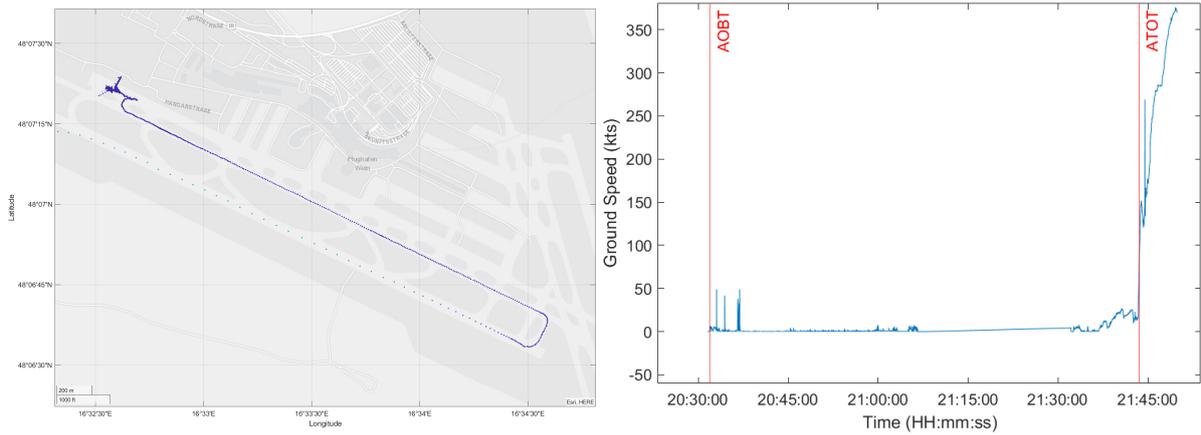


Figure 3.28: Ground track and speed profile for flight 'B'

In [Figure 3.28](#) many data points can be observed in and around the apron area just after push-back, indicating that the aircraft spent a lot of time here. In this case, the reason for the long TXOT is that the aircraft has been pushed back, and for some unknown reason has been forced to wait in this area until roughly 21:32, almost 1 hour after the AOBT of the aircraft. The aircraft then takes a realistic further 11 minutes to complete the taxi-out process, departing at 21:43. Due to this unknown reason and for simplicity, the flight was excluded from the final dataset.

3. 16-Jul-2015 - Flight 'C'

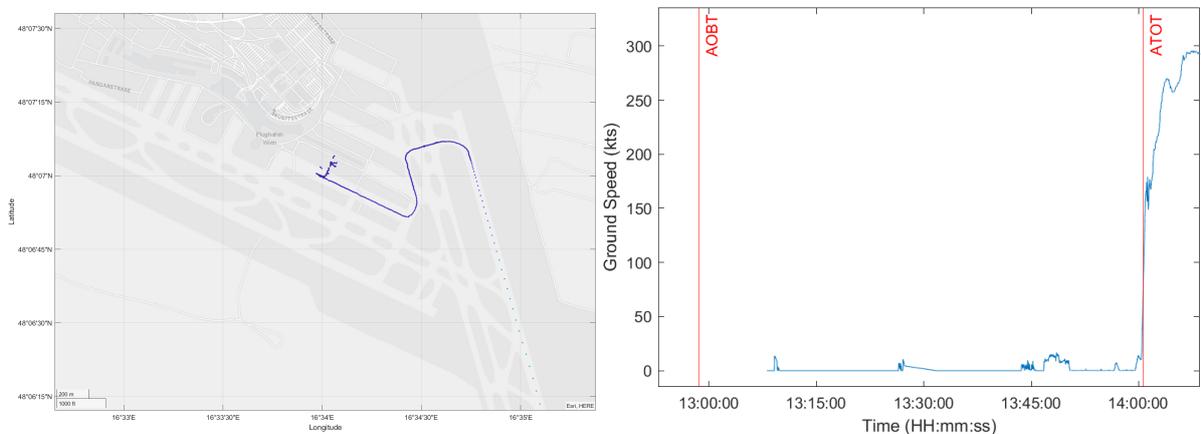


Figure 3.29: Ground track and speed profile for flight 'C'

From the ground speed profile in [Figure 3.29](#) it can be seen that the AOBT is registered before any movement has actually occurred. Additionally, two random spikes occur at 13:09 and 13:27 while the aircraft is actually at a standstill. These spikes can be observed in the ground track profile where there are small groups of points disjointed from the main ground track. These are caused by errors in the radar where the aircraft position is incorrectly registered. In reality, the AOBT does not occur until 13:43. With the ATOT at 14:00, the resulting TXOT is a much more realistic TXOT of roughly 17 mins. For these reasons, this flight is excluded from the final

dataset.

4. 05-Aug-2015 - Flight 'D'

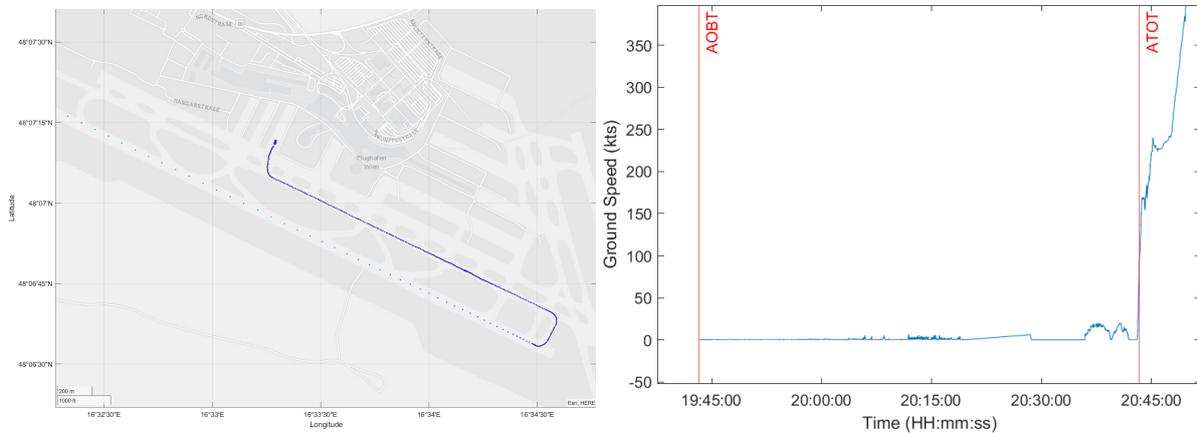


Figure 3.30: Ground track and speed profile for flight 'D'

This extended TXOT is once again caused by errors from the radar. As with other cases, the AOBT is registered before the aircraft actually starts moving. A possible cause of this early AOBT registration is that some calibration is required when the aircraft's onboard systems are activated. This calibration causes the aircraft to appear to be moving, thus an AOBT is registered. By observing the speed profile, the AOBT does in fact not occur until 20:36 resulting in a TXOT of roughly 7 mins rather than the initial 60 mins. As such, this entry is removed.

5. 26-Aug-2015 - Flight 'E'

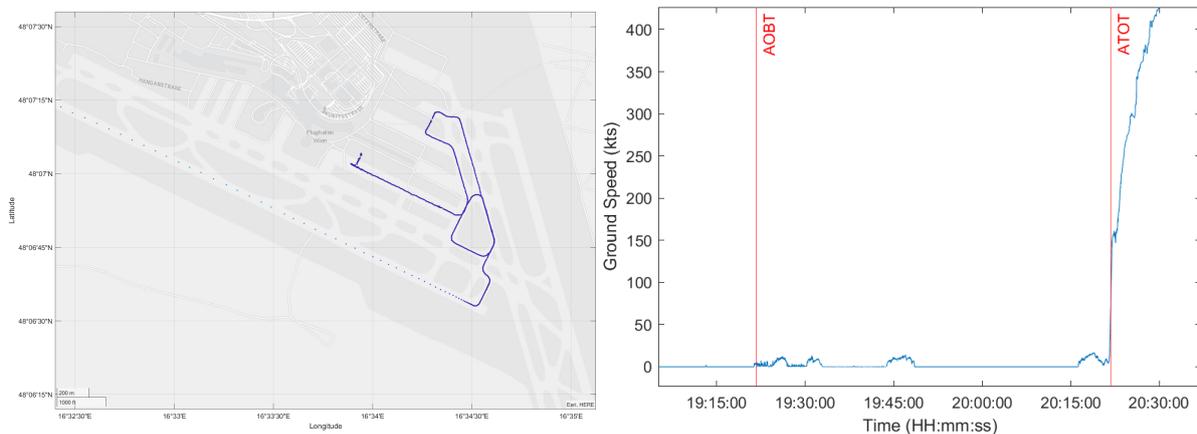


Figure 3.31: Ground track and speed profile for flight 'E'

From [Figure 3.31](#) it is clear that the ground track followed is irregular. The aircraft initially departs from gate F03, performs a loop around one of the taxi-ways and re-parks at stand H48 before proceeding to take-off on RWY29. The reason for this detour is unclear, and the flight is therefore excluded from the final data set.

6. 11-Oct-2015 - Flight 'E'

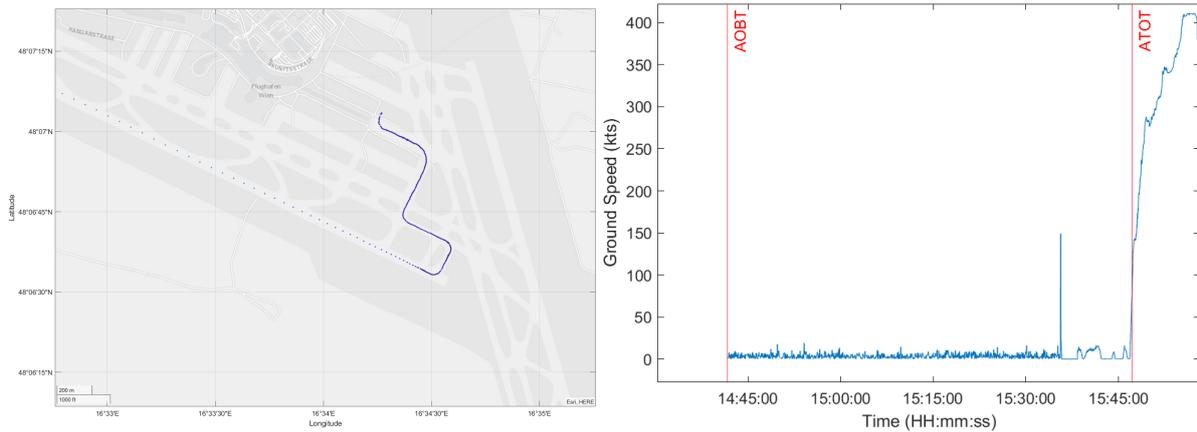


Figure 3.32: Ground track and speed profile for flight 'E'

The speed profile of Figure 3.32 shows that this outlier is once again caused by errors from the radar. A false AOBT is registered long before the actual AOBT (15:38), after which the ground speed fluctuates irratically. This is proceeded by a huge spike in ground speed reaching 150 kts which is clearly not feasible for ground operations. Given the actual AOBT at 15:38 and the ATOT at 15:47, the actual TXOT of the flight is roughly 9 mins rather than the initial 65 mins. As such, this record is excluded from the final data set.

7. 23-Nov-2015 - Flight 'F'

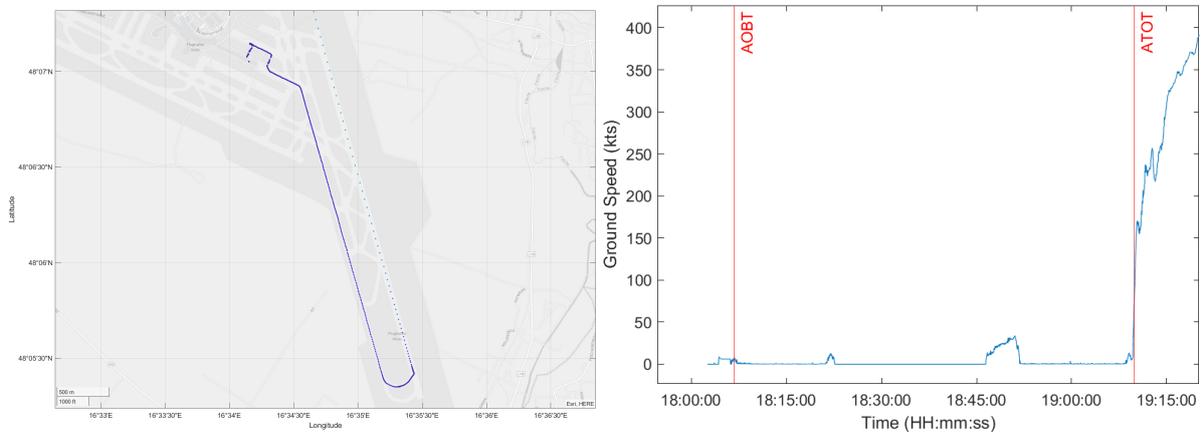


Figure 3.33: Ground track and speed profile for flight 'F'

In the speed profile of Figure 3.33, 3 peaks can be observed before the take-off. The first represents the pushback, and corresponds with the AOBT at 18:06. Next, after a 15 minute wait, the aircraft performs a small movement at 18:21. The movement represents the aircraft taxiing into de-icing stand F44, where the 15 minute wait preceding this movement was required due to the de-icing stands being occupied. De-icing stand will be further discussed in subsection 3.10.2. The aircraft then waits a further 25 minutes while being de-iced before taxiing to RWY34 at 18:46. Finally, the aircraft waits another 18 minutes at the holding position before taking off

3.10. EXTENDED TXOT ANALYSIS

19:08 resulting in a TXOT of 63 minutes. While in this case the TXOT is in fact accurate, the data point is excluded for convenience. This will be further discussed at the end of this section.

8. 07-Dec-2015 - Flight 'G'

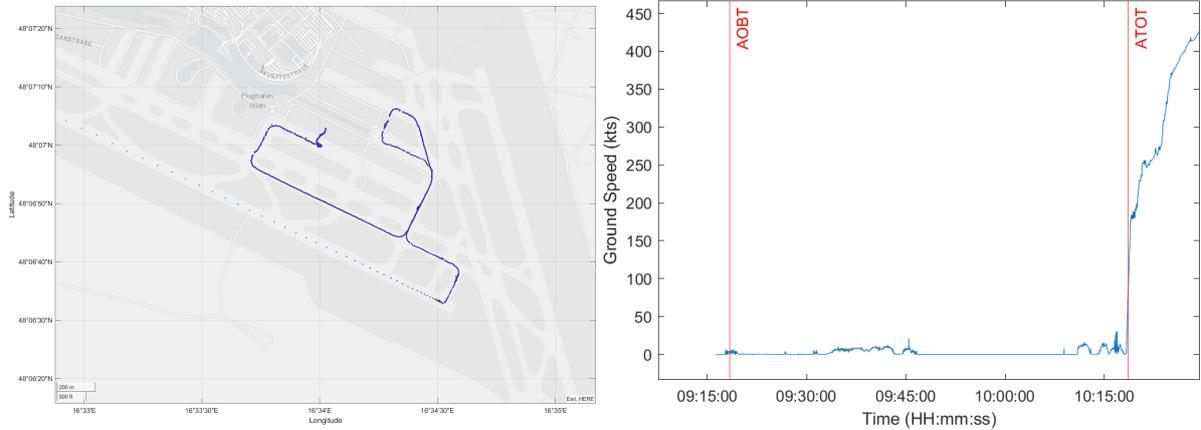


Figure 3.34: Ground track and speed profile for flight 'G'

Like Figure 3.33, the extended TXOT in Figure 3.34 is caused by the de-icing of the aircraft. The aircraft performs a significant detour to reach the de-icing stand and has to wait there for almost half an hour. As with Figure 3.33, this flight was excluded for convenience and will be further discussed at the end of this section.

9. 20-Nov-2015 - Flight 'H'

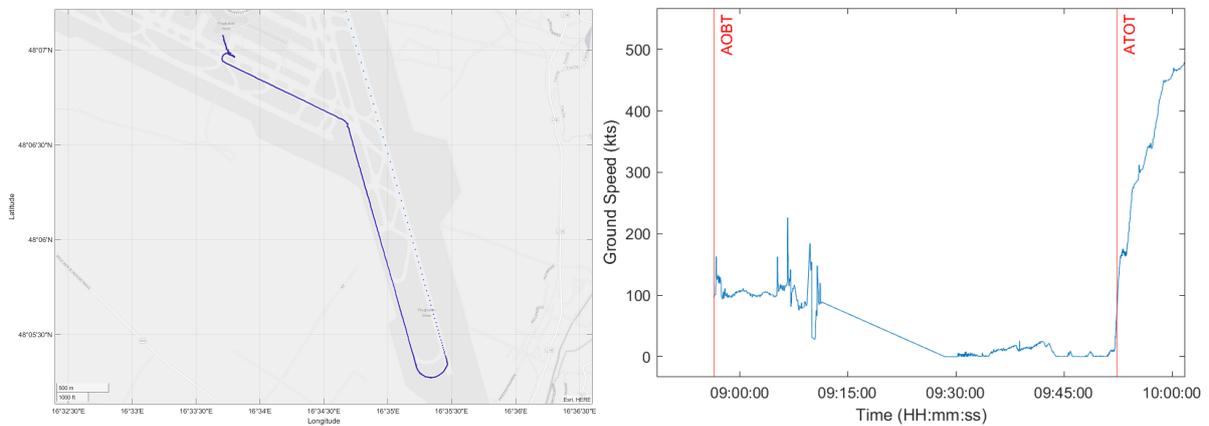


Figure 3.35: Ground track and speed profile for flight 'H'

From the speed profile in Figure 3.35, an error has clearly occurred with the radar. False ground speeds ranging from 100 - 230 kts are recorded at the AOBT which is not feasible for ground movements. Additionally, flight levels of between 2400-3100 ft are recorded at these times confirming the fault in the radar. Given the actual AOBT at 09:30 and ATOT at 09:52, the actual TXOT should be roughly 22 mins rather than the indicated 55 mins. For these reasons, this flight is excluded from the final data set.

Based on the analysis of these ‘outliers’, it was observed that most TXOTs longer than 55 mins were caused by radar errors. While only flights after June 25th could be analysed, it is assumed that flights longer than 55 mins were also caused by errors in the radar data. Therefore, all flights with a TXOT longer than 55 minutes which occur before June 25th are excluded from the final data set. This is with the exception of the flights on February 9th. Based on the larger number of flights with extended TXOTs, it can be deduced that these flights all required de-icing. As such, flights with a TXOT greater than 55 mins on this day were kept in the final dataset.

3.10.2. Extended TXOTs due to de-icing

The next points of interest are those found in the red boxes displayed in Figure 3.25. Once again these points were analysed by plotting the ground tracks and speed profile of the flights in question. An example of one such flight, flight ‘I’, can be seen in Figure 3.36.

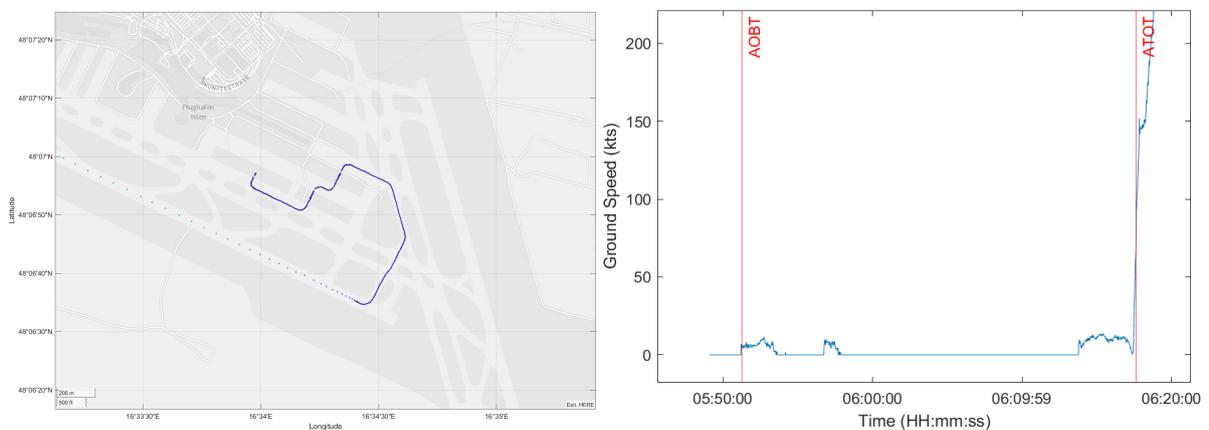


Figure 3.36: Ground track and speed profile for flight ‘I’ on 05-Nov-2015

As observed in Figure 3.36, the AOBT and ATOT after been registered correctly. The extended TXOT is therefore not due to radar errors. Instead, this is due to the fact that the aircraft requires de-icing. From the speed profile the aircraft can be seen to have taxied for a small distance before coming to a stand still from 05:53 - 05:56. This is the time the aircraft spends waiting at the de-icing standby position. The aircraft then taxis into the de-icing stand, as shown by the 2nd peak in the speed profile. The aircraft then waits from 05:57 - 06:13, during which the aircraft is being de-iced. Finally, the aircraft proceeds to RWY29 for take-off at 06:17.

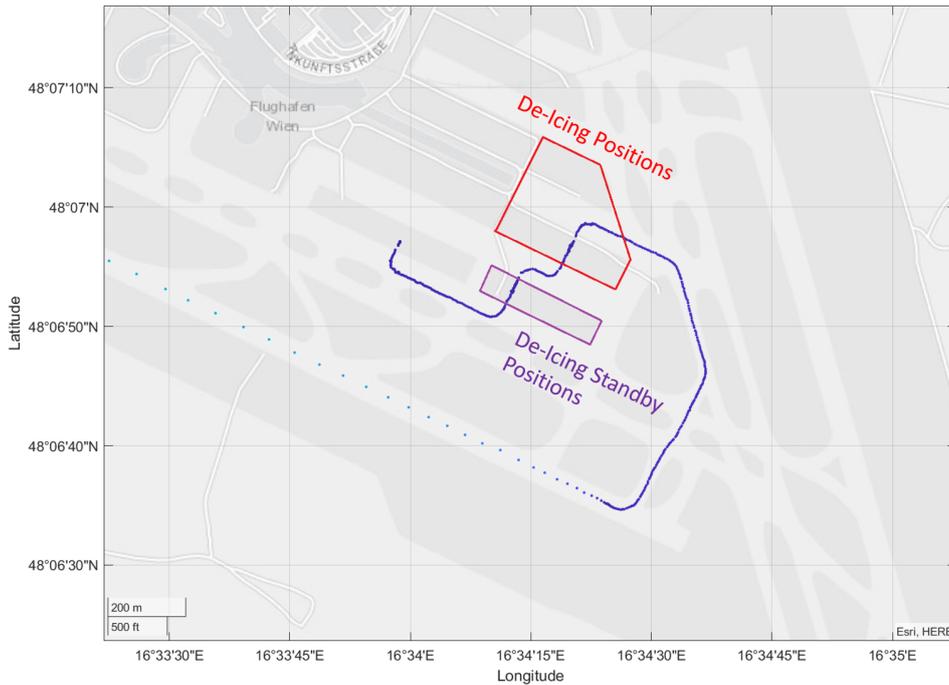


Figure 3.37: De-icing standby and de-icing positions

Having plotted all flights in the red boxes, it can be observed that all these flights required de-icing. The de-icing stands and de-icing standby positions are shown in [Figure 3.37](#). The de-icing positions are as follows:

- F43, F45, F47, F49
- F51, F53, F55, F57, F59
- F42, F44, F46, F48, F50

De-icing standby positions:

- E48, E49, E50, E51, E52
- E97, E98, E99

If an aircraft passes through any of these positions, it is assumed that they required de-icing. Clearly the de-icing of an aircraft has a significant impact on the TXOT. A variable relating to de-icing stand would therefore likely increase the predictive accuracy of the model. However, since radar track data was only available up until June 25th, this feature could not be added. For future work, more data availability would therefore be beneficial.

4

Results normal TXOT

Having determined in [chapter 3](#) that the regression tree was the most appropriate technique for predicting TXOT, the technique is further explored. The purpose of creating a regression tree is to extract a set of if-then-else split conditions in order to identify the precursors which have the most influence on TXOT. Starting at the root node, a series of questions are asked about the predictors. The tree then chooses the predictor which achieves the minimum MSE between the prediction and the actual TXOT after the split. This process is then repeated for each subsequent node until a stopping rule is applied. Each terminal node, or “leaf”, then consists of a certain number of flights from which a distribution and predicted TXOT can be extracted from. The aforementioned stopping rule takes the form of the minimum leaf size or the maximum tree depth. These are the parameters that must be adjusted in order to create a model that is accurate and not overcomplicated. Since 2 different RTs are trained (normal TXOT and extended TXOT), the parameters will need to be tuned for each. An example regression tree with tree depth 3 is shown [Figure 3.22](#) to help visualise these concepts.

In this chapter, the selection of these tuning parameters are presented. For normal TXOT prediction, the parameter selection, along with the results of the tree, will be presented in [section 4.1](#). For extended TXOT prediction, the results and parameter selection will be presented in [chapter 5](#). Finally, an additional analysis of the effects of the top features on TXOT is performed in [section 4.3](#).

4.1. Normal TXOT analysis

In [chapter 3](#), the regression trees were trained using the default settings and values. This included the defaults for the minimum leaf size, and maximum tree depth. In subsections [4.1.1](#) and [4.1.2](#) these parameters will be further optimised to create a final RT model for normal TXOTs.

4.1.1. Leaf Size Determination

The first parameter is the minimum leaf size l_{min} , for which enough data points are required in each terminal node to create a distribution while still producing accurate results. A fine tree with many small leaves is usually highly accurate on the training data. However, the tree might not show comparable accuracy on an independent test set. A very leafy tree tends to overfit, and its validation accuracy is often far lower than its training (or resubstitution) accuracy. In contrast, a coarse tree with fewer large leaves will not achieve a high training accuracy. A coarse tree can, however, be more robust in that its training accuracy can be near that of a representative test set.

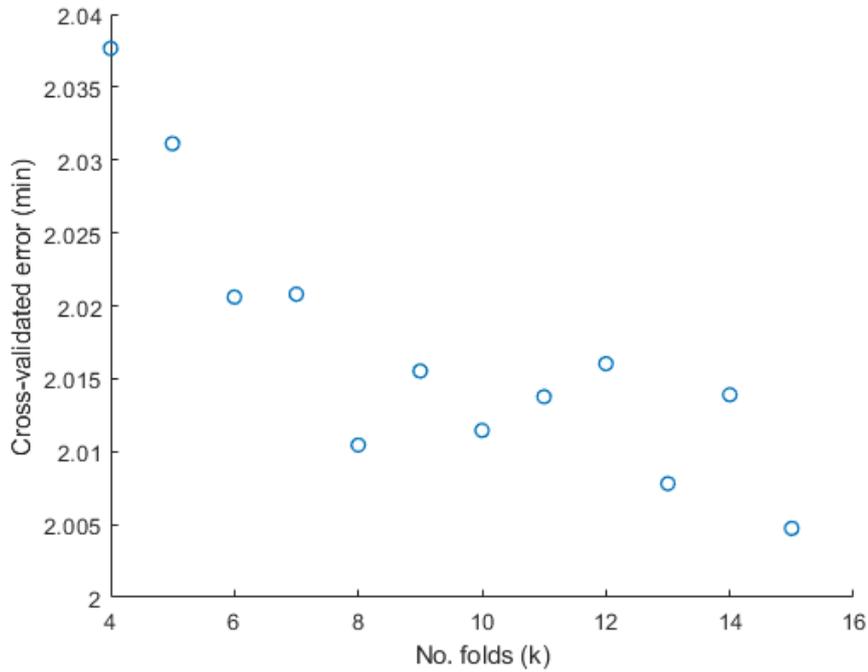


Figure 4.1: Cross-validated error for varying number of folds k

k-fold determination

To determine the minimum leaf size, cross validation is used. This concept was explained in [section 3.7](#). As mentioned, cross validation requires the number of folds ‘k’ as input. An analysis was performed to determine how the number of folds affects the prediction accuracy. This was done using the default values for tree depth and leaf size. The model was trained multiple times with varying k-fold values ranging from 4-15, the results of which can be seen in [Figure 4.1](#).

While the accuracy of predictions tends to increase with increasing folds, this increase in accuracy is negligible. Between 4-folds and 15-folds there is a difference in error of 0.033. Since a larger number of folds also requires a larger computational time, a lower number of folds is preferred. A conservative k-value of 8 was chosen to achieve near optimal accuracy while not requiring too much computational time.

Results minimum leaf size determination

Having selected the number of folds, an analysis was performed to select the minimum leaf size l_{min} . The minimum leaf size was varied from 10-100 using 8-fold cross validation, the results of which can be seen in [Figure 4.2](#).

From the figure it can be seen that the minimum cross-validated error occurs at a minimum leaf size of 28. It is important to note, however, this is the performance error on the training set. In order to avoid over-fitting, a slightly larger leaf size should be chosen. Additionally, a larger leaf size means more data points to which a more accurate distribution can be fitted. A leaf size of 42 was therefore chosen.

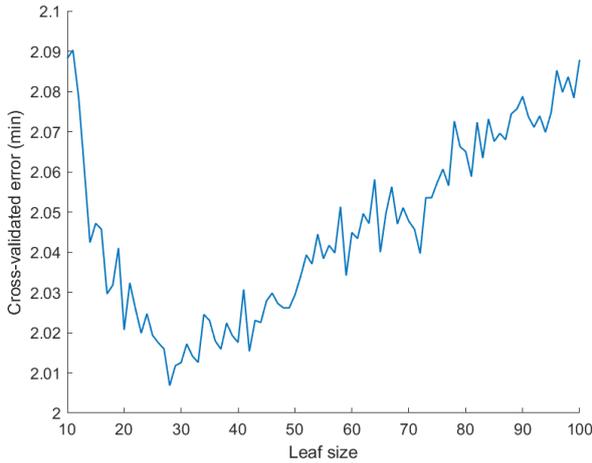


Figure 4.2: Cross-validated error for varying minimum leaf size l_{min}

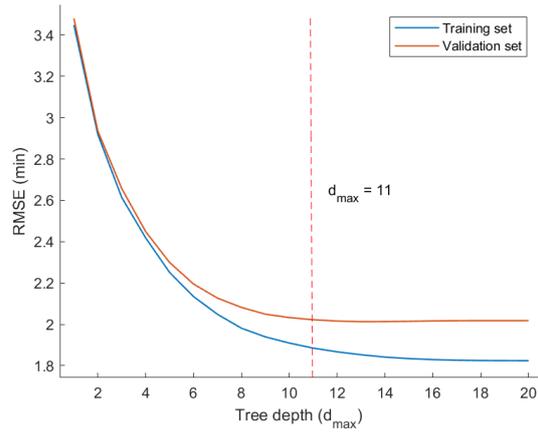


Figure 4.3: RMSE vs. tree depth for training data set and validation data set

4.1.2. Tree Depth determination

Having chosen the optimal leaf size, the second tuning parameter, maximum tree depth (d_{max}), had to be determined. In chapter 3, no maximum tree depth was set as this was the default. The maximum tree depth is used to restrict the number of layers of a tree. This is often done to improve the interpretability of a tree. This is an important factor in this study as the tree needs to be easily interpretable for an ATCO in order to assist with decision making. Additionally, a very large tree with many leaves might overfit the data, whereas a small tree might not be able to capture the important structure of all the variables or top feature variables. It is therefore important to find a balance. The parameter d_{max} is determined using the RMSE. The model is trained multiple times with a varying maximum tree depth from 1-20, the results of which can be seen in Figure 4.3.

The results for both the training set and validation set can be observed in the figure. While the predictive accuracy of the model on the training set may increase past a tree depth of 11, the accuracy on the validation set does not. Were the tree depth to be set any higher than 11, the tree would ‘overfit’ to the training data and be less robust when introduced to new data. Additionally, the greater the tree depth, the more difficult it is to interpret. The maximum tree depth was therefore set at 11 to ensure the interpretability of the tree while maintaining a sufficiently high predictive accuracy.

4.1.3. RT results for normal TXOT prediction

A tree was trained using the selected parameters ($d_{max} = 11$, $l_{min} = 42$) and all of the prediction variables. The model was then applied to the test data set yielding the results displayed in Table 4.1. The corresponding plots can be found in figures 4.4 and 4.5.

Features	Data set	RMSE	R	Computation time (s)	% 3 mins	% 5 mins	Median
All	Test	2.024	0.875	3.49	91.14	97.90	8.48

Table 4.1: Results RT for normal TXOT using all prediction variables

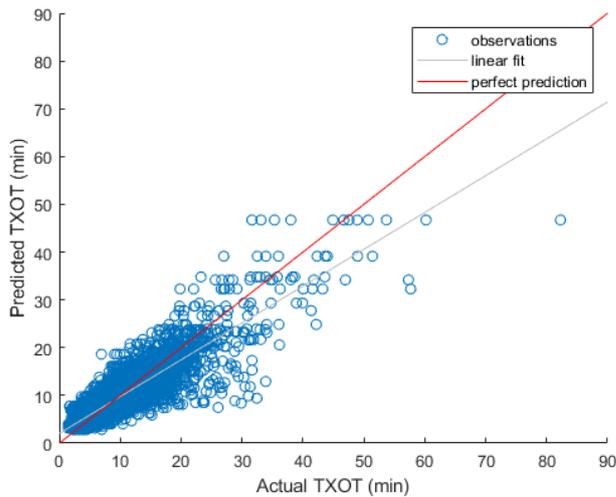


Figure 4.4: Regression plot for testing data set using all features

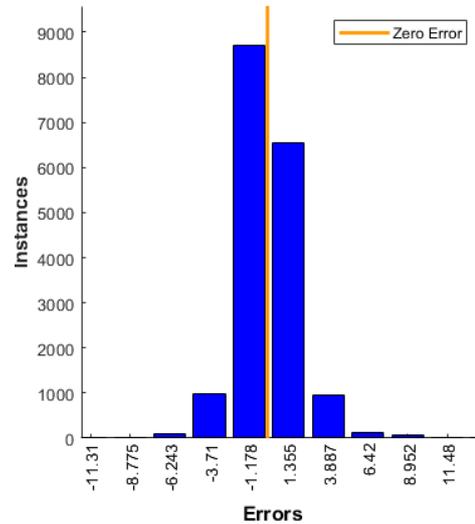


Figure 4.5: Error histogram for tree trained with all features

4.1.4. Top feature extraction

In [section 3.4](#), the RReliefF method for determining the most important features for predicting TXOT was discussed. This feature selection technique is necessary to determine the top features for ML techniques such as NNLM as there is no way of extracting the utilised features from the network. For RTs, however, this is not the case as the features used in a prediction can be viewed in the tree. Given that the features used in the tree are those that result in the most accurate predictions, it can be deduced that these are also the most influential factors on TXOT. As such, a new set of top features is directly extracted from the tree. This is done using the ‘predictorImportance’ function in MATLAB. This function computes estimates of predictor importance for the trained tree by summing changes in the mean squared error due to splits on every predictor and dividing the sum by the number of branches.

The function outputs a row vector with the same number of elements as the number of predictors (columns) in the trained tree. The entries are the estimates of predictor importance, with 0 representing the smallest possible number. The function was performed on the tree resulting in the top 11 features which have been visualised in [Figure 4.6](#).

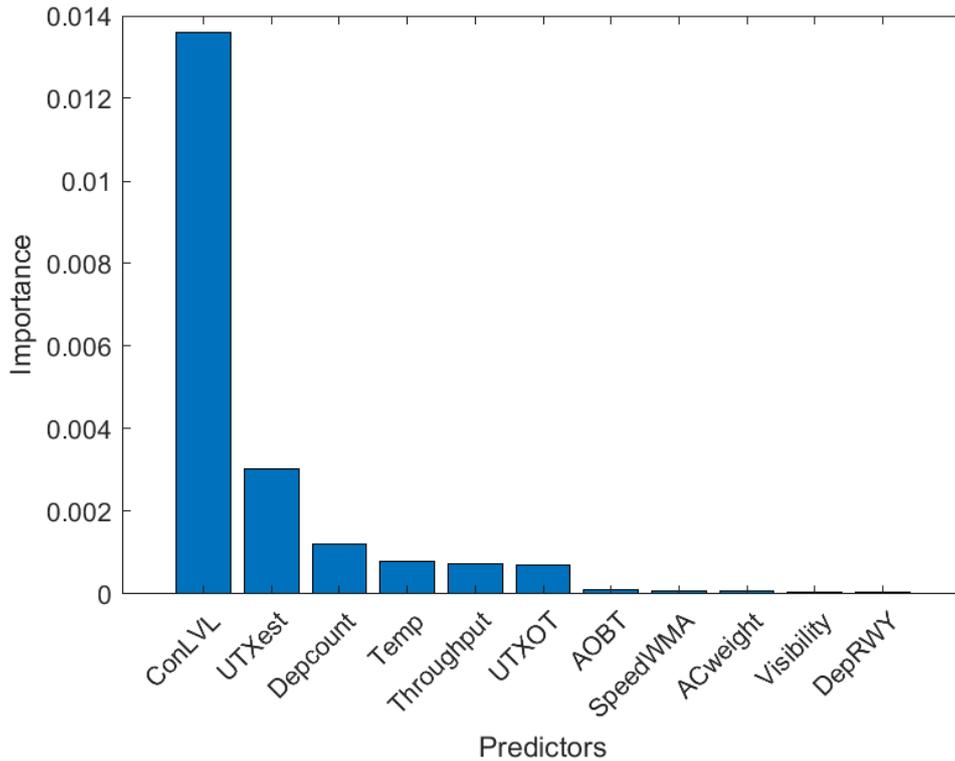


Figure 4.6: Top features extracted from trained tree

An advantage of using the `predictorImportance` function over the `RreliefF` algorithm is that the `RreliefF` algorithm does not account for the problem of “Multicollinearity”. Multicollinearity occurs when one prediction variable in a multiple regression model can be linearly predicted from the others with a high degree of accuracy i.e. the variables are highly correlated. Multicollinearity is a problem because it undermines the statistical significance of an independent variable which can lead to skewed or misleading results. When determining the top features, the `RreliefF` algorithm will give certain features a high importance regardless of their correlation, thus minimizing the importance of other potentially important features.

Decision trees are immune to multicollinearity by nature. When they decide to split, the tree will choose only one of the perfectly correlated features. Since the `predictorImportance` extracts the top variables directly from the tree, this problem is automatically overcome. As such, non-correlated variables, are given a higher importance.

4.1.5. Leaf size - top features

Once the top features had been extracted from the tree, a new tree was trained using only the top variables. Since only the top variables were used, the parameters d_{max} and l_{min} had to be re-evaluated. The methodology used to determine these parameters in sections 4.1.1 and 4.1.2 is repeated for the new tree. To determine leaf size, cross-validated error was once again used, as shown in Figure 4.7.

A similar figure to that of Figure 4.2 is obtained. As such, the same reasoning is applied to select the leafsize. From the figure it can be seen that the minimum cross-validated error occurs in the minimum leaf size range of 20-35. however, that this is the performance error on the training set. In

4.1. NORMAL TXOT ANALYSIS

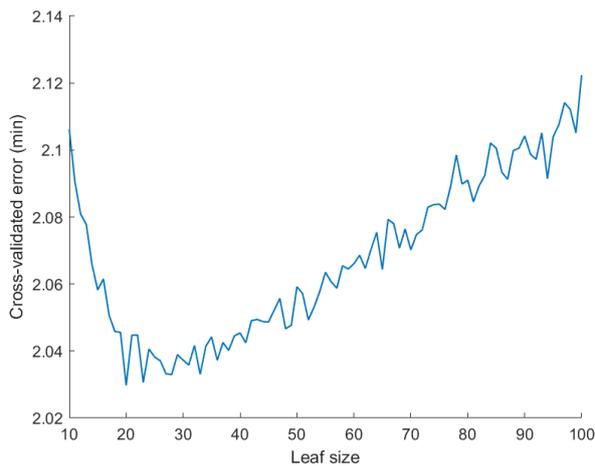


Figure 4.7: Cross-validated error for varying minimum leafsize l_{min} using top features only

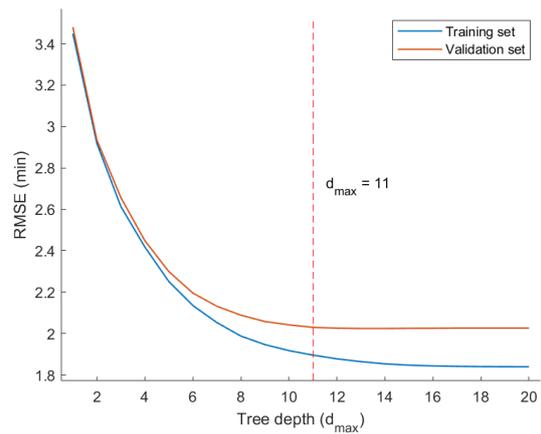


Figure 4.8: RMSE vs. tree depth for training data set and validation data set using top features only

order to avoid over-fitting, and to ensure there are sufficient data points in a to fit a distribution, a slightly larger leaf size should be chosen. Therefore, a leaf size of 42 was once again chosen.

4.1.6. Tree depth - top features

The maximum tree depth is also re-evaluated, where the RMSE is once again used. The corresponding figure is shown in [Figure 4.8](#).

Once again, a tree depth of 11 was chosen as the predictions for the validation set do not improve past this point. Additionally, the deeper the tree becomes, the more difficult it is to interpret. A comparison between the validation data set results of a tree trained with all features and a tree trained with only the top variables can be found in [Figure 4.9](#).

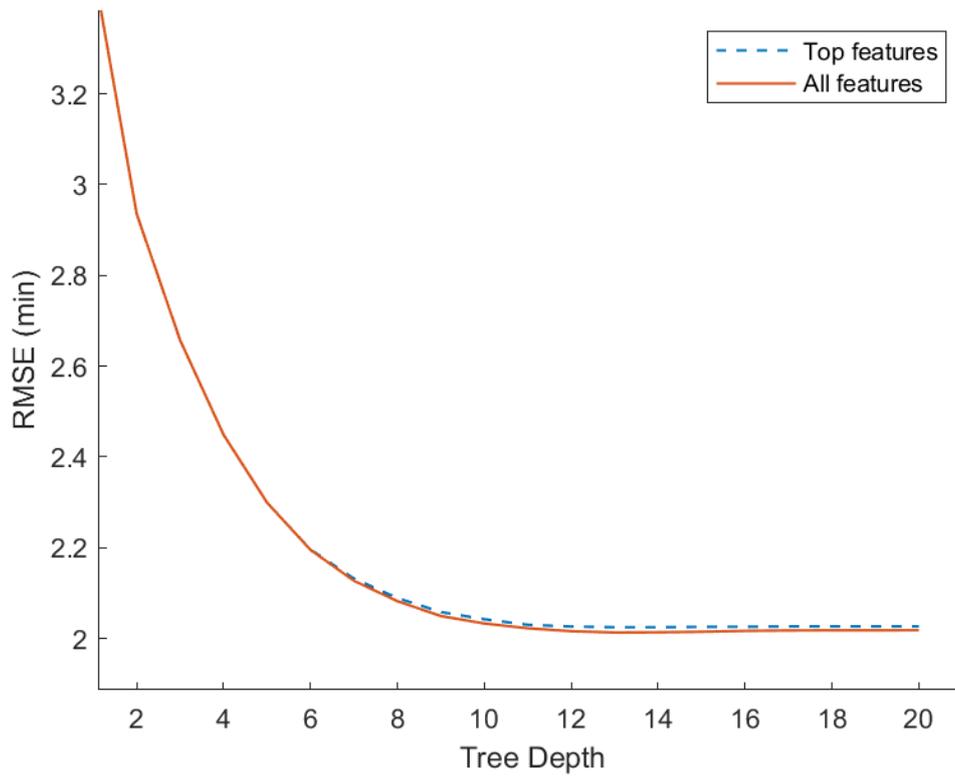


Figure 4.9: RMSE vs. tree depth for validation data sets using top features and all features

As can be seen there is an almost negligible difference in RMSE between the two trees, confirming the effectiveness of using top features only.

4.1.7. Final tree results

A final tree was trained using a minimum leaf size of 42, a maximum tree depth of 11, and only the top variables specified in [subsection 4.1.4](#). The results of this tree when applied to the test data set are shown in [Table 4.2](#). The corresponding regression and error histogram can be seen in [figures 4.10](#) and [4.11](#). Additionally, the PDFs and CDFs of the predictions vs the actual TXOT are presented in [figures 4.12](#) and [4.13](#), respectively.

Features	Data set	RMSE	R	Computation time (s)	% 3 mins	% 5 mins	Median
Top	Test	2.026	0.875	3.14	91.32	97.88	8.42

Table 4.2: Results RT for normal TXOT using only top prediction variables

4.1. NORMAL TXOT ANALYSIS

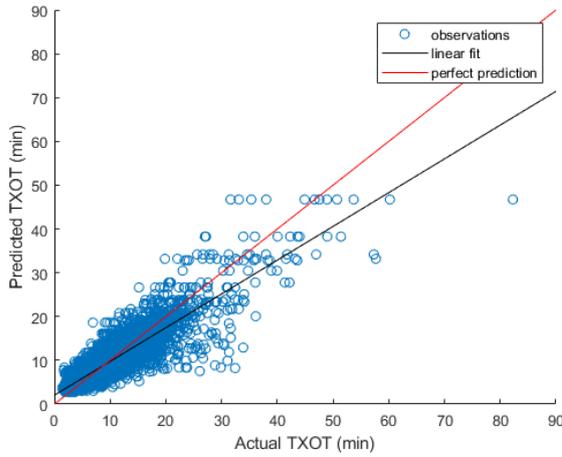


Figure 4.10: Regression plot for final tree using top features only

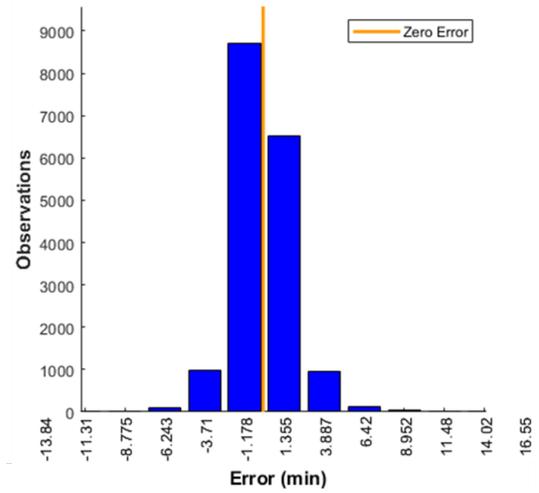


Figure 4.11: Error histogram for final tree using top features only

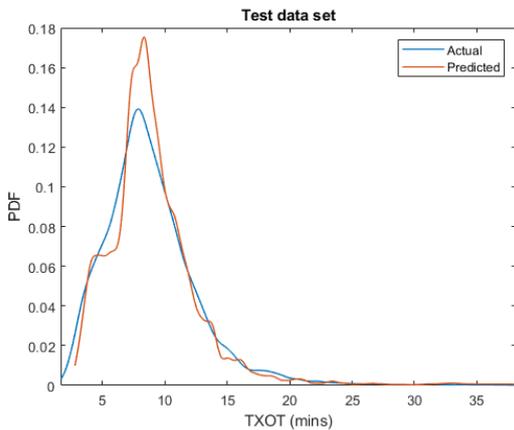


Figure 4.12: PDF of actual TXOT vs predicted TXOT

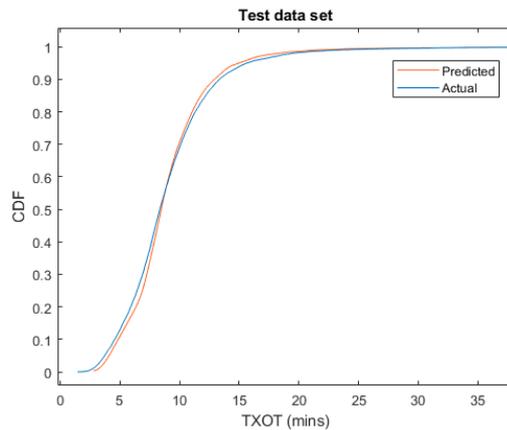


Figure 4.13: CDF of actual TXOT vs predicted TXOT

From figures 4.12 and 4.13, it can be observed that for low actual TXOTs (< 7.5 mins) the tree tends to overpredict the TXOT. Similarly, for actual TXOTs ranging from 7.5 - 10 mins, the tree tends to underpredict the TXOT. In general, however, the tree provides a very good fit for the majority of predictions, as observed in Figure 4.13.

Compared to the trees trained in sections 4.1 and 3.7.2, this tree has a marginal decrease in predictive performance on the test set. However, the advantages of this tree far outweigh this decrease in performance. A comparison of these trees is displayed in Table 4.3.

Parameter settings	l_{min}	d_{max}	Features	RMSE	R	Computation time (s)	% 3 mins	% 5 mins	Median
Default	36	-	All	1.80	0.902	5.8	93.43	98.47	8.44
Default	36	-	Top (RRF)	1.84	0.897	5.1	93.17	98.25	8.46
Custom	42	11	All	2.024	0.875	3.49	91.14	97.90	8.48
Custom	42	11	Top (RT)	2.026	0.875	3.14	91.32	97.88	8.42

Table 4.3: RT result comparison

The final tree is trained the most quickly, and will be the most robust when making predictions based on new data due to these use of only the top prediction variables. Regarding the parameter l_{min} , the final tree trained has a larger minimum leaf size, meaning that better distributions can be fit. Regarding d_{max} , a smaller maximum tree depth makes the tree much easier to interpret. Additionally, limiting the tree depth significantly reduces the number of terminal nodes (leaves). This means that while the minimum leaf size is set to 42, there are more leaves which consist of more than 42 data points. The more data points in a leaf, the more reliable the fit of the distribution. The fitted distributions per leaf are presented and discussed in the following section, [section 4.2](#).

4.2. Distribution fitting

After training a tree, a mean and distribution can be extracted per decision node. This is needed to observe precursors and understand the result of the TXOT prediction. The final tree trained in [subsection 4.1.7](#) produced 668 leaf nodes, for which a distribution needs to be fit to each leaf.

To fit a distribution to a leaf, the `fitdist()` function in MATLAB® was used. The following distributions were tested as contenders for fitting the leaves: ‘Birnbbaum-Saunders’, ‘Exponential’, ‘Gamma’, ‘Generalized Extreme Value’ (or Gumbel), ‘Half-Normal’, ‘Logistic’, ‘Lognormal’, ‘Normal’, ‘Rayleigh’, ‘tLocationScale’, and ‘Weibull’. For each of the proposed distributions, the `fitdist()` function finds the best fit per leaf using the maximum likelihood estimation. A goodness-of-fit test was then performed per distribution to test if the proposed distribution on the leaf is suitable.

4.2.1. Kolmogorov-Smirnov test

The goodness-of-fit test chosen is a one-sample Kolmogorov-Smirnov test. This is performed using the `kstest()` function in MATLAB. The Kolmogorov-Smirnov test measures the maximum vertical distance, D between the empirical CDF and the parametric CDF and compares this distance to a critical value. A visual representation of the Kolmogorov-Smirnov test can be seen in [Figure 4.14](#).

The test returns a test decision for the null hypothesis that the empirical comes from a parametric distribution, against the alternative that it does not come from such a distribution. The result h is 1 if the test rejects the null hypothesis at the 5% significance level, or 0 otherwise. Note that the `kstest()` algorithm does not actually compare the distance, D , to the critical value, since the critical value is also an estimate. Instead, it compares the p -value to the significance level α . p is the probability, given the null hypothesis, of observing a test statistic as extreme as, or more extreme than, the observed value. Small values of p question the validity of the null hypothesis. Thus, when the p -value is lower than the significance level, the null hypothesis is rejected.

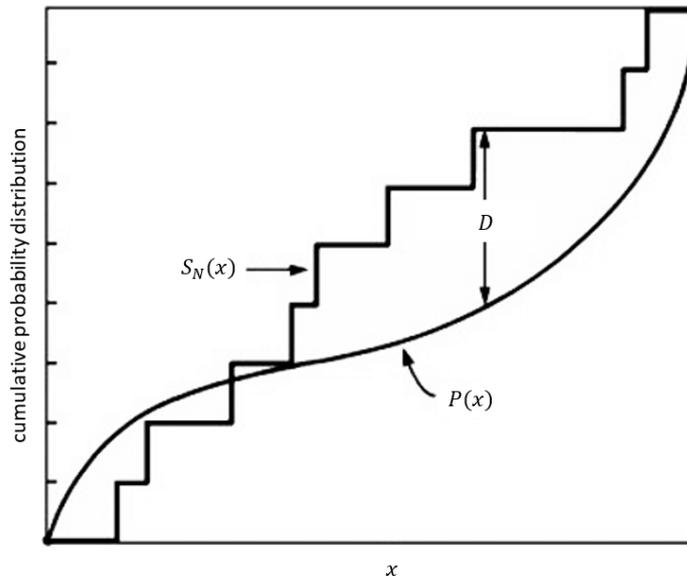


Figure 4.14: Kolmogorov-Smirnov test

An example leaf, node 1099, is chosen to show this process. In Figure 4.15, the cumulative distribution function (CDF) for the empirical data as well as the proposed distributions for that leaf is shown. The $kstest()$ is applied, the results of which can be seen in Table 4.4. The table suggests that the null hypothesis is rejected for the ‘Exponential’, ‘Rayleigh’, and ‘HalfNormal’ distributions, and thus these distributions are excluded from further testing. From Figure 4.15 it is also clear that these 3 distributions do not fit. A plot of the CDFs of the remaining distributions can be seen in Figure 4.16. The corresponding probability density function plot is shown in Figure 4.17, where the actual shapes of the distributions can be seen.

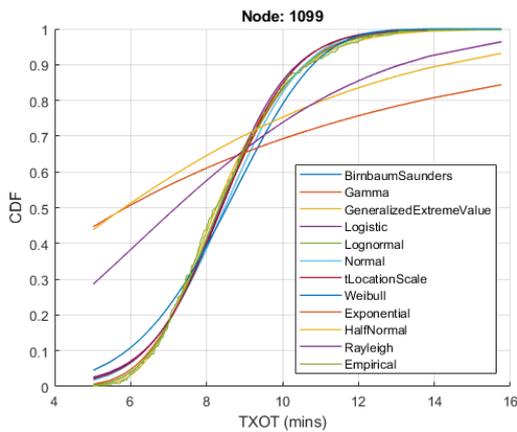


Figure 4.15: CDFs of empirical data and possible distributions for example node 1099

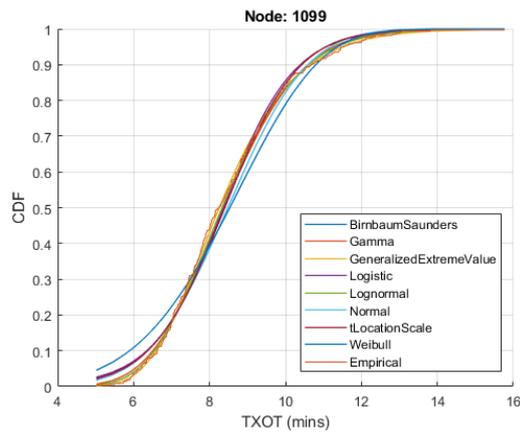


Figure 4.16: CDFs of empirical data and accepted distributions for example node 1099

4.2. DISTRIBUTION FITTING

Distribution	Test result (h)	Distribution	Test result (h)
Birnbaum-Saunders	0	Lognormal	0
Exponential	1	Normal	0
Gamma	0	Rayleigh	1
Generalized Extreme Value (Gumbel)	0	tLocationScale	0
HalfNormal	1	Weibull	0
Logistic	0		

Table 4.4: Results of Kolmogorov-Smirnov test for example node 109

After having rejected the bad fits, the best of the remaining distributions needs to be chosen. This is done using the variance of the distributions, where the distribution with the lowest variance for that leaf is the best. As seen in Table 4.5, the best fit for this example is the Gamma distribution. The Gamma distribution on the example leaf is show in Figure 4.18. From inspection of the figure it is also clear that it is a good fit.

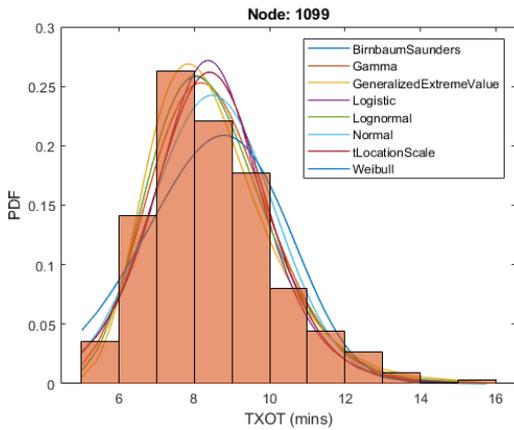


Figure 4.17: Histogram of TXOTs and proposed distributions for example node 1099

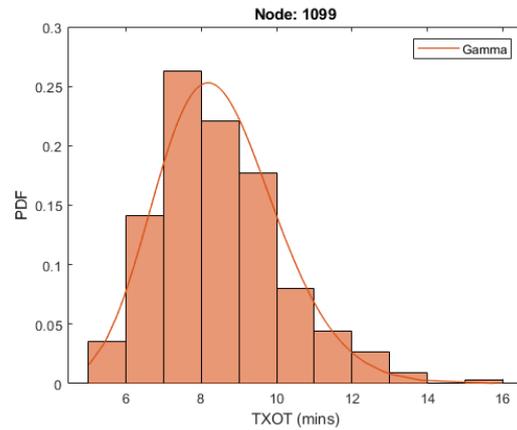


Figure 4.18: Histogram of TXOTs fitted with a Gamma distribution for example node 1099

Distribution	Variance	Distribution	Variance
Birnbaum-Saunders	2.58	Lognormal	2.59
Gamma	2.56	Normal	2.71
Generalized Extreme Value (Gumbel)	2.70	tLocationScale	2.69
Logistic	2.78	Weibull	3.60

Table 4.5: Variance of distributions for example node 1099

The process was repeated for all of the leaves, and the best distribution for each of the leaves was determined. The best distribution overall is the distribution which provides the best fit for most leaves. In Table 4.6, the percentage of best fits per distribution is displayed.

Distribution	% Best fits	Distribution	% Best fits
Birnbaum-Saunders	10.63	Lognormal	1.50
Gamma	27.10	Normal	0.45
Generalized Extreme Value (Gumbel)	24.55	tLocationScale	10.93
Logistic	20.96	Weibull	3.89

Table 4.6: % Best fits per distribution

As can be seen, the ‘Gamma’ distribution provides the highest percentage of best fits over the leaves. The Gamma distribution is therefore fitted to all 668 terminal nodes, a selection of which can be seen in [Figure 4.19](#). The equation of the Gamma distribution is shown in [Equation 4.1](#).

$$f(x) = \frac{\left(\frac{x-\mu}{\beta}\right)^{\gamma-1} \exp\left(-\frac{x-\mu}{\beta}\right)}{\beta\Gamma(\gamma)} \tag{4.1}$$

for $x \geq \mu; \gamma, \beta > 0$.

Here, γ is the shape parameter, μ is the location parameter, β is the scale parameter, and Γ is the gamma function which has the following formula:

$$\Gamma(a) = \int_0^\infty t^{a-1} e^{-t} dt \tag{4.2}$$

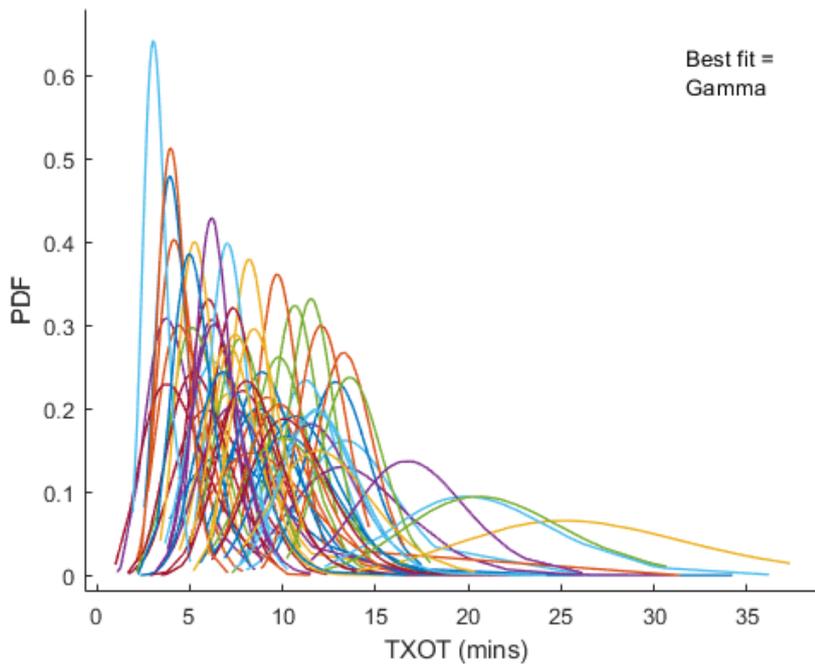


Figure 4.19: Gamma distributions for 60 of the 668 leaf nodes

In general it can be observed that as the TXOT increases, the distribution becomes more spread out, and the probability of the TXOT being predicted accurately is lower. It also indicates that the

uncertainties are higher for flights with longer TXOTs. Conversely, leaves with a lower mean TXOT tend to have much less spread out distributions. This means if a flight matches the conditions of the leaf, there is a higher probability of the TXOT prediction being accurate. Additionally, more confidence can be put in a prediction if the leaf contains more points.

4.3. Top feature analysis

In [Figure 4.6](#), the features which influence TXOTs most significantly according to the regression tree were presented. However, this merely presents what they are and not what effect they have on TXOT. Therefore, an analysis is performed on these top features to observe their influences on TXOT.

1. Congestion level (ConLVL)

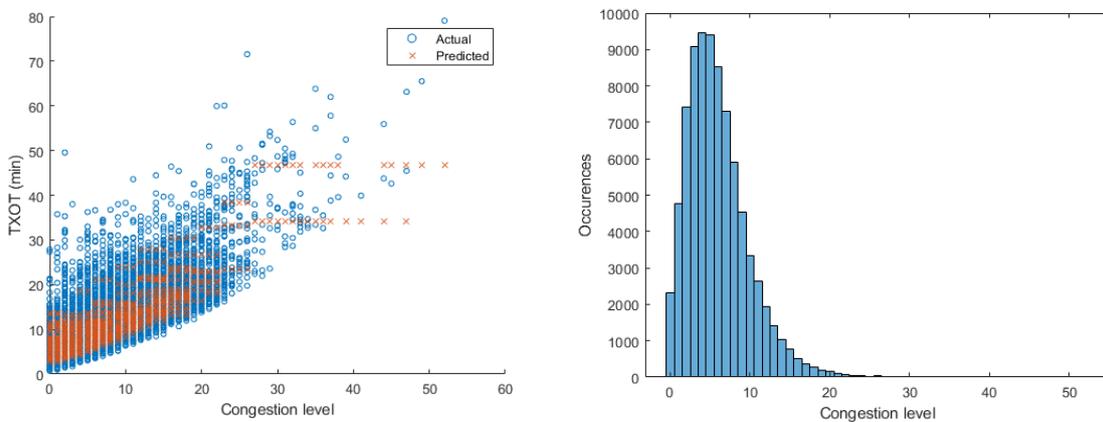


Figure 4.20: Congestion level vs. TXOT and Congestion level histogram

The first, and by far the most influential feature to be analysed is the congestion level. In [Figure 4.20](#), the congestion level is plotted vs both the actual TXOT and predicted TXOT of the flights. A histogram of how often certain congestion levels occur is also displayed. From the regression there is a clear relationship between congestion level and TXOT, namely, a higher congestion level results in a longer TXOT. This is logical as aircraft will spend longer queuing if there are more aircraft in the queueing system. This is also in accordance with the analysis performed in [subsection 2.2.1](#). Given the strong correlation, it is clear why this variable is the top feature.

2. 1st Estimation of unimpeded taxi-out time (UTXest)

From the regression plot in [Figure 4.21](#), there is a clear linear relationship between the 1st estimation of the UTXOT and the TXOT. In general, as the 1st estimation UTXOT increases, the TXOT also increases. This is logical as a longer UTXOT is associated with a longer taxi distance. Obviously the further an aircraft has to taxi, the longer its TXOT. While there appears to be a spike in longer TXOTs at around 7 minutes, this is due to there being a much larger number of flights with a 1st estimation UTXOT of 7 mins. This means there is a higher chance of TXOTs being delayed due to other external factors.

4.3. TOP FEATURE ANALYSIS

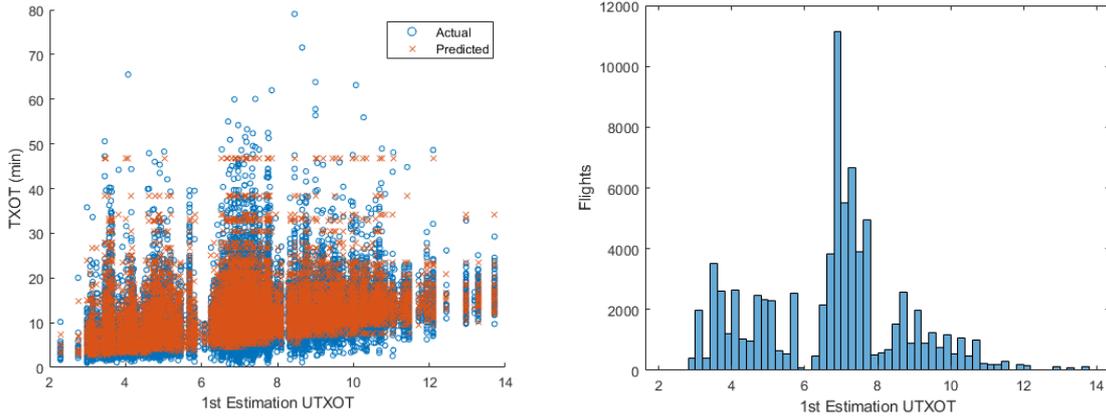


Figure 4.21: 1st Estimation UTXOT vs. TXOT and histogram of 1st estimation UTXOT

3. Departures in the last 20 minutes (Depcount)

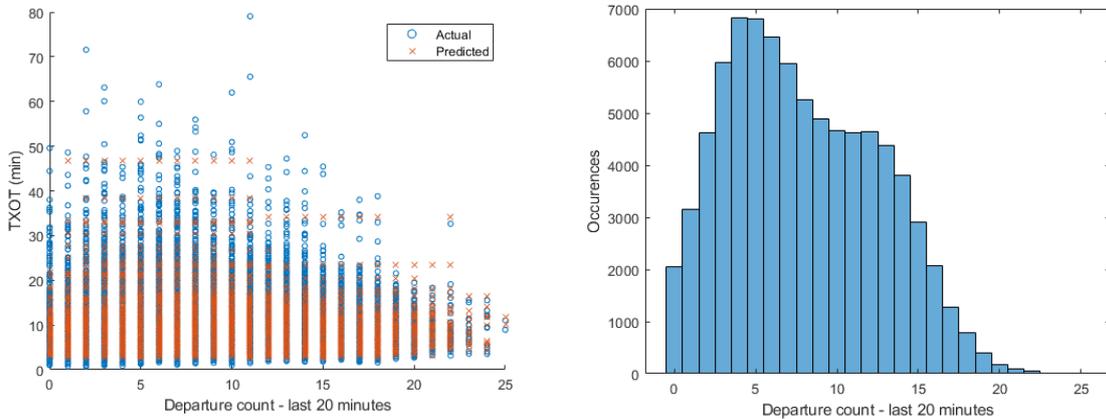


Figure 4.22: Departures in last 20 mins vs. TXOT and histogram of departures in last 20 mins

From [Figure 4.22](#), it is not immediately clear how the number of departures in the last 20 minutes relates to the TXOT. While it would be expected that as the departure count increases the TXOT also increases due to the number of aircraft in the system, this is not the case. A possible explanation for this is that if the departure count is high, it means operations are running very smoothly and thus the aircraft in the system spend little time queueing and have reduced TXOTs. Conversely, if the departure count is lower, operations may be running less smoothly and therefore TXOTs may be larger. This would explain the apparent increased TXOTs occurring for aircraft with a departure count of between 5-10. Of course, a large portion of flights in this range also have normal TXOTs which is possible if the airport is not busy at the time.

4. Temperature (Temp)

While above 3 degrees, the temperature has little to no effect on the TXOT. However, for departure flights with a temperature of less than 3°, there is a clear increase in TXOT. This may be due to numerous factors such as snow or ice on the taxi/run-ways, in which case the aircraft

4.3. TOP FEATURE ANALYSIS

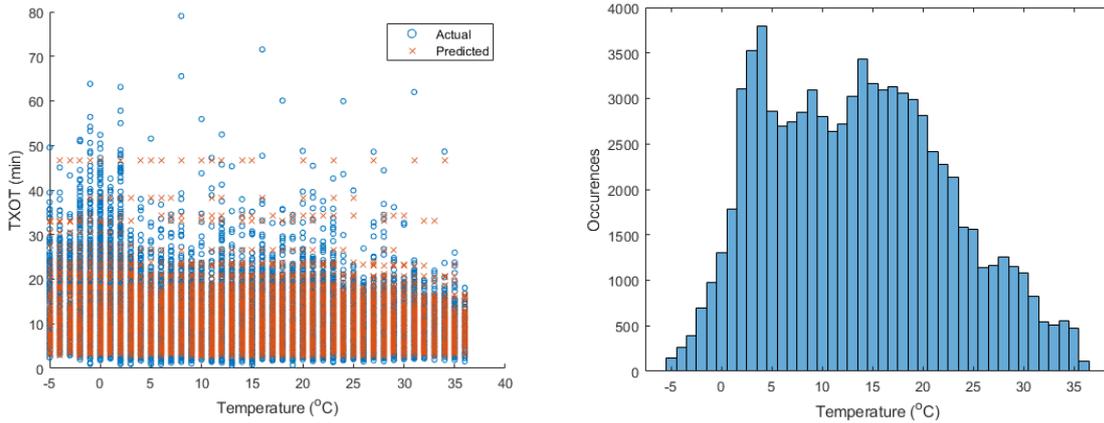


Figure 4.23: Temperature vs. TXOT and temperature histogram

must taxi slower, or due to the aircraft requiring de-icing, in which case the aircraft must wait to be de-iced. Given many aircraft requiring de-icing, large queues can form while waiting for de-icing, as discussed in [Figure 3.37](#).

5. Airport throughput in the last hour (Throughput)

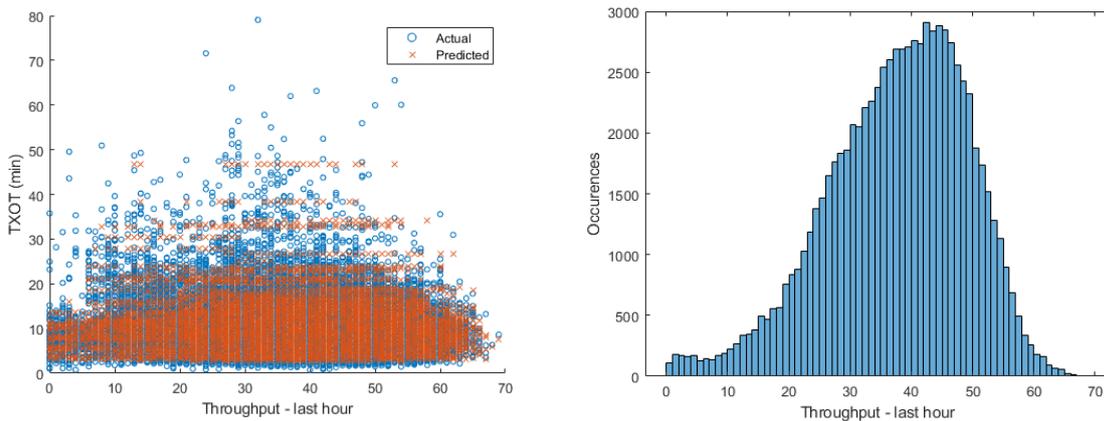


Figure 4.24: Throughput in last hour vs. TXOT and histogram of throughput in last hour

Similar to the departure count, there is not an immediately apparent trend of the airport throughput. As would be expected, at higher levels of throughput, higher TXOTs would be expected due to more aircraft being in the system, and thus more queue time. However, this is not the case. At very high levels of throughput, the TXOTs are smaller. This is likely due to the operations running very smoothly at this moment in time. Conversely, given lower airport throughputs i.e. between 25-50 movements in the last hour, more longer TXOTs occur. In this period, operations may be running less smoothly, causing congestion and longer TXOTs. At low throughput levels, the airport is not very busy, and the TXOTs are therefore lower in general.

6. Unimpeded taxi-out time (UTXOT)

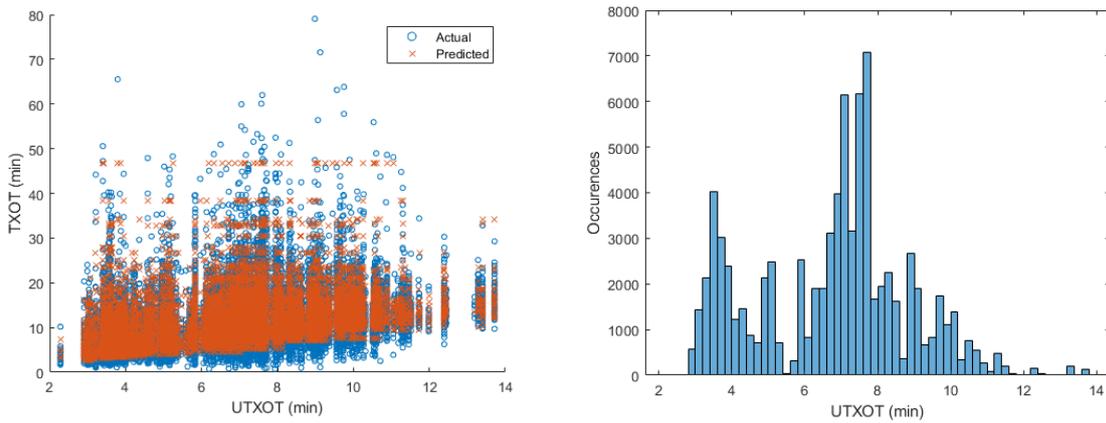


Figure 4.25: UTXOT vs. TXOT and UTXOT histogram

The UTXOT (or 2nd estimation UTXOT), behaves very similarly to that of the 1st estimation of the UTXOT. However, providing 2 separately calculated measures of the UTXOT can help increase the prediction accuracy of the model. As previously mentioned, the UTXOT relates to the distance of the gate to the runway. A larger distance, and therefore UTXOT, will of course result in a longer TXOT which is captured in the regression.

7. Actual off-block time (AOBT)

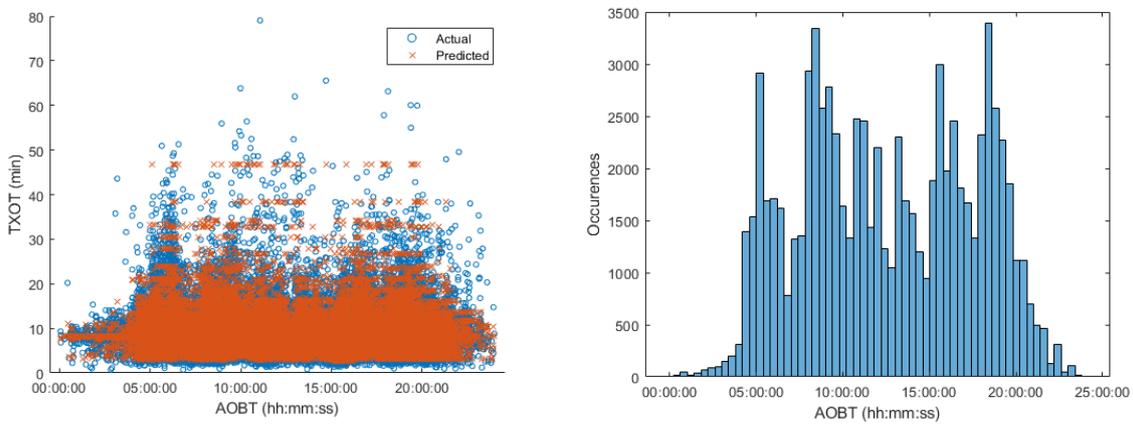


Figure 4.26: AOBT vs. TXOT and AOBT histogram

In [Figure 4.26](#), the AOBT regression and AOBT histogram is displayed. From the histogram, it is clear that there are certain times of the day in which more AOBTs occur. The peaks in the histogram also tend to correspond with the peaks in the regression plot. This makes sense since if many aircraft are pushed back in the same time-interval, the congestion level will be higher. Based on the regression plot in [Figure 4.20](#), the higher the congestion level, the higher the TXOT. Therefore, if an aircraft has an AOBT during one of the ‘peaks’ in AOBT histogram, it is more likely to have a longer TXOT.

8. Wind speed (SpeedWMA)

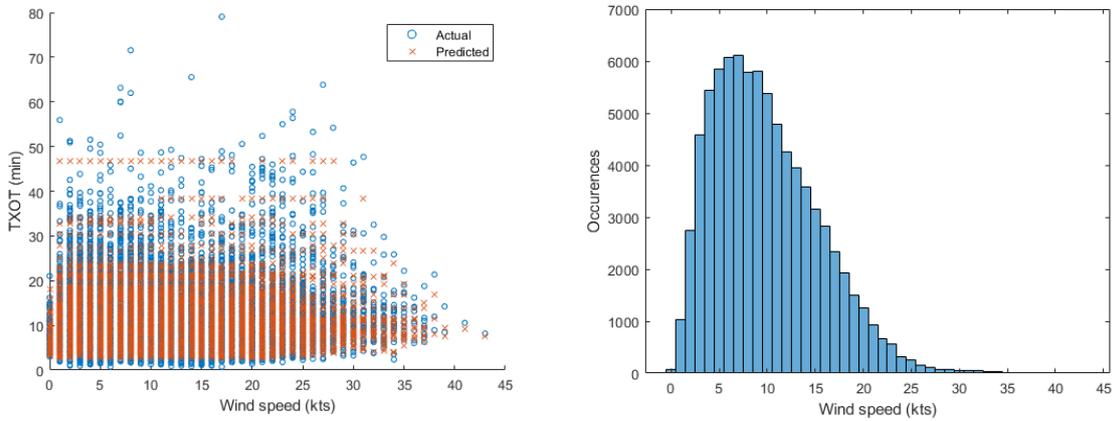


Figure 4.27: Wind speed vs. TXOT and wind speed histogram

Based on the regression plot in Figure 4.27, up until roughly 20 kts, there appears to be very little correlation between wind speed and TXOT. Past 20 kts, however, a very slight positive linear trend can be observed, that is, if the wind speed increases, the TXOT also increases. These additional TXOTs may be induced by either the pilot or an ATCO deciding to delay take-off due to crosswinds. While wind speeds of around 30-35 kts are normally required to influence take-offs and landings, flights may begin to experience delay due to wind speeds after 20 kts since the wind speed recorded is an average. Therefore, while the average may be 20 kts, gusts of 30-35 kts may in fact occur causing delay. Additionally, high wind speeds are often linked to storms which may cause further delays.

9. Aircraft maximum take-off weight (ACWeight)

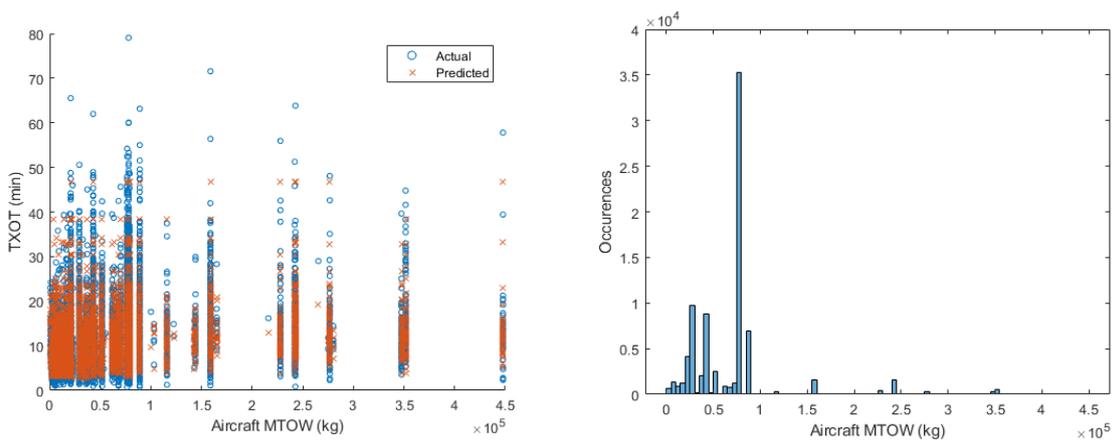


Figure 4.28: Aircraft weight (MTOW) vs. TXOT and histogram of aircraft weight

The final feature to be analysed is the aircraft weight or MTOW. From the regression plot in Figure 4.28, a slight positive linear relationship can be observed namely, a heavier aircraft tends

4.3. TOP FEATURE ANALYSIS

to have a longer TXOT. This is due to the fact that movements such as pushback and turning corners are generally performed more slowly for larger aircraft. Larger aircraft also require a longer take-off run, and while this contribution is minimal, does result in a longer TXOT.

5

Results extended TXOT

While several studies have been performed investigating different methods for normal TXOT prediction, little has been done regarding the prediction of abnormal, or extended TXOT prediction. As discussed in [section 3.9](#), flight TXOTs are considered to be extended if they are more than 2 standard deviations from the mean, or have a TXOT greater than 17.3 mins. Based on the data provided by Vienna Airport, this meant that 4050 flights in 2015 could be considered as having ‘extended’ TXOTs. To help visualise this, a histogram of all the flight TXOTs and the ‘extended’ TXOT threshold is shown in [Figure 5.1](#). Additionally, a scatter plot of only the extended TXOT flights is shown in [Figure 5.2](#).

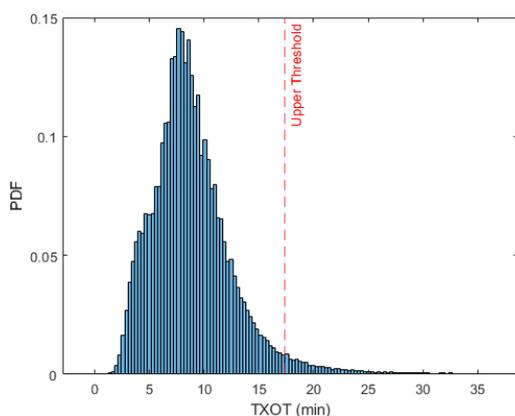


Figure 5.1: Histogram of all 2015 flight TXOTs

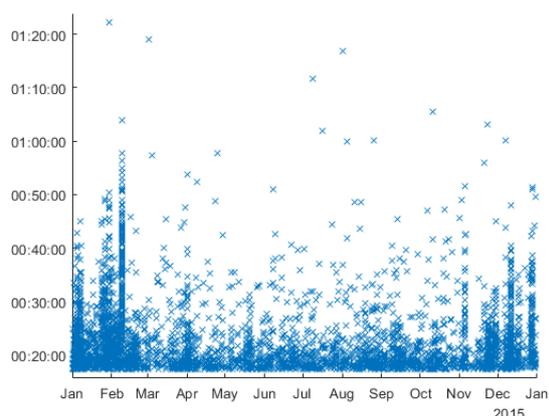


Figure 5.2: Extended flight TXOTs

5.1. Regression tree setup

The same methodology used in the creation of the regression in [chapter 4](#) was used to create the regression tree for predicting ‘extended’ TXOTs and observing the most influential features. As with the previous tree, the first step is to determine the parameters for the tree, namely, minimum leaf size l_{min} and maximum tree depth d_{max} . Once again, the leaf size is determined using cross-validation and can be observed in [Figure 5.3](#).

5.1. REGRESSION TREE SETUP

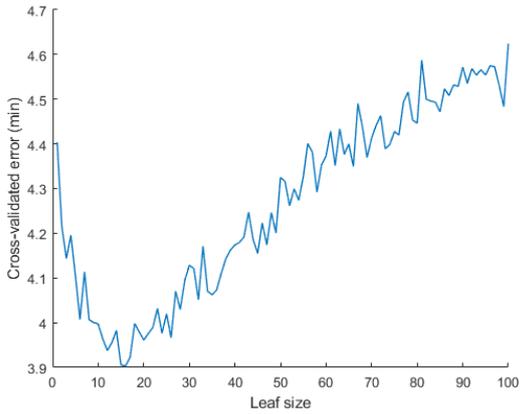


Figure 5.3: Leaf size vs. cross-validated error for

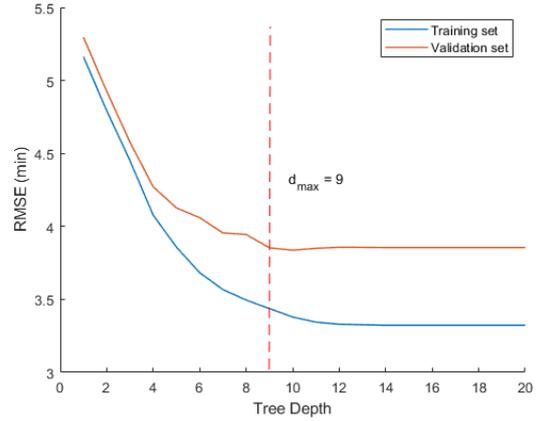


Figure 5.4: Caption

To prevent over-fitting and to ensure there are enough points in a leaf to fit a distribution, the leafsize is set slightly higher than that which achieve the minimum crossvalidated error. A leafsize of 20 is therefore chosen.

Next, as shown in [Figure 5.4](#), the treedepth is determined using the RMSE. To ensure the tree is interpretable and to avoid overfitting. A tree depth less than what results in the minimum RMSE on the training set should be chosen. By comparing the RMSE of the validation set to the training set, it can be seen that the validation RMSE does not improve past a tree depth of 9. A tree depth of 9 is therefore selected.

A tree is then trained using all the prediction variables and the selected parameters ($l_{min} = 20$, $d_{max} = 9$), the results of which can be seen in [Table 5.1](#) and figures [5.5](#) and [5.6](#)

Features	Data Set	RMSE	R	% 3 mins	% 5 mins	% 10
All	Test	3.75	0.789	68.21	84.93	97.52

Table 5.1: Results RT for extended TXOT using all prediction variables

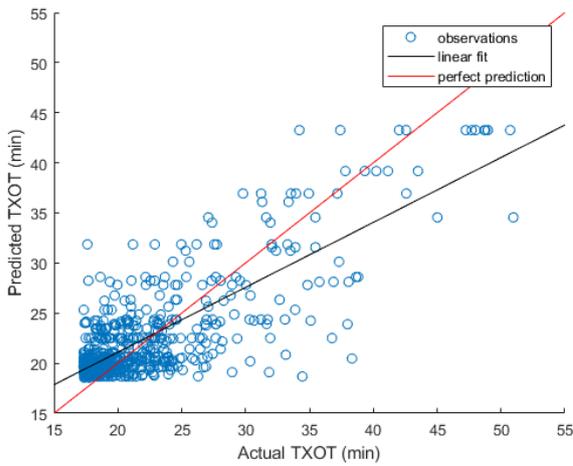


Figure 5.5: Regression plot for testing data set using all features

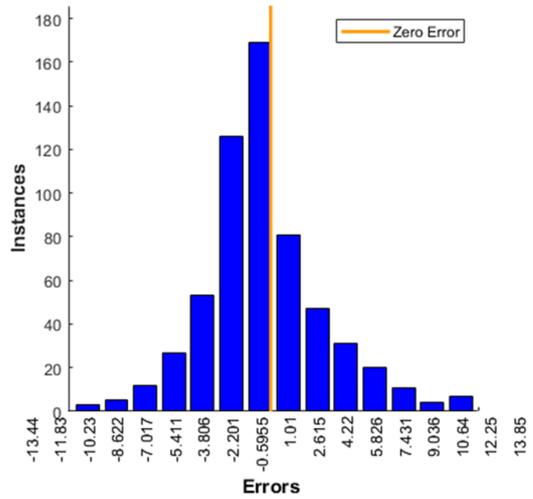


Figure 5.6: Error histogram for tree trained with all features

5.2. Top features

Having trained the tree with all the variables, the top variables could then be extracted from it. The `predictorImportance` function in MATLAB was once again used. The most importance features that influence extended TXOTs are shown in Figure 5.7.

5.3. Final tree

Having extracted the top features from the previous tree, a new tree was trained using only the top features. Given the new features, new parameters values l_{min} and d_{max} had to be determined. This was once again done using the same method as previously discussed. Almost identical plots to those in figures 5.3 and 5.4 were obtained, and thus the parameter values remained the same namely, $l_{min} = 20$, and $d_{max} = 9$. The results of the tree can be found in Table 5.2. The corresponding regression plot and error histogram are presented in figures 5.8 and 5.9 respectively. Additionally, the PDFs and CDFs of the predicted and actual extended TXOTs are shown in figures 5.10 and 5.11 respectively. Finally, in Figure 5.12, the final trained tree and its what-if statements are shown. If the statement ($<$) is true, the downward path is followed; if the statement is false, the upward path is followed.

Features	Data Set	RMSE	R	% 3 mins	% 5 mins	% 10
Top	Test	3.75	0.790	68.38	84.60	97.35

Table 5.2: Results RT trained with top features only

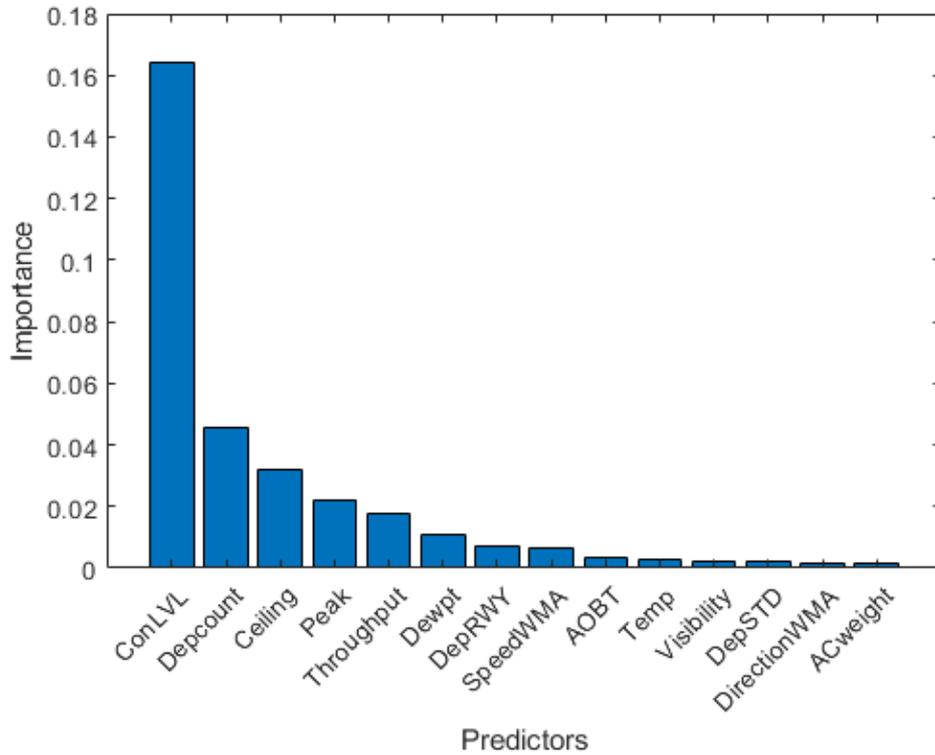


Figure 5.7: Top extended TXOT features

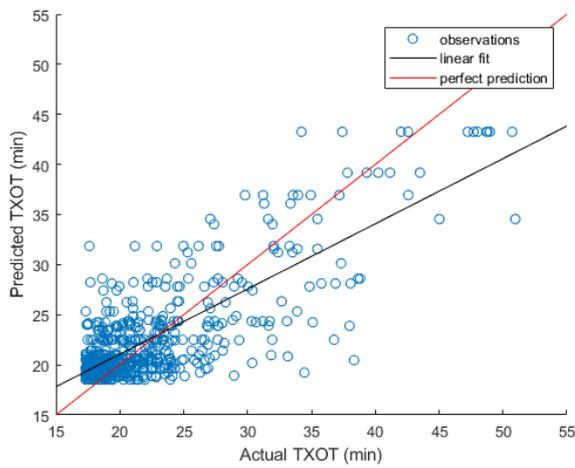


Figure 5.8: Regression plot for testing data set using top features only

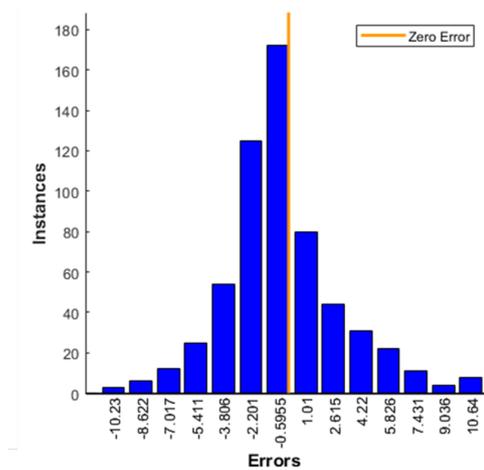


Figure 5.9: Error histogram for tree trained with top features only

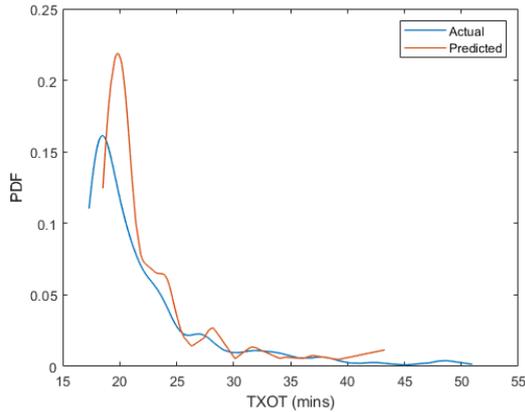


Figure 5.10: PDF of actual TXOT vs predicted TXOT

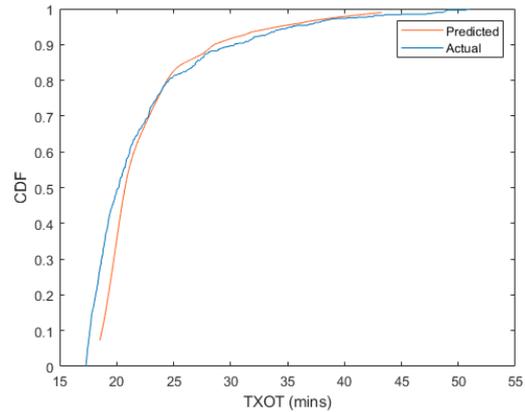


Figure 5.11: CDF of actual TXOT vs predicted TXOT

5.3.1. Example flight

In order to help visualise how a prediction is actually made by the tree, an example of the decisions made for an example flight is shown in [Figure 5.12](#). Due to confidentiality, this flight will be referred to as flight ‘X’. This flight was chosen at random. In [Table 5.3](#) dummy data for flight ‘X’ is presented. The highlighted cells are those that were used in order to make the TXOT prediction and correspond to the highlighted path in [Figure 5.12](#). The flight has a TXOT of 00:24:14 mins, almost 20 minutes longer than the UTXOT of the flight. To determine the cause of this delay, an analysis of the RT and the conditions is performed.

Starting at the root node, if the split condition is true (i.e. congestion level < 22.5) the downward path to the next node is followed. Conversely, if the split condition is false (i.e. congestion level \geq 22.5), the upward path is followed. In this case, the flight has an expected congestion level of 18. While this is not considered peak congestion, there are enough aircraft present in the system for which the aircraft is required to queue. The next condition is a cloud ceiling height of less than 3050ft. In this case, the cloud height is 100ft, significantly less than the 3050ft threshold. This low cloud ceiling, paired with restricted visibility (250m) and low temperature (-2°), suggests it may be snowing. The next condition is a dew point less than 1.5° . In this case, the dew point is -3° , and thus the condition is met. As discussed in [section 5.5](#), a negative dew point suggests freezing temperatures. Additionally, a dew point close to the actual temperature suggests the formation of dew or, when these temperatures are negative (which in this case is true), snow and ice. All of this data points towards the likelihood of snow, a reason for the delay of the flight. Additionally, given snow and ice, it is likely that the aircraft needs to be de-iced, another reason for its delay.

Next, the throughput and departure count conditions are analysed (nodes at tree depth 4 and 7, respectively). While the congestion level remains relatively high, the departure count in the last 20 minutes and the throughput in the last hour are relatively low (6 and 39, respectively). This indicates that while there are plenty of aircraft in the system, the rate of arrivals and departures are low, suggesting that the surrounding aircraft are also experience delayed taxi-outs. Finally, both wind speed conditions (<26.5 kts and <10.5 kts) are met, with the average wind speed being 4kts. This low wind speed indicates that the current cloud conditions will not change quickly. If it is in fact

5.4. DISTRIBUTIONS

snowing, it is likely to remain so for a while.

Variable	Value	Variable	Value	Variable	Value
Date	10/12/15	Con. lvl	18	Temperature (°C)	-2
Flt. no	'X'	Throughput	39	Dew point (°C)	-3
AC Type	A320	Dep. count	6	Hour	9
ICAO Cat	M	U1	5.017	Day week	5
MTOW (kg)	78000	U1 Group	5.083	Day month	10
Origin	LOWW	Sat. lvl	4.264	Day year	344
Destination	-	Uind	0	Week month	2
Dep STD	B82	UTXOT (min)	5.242	Week	50
STD Group	AB	Wind dir. (°)	160	Month	12
Dep RWY	R29	Wind speed (kts)	4	Peak	0
AOBT	09:29:10	Visibility (km)	0.25	TXOT	00:24:14
ATOT	09:53:24	Ceiling (ft)	100		

Table 5.3: Data for flight AUA601

Having traced the conditions using the data from [Table 5.3](#), the terminal node (leaf) with a TXOT of 24.3 minutes is reached. Therefore, a TXOT prediction of 24.3 minutes (00:24:21) is made for flight 'X'. This is a 7s difference compared to the actual TXOT of 24.2 mins (00:24:14) and is thus a very accurate prediction of the TXOT. To validate the conclusions made regarding the reasons for the delay of the aircraft, the ground track and speed profile of the aircraft was plotted in [Figure 5.13](#). From both the ground track and speed profile, it is clear that the aircraft passes through a de-icing stand, thus confirming the suspicion of ice and snow, and thus the main reason for the delay. Additionally, the 2nd set of movements starting at 09:47 shows the queueing of the aircraft due to the congestion.

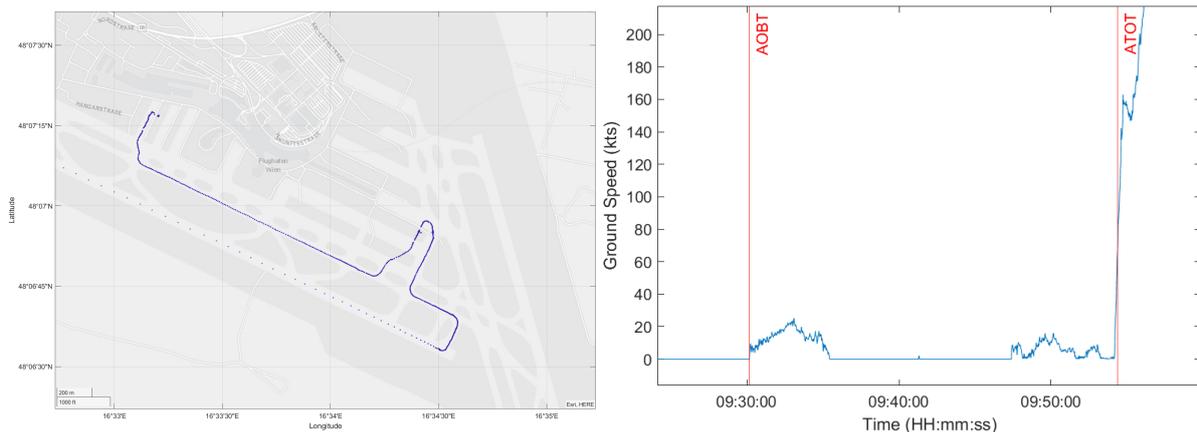


Figure 5.13: Ground track and speed profile for example flight 'X'

5.4. Distributions

The same methodology used in [section 4.2](#) was used to fit distributions to the leaves of the new tree. In [Table 5.4](#), the percentage of best fits per distribution is shown.

Distribution	%	Distribution	%
Birnbaum-Saunders	47.22	Lognormal	0
Gamma	6.94	Normal	0
Generalized Extreme Value (Gumbel)	19.44	tLocationScale	1.39
Logistic	20.83	Weibull	4.17

Table 5.4: % Best fit per distribution

Clearly, the best distribution is the ‘Birnbaum-Saunders’ distribution, whose equation is shown in [Equation 5.1](#):

$$f(x) = \left(\frac{\sqrt{\frac{x-\mu}{\beta}} + \sqrt{\frac{\beta}{x-\mu}}}{2\gamma(x-\mu)} \right) \phi \left(\frac{\sqrt{\frac{x-\mu}{\beta}} - \sqrt{\frac{\beta}{x-\mu}}}{\gamma} \right) \quad (5.1)$$

for $x > \mu; \gamma, \beta > 0$.

Here, γ is the shape parameter, μ is the location parameter, β is the scale parameter, and ϕ is the probability density function of the standard normal distribution.

In figure [Figure 5.14](#), the ‘Birnbaum-Saunders’ distributions fit to all 72 leaf nodes is shown.

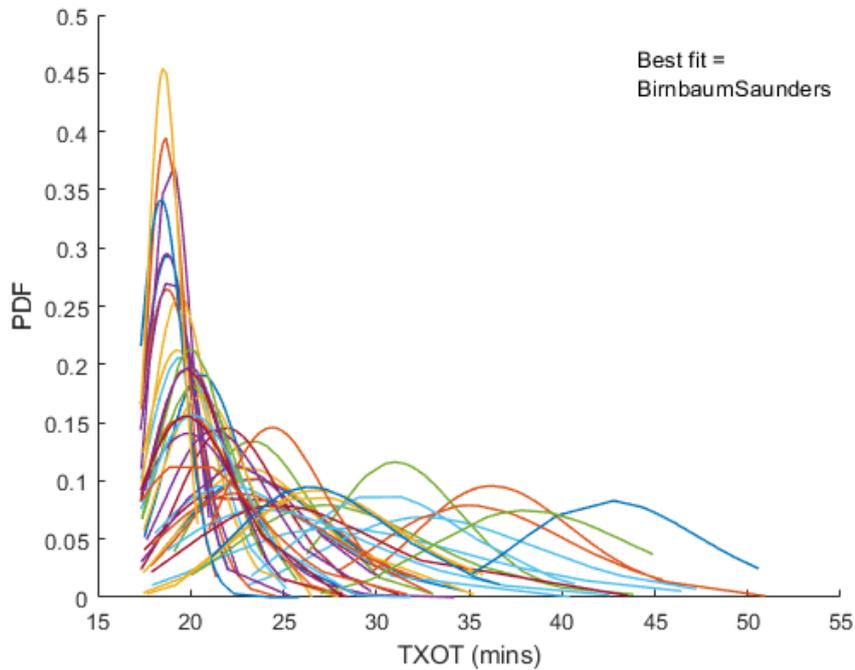


Figure 5.14: Birnbaum-Saunders distributions for all 72 leaf nodes

5.5. Top feature analysis

In [Figure 5.7](#), the top features extracted from the extended TXOT regression tree were presented. However, this figure presents only what they are, rather than the effect they have on TXOT. Since the

top features of the extended TXOTs differ to those in [section 4.3](#), an analysis is performed on these top features to observe their influence on TXOT.

1. Congestion level (ConLVL)

Once again, the most influential feature by far is the congestion level. As observed in [Figure 5.15](#), there is clear positive linear relationship between congestion level and TXOT. This feature was discussed in detail in [sections 4.3](#) and [subsection 2.2.1](#).

2. Departure count in the last 20 mins (Depcount)

In [Figure 5.16](#), the regression and histogram plots for the departure count can be found. The influence of departure count on extended TXOTs is very similar to that of normal TXOTs. A discussion of this can be found following [Figure 4.22](#) in [section 4.3](#).

3. Cloud ceiling height (Ceiling)

A unique top feature for extended TXOTs is the cloud ceiling height. From the regression plot in [Figure 5.17](#), it can be observed that in general, the lower the cloud ceiling, the longer the TXOT. This appears to be especially true for flights with a cloud ceiling of less than roughly 3000 ft. This increase in may be due to multiple factors. One such factor may be the reduction in visibility, which tends to reduce the operational efficiency of airports. Additionally, low clouds are often associated with rain or snow which can also influence operations.

4. Airport peak time (Peak)

Another top feature unique to extended TXOTs is whether or not an aircraft's AOBT occurs in peak hours. Peak hours at Vienna Airport were defined as between 07:30 - 09:30 and 16:30 - 19:30. Surprisingly, based on the regression in [Figure 5.18](#), it appears that higher TXOTs tend to occur outside of peak hours. One such explanation for this is that many extended TXOTs occur either very early in the morning, or late at night. Weather related factors are often more severe at these times which could result in longer TXOTs. Additionally, the peak hours may not have been defined completely accurately, with the 2nd peak encapsulating small non-peak times.

5. Airport throughput in the last hour (Throughput)

In [Figure 5.19](#), the regression and histogram plots for the throughput can be found. The influence of airport throughput on extended TXOTs is very similar to that of normal TXOTs. A discussion of this can be found following [Figure 4.24](#) in [section 4.3](#).

6. Dew point (Dewpt)

The dew point is defined as the "air temperature at which a sample of air would reach 100% humidity based upon its current degree of saturation"¹.

If the relative humidity of an air mass becomes 100% and the temperature falls, the mass can no longer hold all of the water vapour within it. This excess water vapour will then condense

¹https://www.skybrary.aero/index.php/Dew_Point, accessed 03/08/2020

into cloud or fog. Additionally, if this vapour is in contact with objects on or near the ground, dew or frost will form.

In [Figure 5.20](#), a sudden increase in TXOTs is observed given a dewpoint of less than 1° . Based on the description of the dew point, this increase is due to either ice formation on aircraft, in which case the aircraft will require de-icing, or the formation of fog, in which case visibility will be impeded, reducing operational efficiency.

A good indication of impending low visibility conditions and the possibility of fog is a reducing gap between the actual temperature and the dew point. In future work, an additional feature representing this difference should be added.

7. **Departure runway (DepRWY)**

Based on the regression plot in [Figure 5.21](#) it appears that most of the extended TXOTs occur on runway 29. It was expected that runway 34 incurs the most extended TXOTs due to its distance from the terminal. However, as observed, distance from the terminal appears to have very little effect. Instead, extended TXOTs occur most often based solely on the usage of the runway. Since R29 is the runway most commonly used for departures, queues are more likely to form at R29, and therefore additional TXOTs are incurred. Since the other runways are used much less frequently (especially R11) for departures, there is less chance of large departure queues forming at these runways.

8. **Wind speed (SpeedWMA)**

In [Figure 5.22](#), the regression and histogram plots for the wind speed can be found. This influence of wind speed on extended TXOT is the same as that of normal TXOT. A discussion of this can be found following [Figure 4.27](#) in [section 4.3](#).

9. **Actual off-block time (AOBT)**

The final top feature analysed is the AOBT, a feature also found in the top features of the normal TXOT. The regression and histogram plots for the AOBT can be found in [Figure 5.16](#). Once again, the influence the AOBT on extended TXOT is very similar to that of normal TXOT. A discussion of this can be found following [Figure 4.26](#) in [section 4.3](#).

5.5. TOP FEATURE ANALYSIS

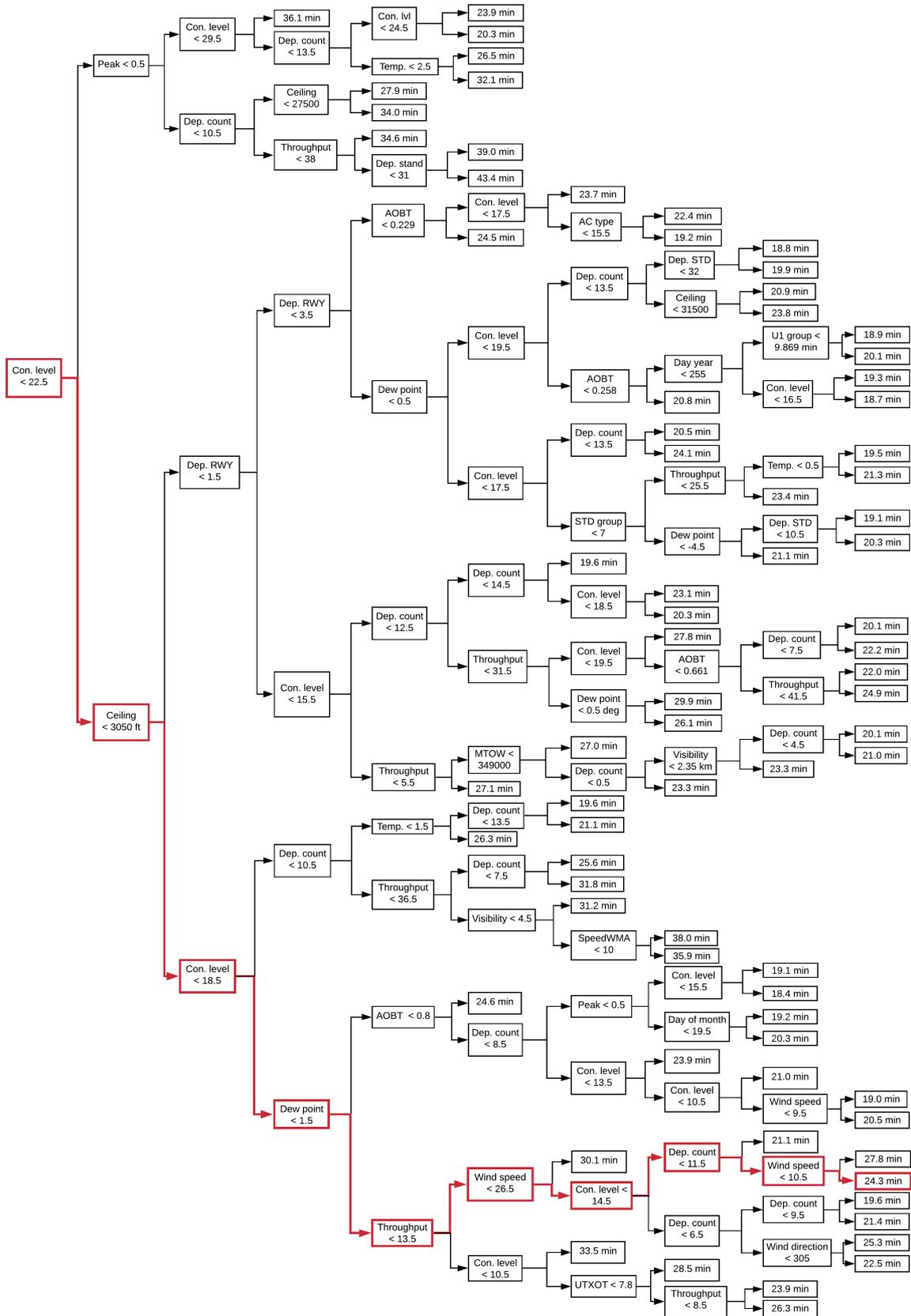


Figure 5.12: Regression Tree for extended TXOT prediction with example prediction flight 'X'

5.5. TOP FEATURE ANALYSIS

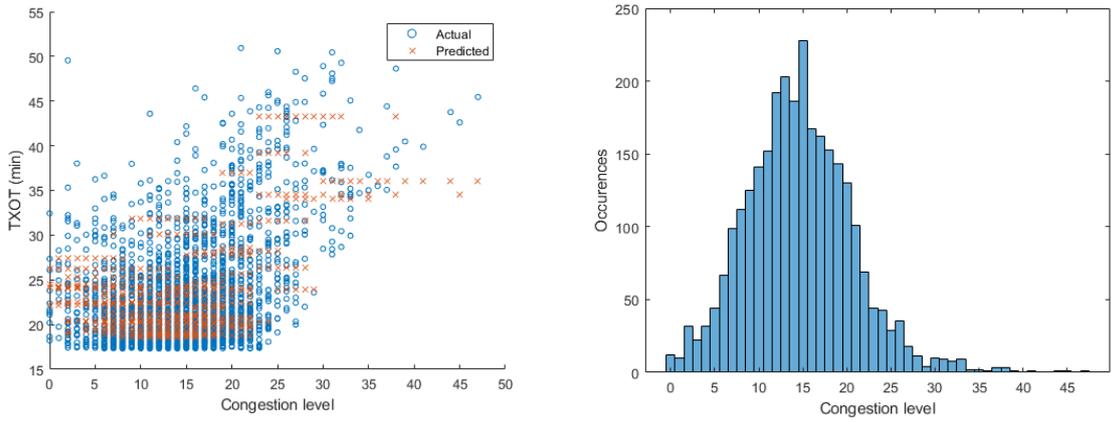


Figure 5.15: Congestion level vs. TXOT and Congestion level histogram

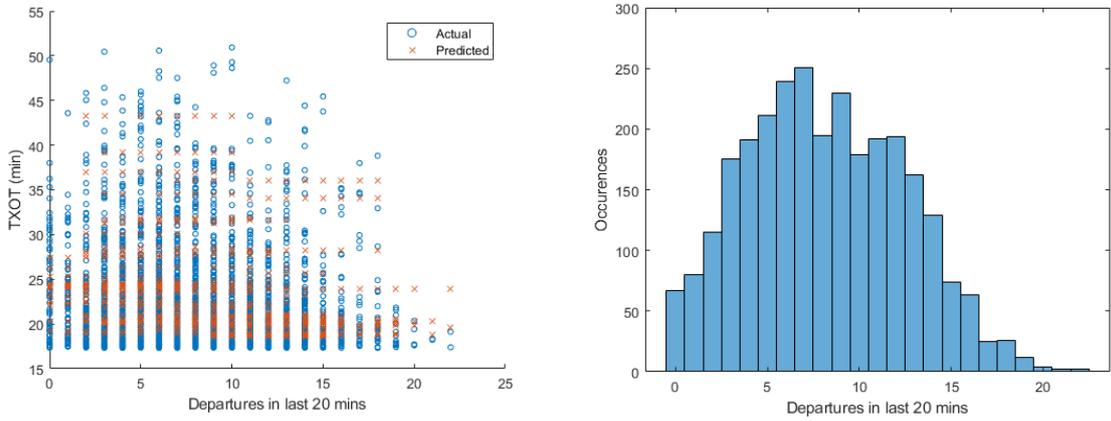


Figure 5.16: Departure count vs. TXOT and departure count histogram

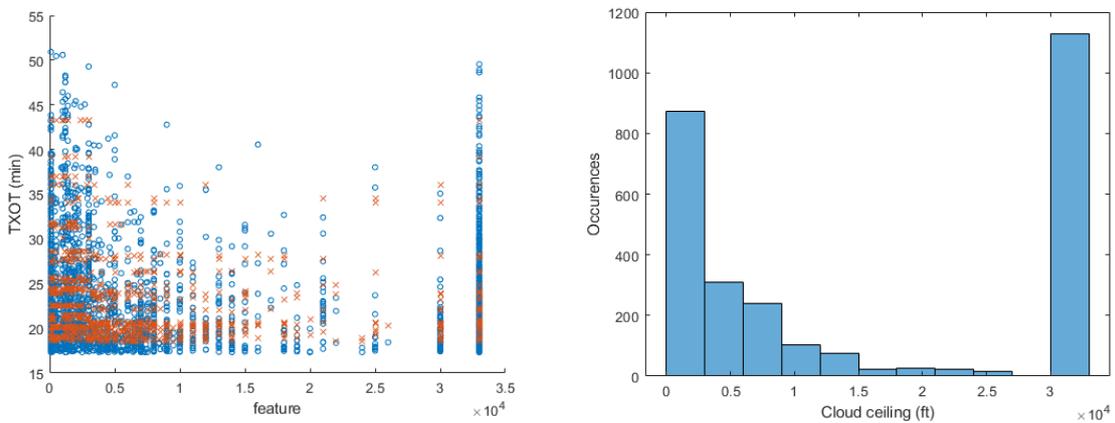


Figure 5.17: Cloud ceiling vs. TXOT and cloud ceiling histogram

5.5. TOP FEATURE ANALYSIS

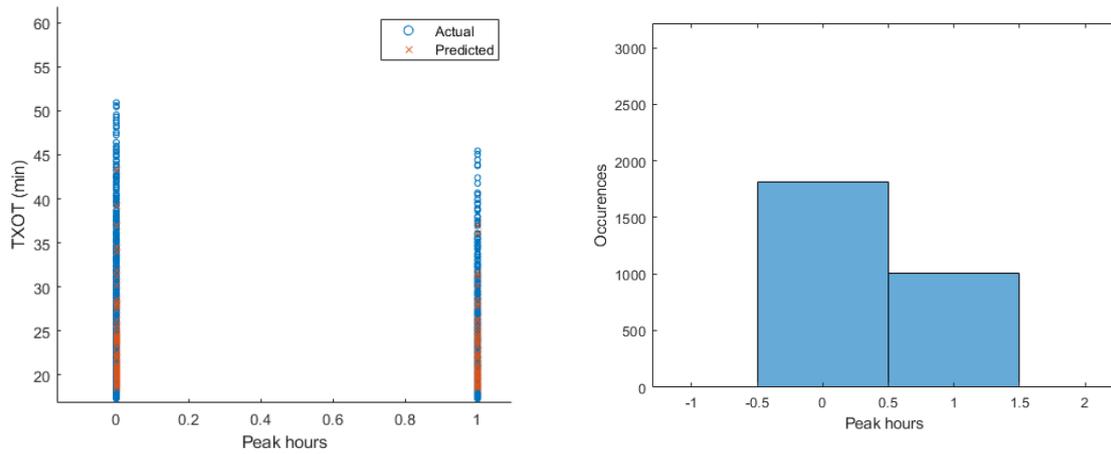


Figure 5.18: Peak time boolean vs. TXOT and peak time histogram

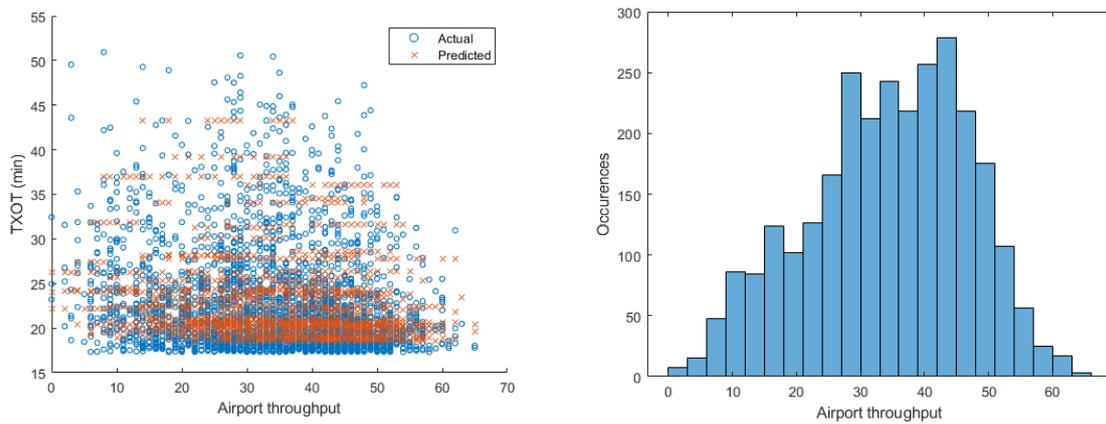


Figure 5.19: Throughput in last hour vs. TXOT and histogram of throughput in last hour

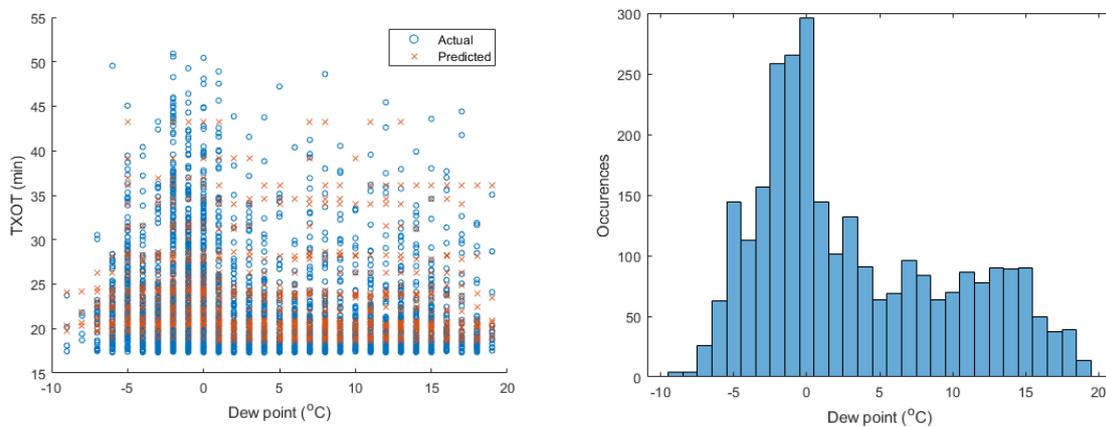


Figure 5.20: Dew point vs. TXOT and dew point histogram

5.5. TOP FEATURE ANALYSIS

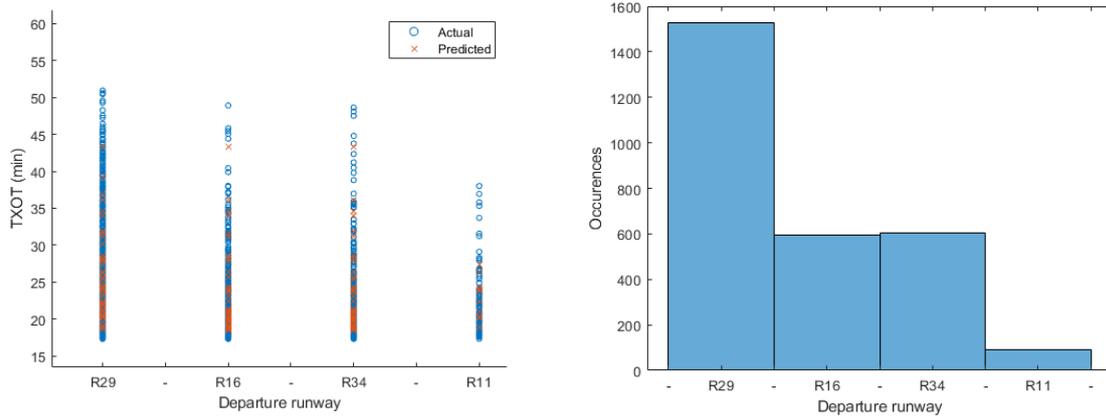


Figure 5.21: Departure runway vs. TXOT and departure runway histogram

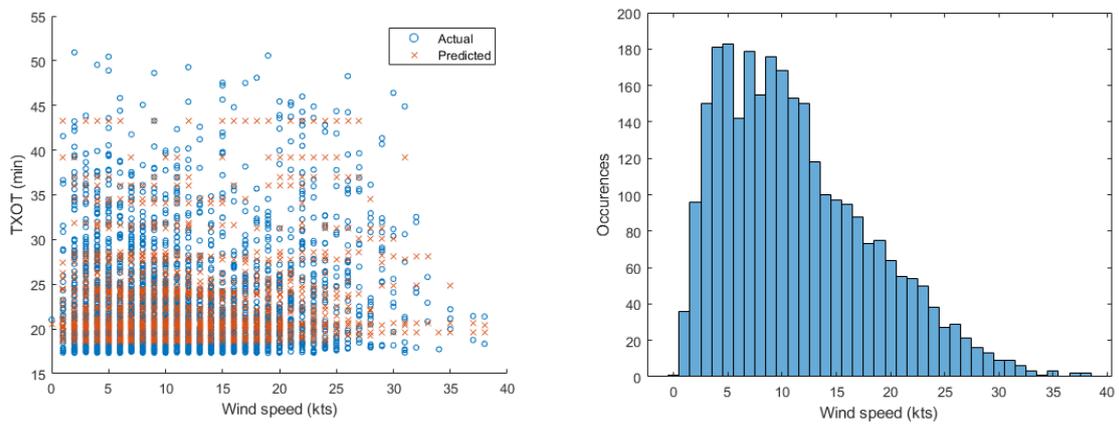


Figure 5.22: Wind speed vs. TXOT and wind speed histogram

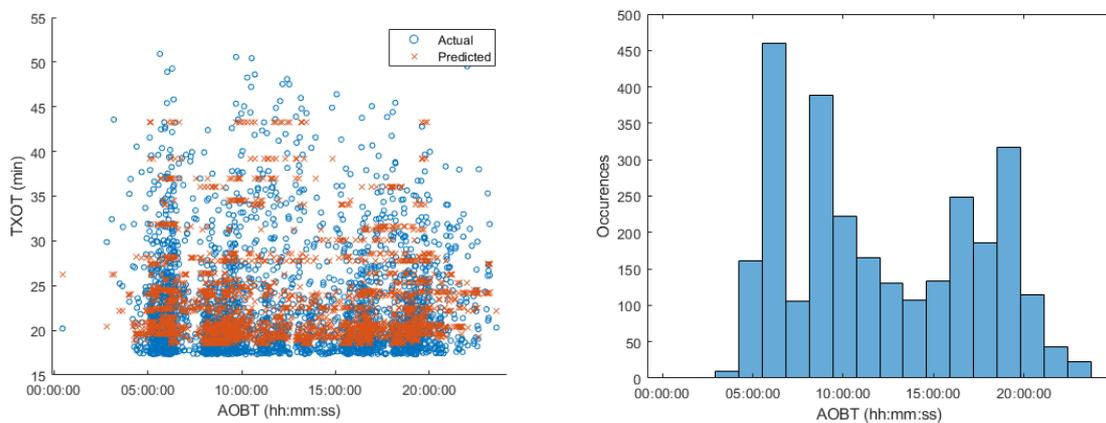


Figure 5.23: AOBT vs. TXOT and AOBT histogram

6

Verification and Validation

In this chapter the verification and validation of the models are discussed. The verification process is presented in [section 6.1](#), while the validation process is presented in [section 6.2](#).

6.1. Verification

In this section, an overview of the methods used to verify the models is presented. In the context of this study, verification consists of ensuring the steps outlined in the thesis were performed correctly. In order to obtain reliable results, the verification of three main steps had to be performed, namely, 'data preparation', 'model creation', and 'model results'. Each of these steps consisted of several substeps which are outlined in the list below. The steps were verified by analysing their inputs and outputs. This included making sure any calculations performed were correct, and any code written produced the expected output. The verification was performed while performing each of the steps, and as such, the model is verified. Additionally, regular meetings were held with supervisors to verify the work performed was sufficient.

1. **Data preparation:** the verification of the data preparation was largely covered in [section 3.2](#). For convenience, the steps are repeated here as follows:
 - (a) Perform a dataset quality check and clean the data:
 - Check consistency of AOBTs and ATOTs from A-SMGCS with Radar Track data, as described in [subsection 3.1.2](#).
 - Reject flights with a negative TXOT; flights with a negative TXOT are impossible and are therefore removed from the data set. Negative TXOTs can occur due to faults in the A-SMGCS where false timestamps are recorded.
 - Reject flights with an actual TXOT of more than 90 minutes
 - Reject helicopter flights. Helicopters do not follow the same departure procedure as airplanes and are thus removed from the dataset.
 - Remove incomplete records.
 - (b) Perform the following for each departure flight:
 - Calculate the actual TXOT: ensure equations is implemented correctly
 - Determine the congestion level by counting the number of take-offs and landings in the time interval between the AOBT and ATOT of the respective flight.

- Determine the airport throughput observed in the hour preceding the AOBT of the flight. The airport throughput is defined as the total number take-offs and landings at the airport.
 - Determine the runway-stand (RWY-STD) combination of the flight.
- (c) Determine the saturation level for each departure flight:
- Estimate the peak airport throughput (R) using the 90th percentile of the airport throughput in the preceding hour of all flights.
 - Next, calculate a first estimation of the UTXOT per RWY-STD combination (U_1) by using the 20th percentile of the actual TXOTs of the flights belonging to that RWY-STD combination.
 - Repeat the previous step to calculate the first estimation of the UTXOT per runway-stand-group combination (U_{group}).
 - Compute the saturation level (L) per RWY-STD and RWY-STD-group.
- (d) Identify unimpeded flights: flights are considered unimpeded if they meet the following condition: $congestionlevel \leq 0.5 * saturationlevel$. Ensure computation is properly performed.
- (e) Computation of UTXOT:
- If the number of unimpeded flights belonging to a specific RWY-STD combination is greater than or equal to 10, the UTXOT for that couple is defined as the median TXOT of the unimpeded flights.
 - If the number of unimpeded flights belonging to a specific RWY-STD combination is less than 10, no UTXOT is computed for that couple. Instead, the median TXOT of all unimpeded flights belonging to the RWY-STD-group is used as the UTXOT.

2. Model creation:

- (a) Input: ensure input prediction variables and target variables are those desired. For extended TXOT prediction, ensure only extended TXOT flights are used for training.
- (b) Minimum leaf size: check that no leaves are in fact smaller than the chosen parameter.
- (c) Maximum tree depth: ensure the actual tree depth matches the tree depth parameter set. This can be done visually or through the `treedepth` function we created. Also, check that the `treedepth` function performs correctly.
- (d) Outliers: remove outliers detected in the outlier analysis for extended TXOT.
- (e) Output: check if predicted TXOTs on validation set are reasonable

3. Model results:

- (a) Leaf distribution fitting: the distribution on each leaf should be verified to ensure the best fit
 - Perform a goodness-of-fit test per leaf: as mentioned in [section 4.2](#), the komogorov-smirnov test was used to test if a distribution fit the data sufficiently. Also, visually inspect the distribution to ensure it is a good fit.
 - Choosing the best fit: the best fit was determined using the minimum variance. Visually inspect the distribution of the leaf to ensure it is a good fit.

- (b) Feature extraction: check that the top features extracted using the `predictorImportance` function match with those in the tree.

6.2. Validation

The models are validated by comparing their results to the actual operational taxi data. This comparison provides an indication of their predictive performance. In chapters 4 and 5 the results of the models were compared to the actual data. The models both produced sufficiently accurate results as compared to the actual data, meaning both models are validated.

Additionally, the use of a validation data set ensured the models were not trained specifically to provide the best possible results for the test set. When training the model, the parameters were adjusted to provide the best results on the validation data set. Once the model parameters were finalised, only then were they applied to the test sets. The test set has no effect on the training, and therefore provides an independent measure of the predictive performance after training.

7

Conclusion and Recommendations

In this chapter the conclusions drawn from the results are presented in [section 7.1](#). Recommendations for future research are presented in [section 7.2](#).

7.1. Conclusions

To answer the main research question "Which taxi-out features most significantly influence the accurate prediction of normal and extended taxi-out times?", two machine learning models were created to predict normal and extended TXOTs at Vienna International Airport, from which these features could be extracted.

For normal TXOT prediction i.e. for flights that are part of the main distribution of TXOTs, a Regression Tree model is built. This model is based on 81829 departure flights from all runways at Vienna Airport between January 1st 2015 and December 31st 2015. The performance of the model is validated using 17535 flights, and finally tested on a further 17535 flights. The taxi-out times of the test set of flights are predicted with an RMSE of 2.03 minutes.

The distribution of the predicted flights shows that flights with an actual TXOT of between 3-5 minutes are often overpredicted, making the model slightly conservative. The remainder of the flights, however, are more accurately predicted. Additionally, the shapes of the distributions of the actual and predicted TXOTs match well, indicating that the characteristics of the taxi-out process are captured in the model.

Regarding the research question, after having trained the ML model, the most important TXOT related features were extracted. The features which most significantly influence the prediction of normal TXOTs are as follows:

- Congestion level
- First estimation of the Unimpeded Taxi-Out Time per RWY-STD combination
- Number of departures in the 20 minutes preceding the AOBT of the departing aircraft
- Temperature
- Airport throughput in the hour preceding the AOBT of the departing aircraft
- Unimpeded Taxi-Out Time

Other features include the Actual Off-Block time, the wind speed at the time of departure, the MTOW of the aircraft, visibility, and the departure runway.

For extended TXOT prediction i.e. for flights more than 2 standard deviations from the mean of all TXOTs, another Regression Tree model is built. This model is based on 2818 departure flights from all runways at Vienna Airport between January 1st 2015 and December 31st 2015. The performance of the model is validated using 603 flights, and tested on a further 604 flights. The test set taxi-out times are predicted with an RMSE of 3.75 minutes.

The distribution of the predicted flights shows that the model often over-predicts the TXOT of flights with lower actual extended TXOTs (17.5 - 18.5 mins). While there is some noise in the rest of the predictions, they are closer to the actual TXOTs. This noise could be smoothed given more 'extended' TXOT flight data to train the model. Lastly, the shapes of the distributions are similar in general, meaning the characteristics of the extended taxi-out process are captured in the model.

Regarding the research question, the following features were deemed most important with respect to extended TXOT prediction:

- Congestion level
- Number of departures in the 20 minutes preceding the AOBT of the departing aircraft
- Cloud ceiling height
- Airport peak hours
- Airport throughput in the hour preceding the AOBT of the departing aircraft
- Dew point
- Departure runway

Other features include, wind speed, actual off-block time, temperature, visibility, departure stand, wind direction, and aircraft weight.

From the above it can be concluded, as expected, that the most significant factors affecting normal and extended TXOTs relate to congestion levels at the airport. It is interesting to note that in the case of extended TXOTs the weather plays a more significant role (e.g. cloud ceiling, dew point).

7.2. Recommendations

In this section, recommendations for further research are discussed.

The first set of recommendations are in relation to the features chosen to assess the machine learning models. As shown in [section 3.10](#), one cause of significant taxi-out delay is the de-icing of aircraft. A 'de-icing stand' prediction variable should therefore be included, where if an aircraft needs to pass through a de-icing stand before take-off, the TXOT is expected to be significantly longer. However, this variable would need to be extracted from the latitude and longitude coordinates of the Radar track data, and thus depends on the data availability. Given only 6 of 12 months of Radar track data was provided at Vienna Airport, this feature could not be included.

Another feature that should be included is wake-vortex relationship between two sequentially departing aircraft. For example, if an aircraft with a lower wake vortex category is behind an aircraft with a higher wake vortex category, the trailing aircraft will not be allowed to take-off until the leading aircraft is a certain distance away. Conversely, if an aircraft with a higher wake-vortex category is departing after an aircraft with a lower wake-vortex category, the trailing aircraft will not have to

7.2. RECOMMENDATIONS

wait as long. This time spent waiting will contribute to TXOT and should therefore be investigated in future.

Other features that could be investigated include Stand/Gate Availability, and whether a flight is private or commercial.

Regarding the training of the tree, the node splitting algorithm used to determine how to split a node was the standard CART (Classification and Regression Tree) algorithm. Different algorithms such as curvature test and the interaction test could be explored. This may result in different top features being found.

Finally, another recommendation would be to test the models at a different, preferably busier airport. Vienna Airport is quite small, meaning it is not saturated often, and large queues do not form frequently. Given a busier airport with more queues, a larger variation in TXOTs could be explored. Whether or not the same features have the same influence at different airports could also be investigated.

Bibliography

- [1] F. Herrema, R. Curran, H. Visser, D. Huet, and R. Lacote, “Taxi-Out Time Prediction Model at Charles de Gaulle Airport,” *Journal of Aerospace Information Systems*, vol. 15, no. 3, pp. 120–130, 2018.
- [2] P. Balakrishna, R. Ganesan, and L. Sherry, “Accuracy of reinforcement learning algorithms for predicting aircraft taxi-out times: A case-study of Tampa Bay departures,” *Transportation Research Part C: Emerging Technologies*, vol. 18, no. 6, pp. 950–962, 2010.
- [3] L. Capelleras, “Additional Taxi-Out Time Performance Indicator document,” tech. rep., EUROCONTROL, 2015.
- [4] Y. Zhang and Q. Wang, “Methods for determining unimpeded aircraft taxiing time and evaluating airport taxiing performance,” *Chinese Journal of Aeronautics*, 2017.
- [5] I. Simaiakis and H. Balakrishnan, “A queuing model of the airport departure process,” *Transportation Science*, vol. 50, no. 1, pp. 94–109, 2016.
- [6] R. A. Shumsky, “Dynamic Statistical Models for the prediction of aircraft take-off times,” *PhD Thesis*, pp. 1–214, 1995.
- [7] N. Pujet, B. Delcaire, and E. Feron, *Input-output modeling and control of the departure process of congested airports*, pp. 1835–1852. American Institute of Aeronautics and Astronautics Inc, AIAA, 1999.
- [8] “Fit Data with a Shallow Neural Network - MATLAB & Simulink - MathWorks Benelux.”
- [9] E. M. De Mol, “Modelling the airport departure process under the influence of uncertainty,” Master’s thesis, TU Delft, 2019.
- [10] S. Ravizza, J. Chen, J. A. Atkin, P. Stewart, and E. K. Burke, “Aircraft taxi time prediction: Comparisons and insights,” *Applied Soft Computing Journal*, vol. 14, no. PART C, pp. 397–406, 2014.
- [11] S. Ravizza, J. A. Atkin, M. H. Maathuis, and E. K. Burke, “A combined statistical approach and ground movement model for improving taxi time estimations at airports,” *Journal of the Operational Research Society*, vol. 64, no. 9, pp. 1347–1360, 2013.
- [12] H. Idris, J.-P. Clarke, R. Bhuvana, and L. Kang, “Queuing Model for Taxi-Out Time Estimation,” *Air Traffic Control Quarterly*, vol. 10, no. 1, pp. 1–22, 2017.
- [13] R. R. Clewlow, I. Simaiakis, and H. Balakrishnan, “Impact of arrivals on departure taxi operations at airports,” *AIAA Guidance, Navigation, and Control Conference*, no. August, pp. 1–21, 2010.

BIBLIOGRAPHY

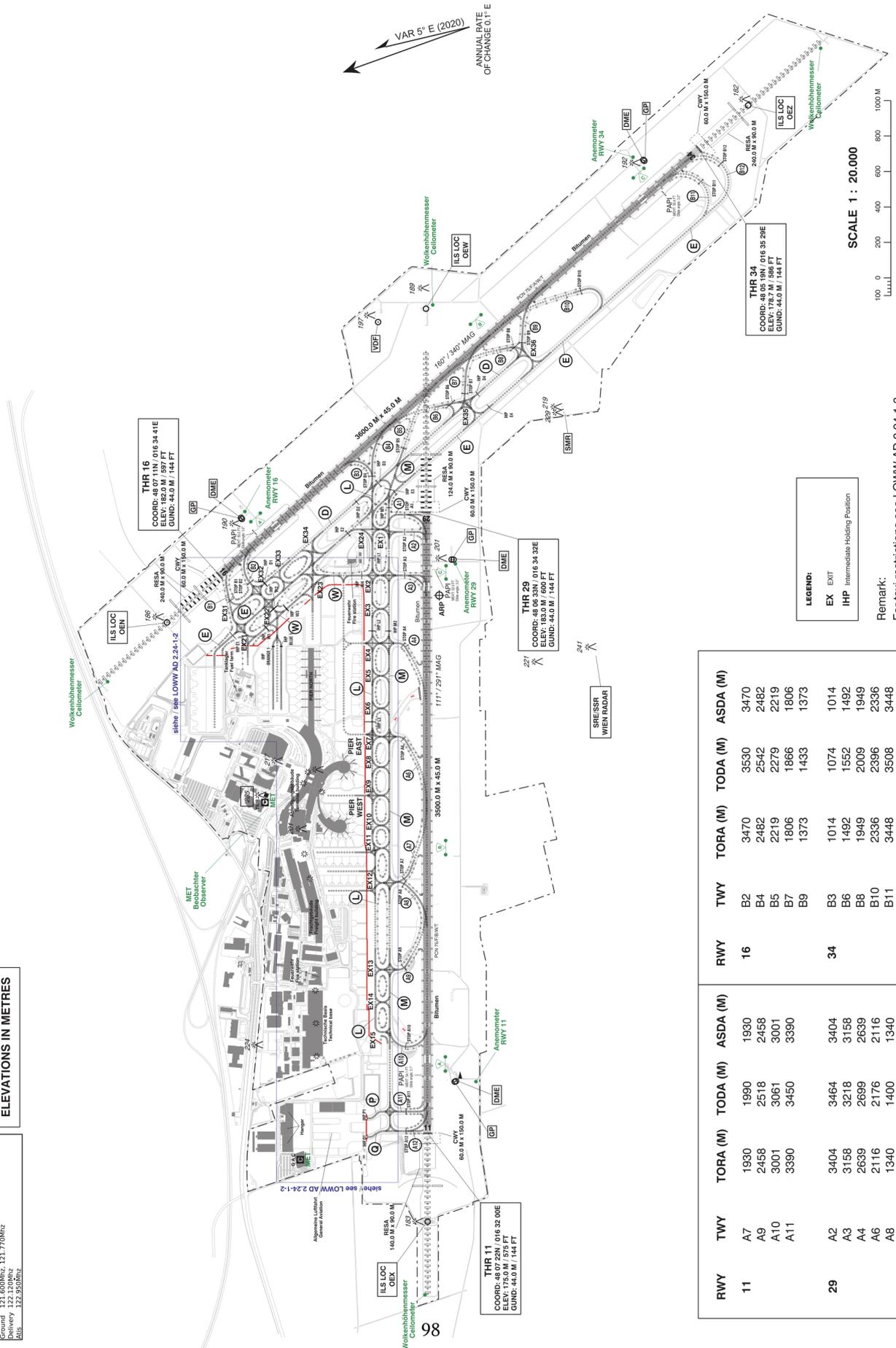
- [14] V. Sundarapandian, *Probability, Statistics and Queuing Theory*. PHI Learning, 2009.
- [15] J. E. Hebert and D. C. Dietz, “Modeling and analysis of an airport departure process,” *Journal of Aircraft*, 1997.
- [16] I. Simaiakis and N. Pyrgiotis, “An analytical queuing model of airport departure processes for taxi out time prediction,” in *10th AIAA Aviation Technology, Integration and Operations Conference 2010, ATIO 2010*, 2010.
- [17] K. Andersson, F. Carr, E. Feron, and W. D. Hall, “Analysis and Modeling of Ground Operations at Hub Airports,” *3rd USA/Europe Air Traffic Management R&D Seminar*, no. June, pp. 13–16, 2000.
- [18] I. Simaiakis and H. Balakrishnan, “Queuing models of airport departure processes for emissions reduction,” in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2009.
- [19] H. F. Friso, C. Richard, H. G. Visser, T. Vincent, and D. Bruno, “Predicting Abnormal Runway Occupancy Times and Observing Related Precursors,” *Journal of Aerospace Information Systems*, vol. 15, no. 1, pp. 10–21, 2017.

A

ICAO Map - Vienna International Airport

Frequenz 118.700kHz, 118.770kHz, 124.550kHz, 129.050kHz
 Director 119.800kHz, 123.800kHz
 Power 119.400kHz, 123.800kHz
 Ground 121.500kHz, 121.770kHz
 ATIS 122.100kHz, 122.170kHz
 ATIS 122.950kHz

HÖHEN IN METERN
ELEVATIONS IN METRES



RWY	TWY	TORA (M)	TODA (M)	ASDA (M)	RWY	TWY	TORA (M)	TODA (M)	ASDA (M)
11	A7	1930	1990	1930	16	B2	3470	3530	3470
	A9	2458	2518	2458		B4	2482	2542	2482
	A10	3001	3061	3001		B5	2219	2279	2219
	A11	3390	3450	3390		B7	1806	1866	1806
		3450	3450	3390		B9	1373	1433	1373
29	A2	3404	3464	3404	B3	1014	1074	1014	
	A3	3158	3218	3158	B6	1492	1552	1492	
	A4	2639	2699	2639	B8	1949	2009	1949	
	A6	2116	2176	2116	B10	2336	2396	2336	
		1340	1400	1340	B11	3448	3508	3448	

LEGEND:
 EX EXIT
 IHP Intermediate Holding Position

SCALE 1 : 20.000



Remark:
 For taxi restrictions see LOWW AD 2.24-1-3

Figure A.1: ICAO Aerodrome Chart of Vienna Airport

B

Gate Group Definitions

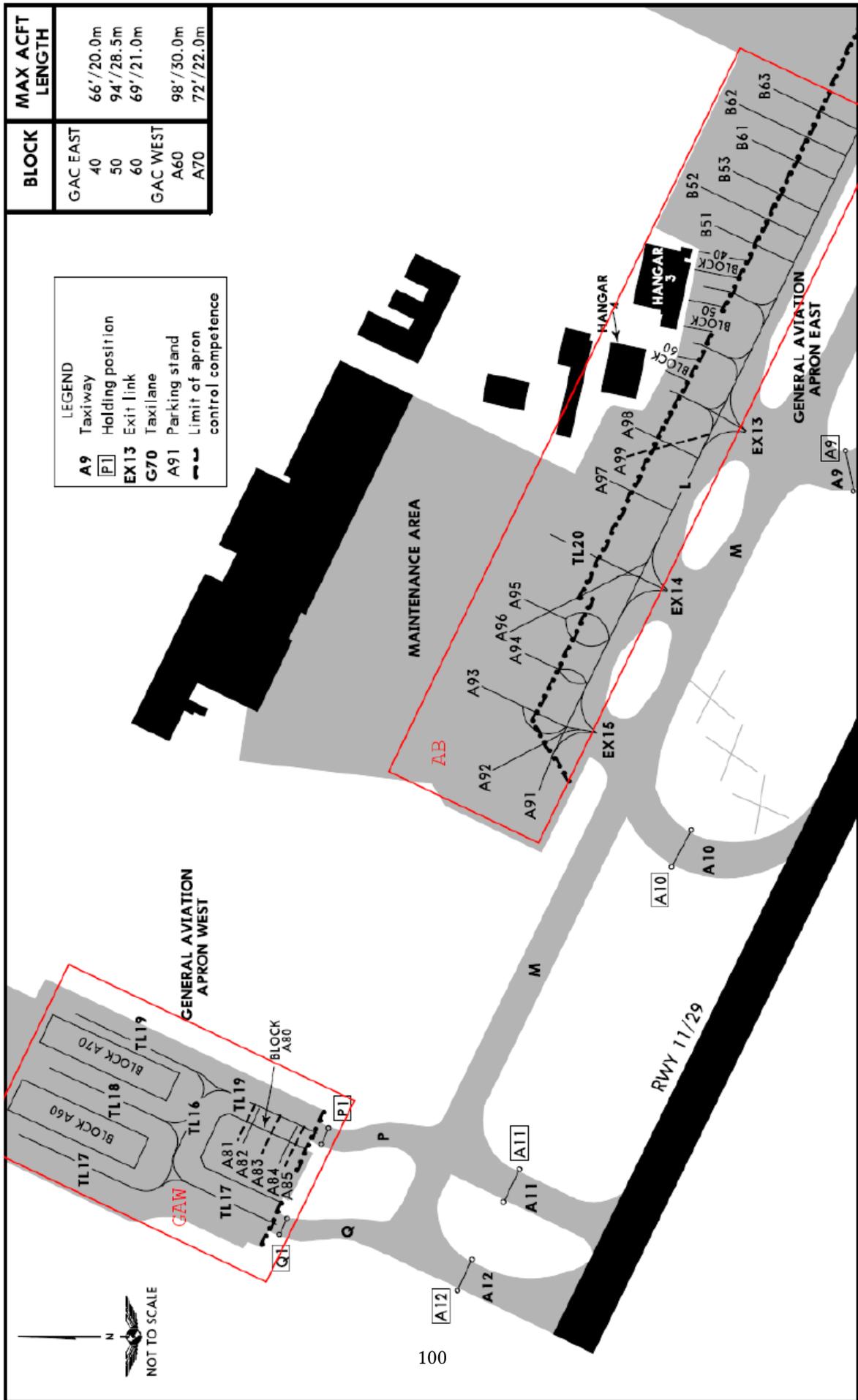


Figure B.1: Map of Vienna Airport: Gate groups - part 1. Obtained from [9]

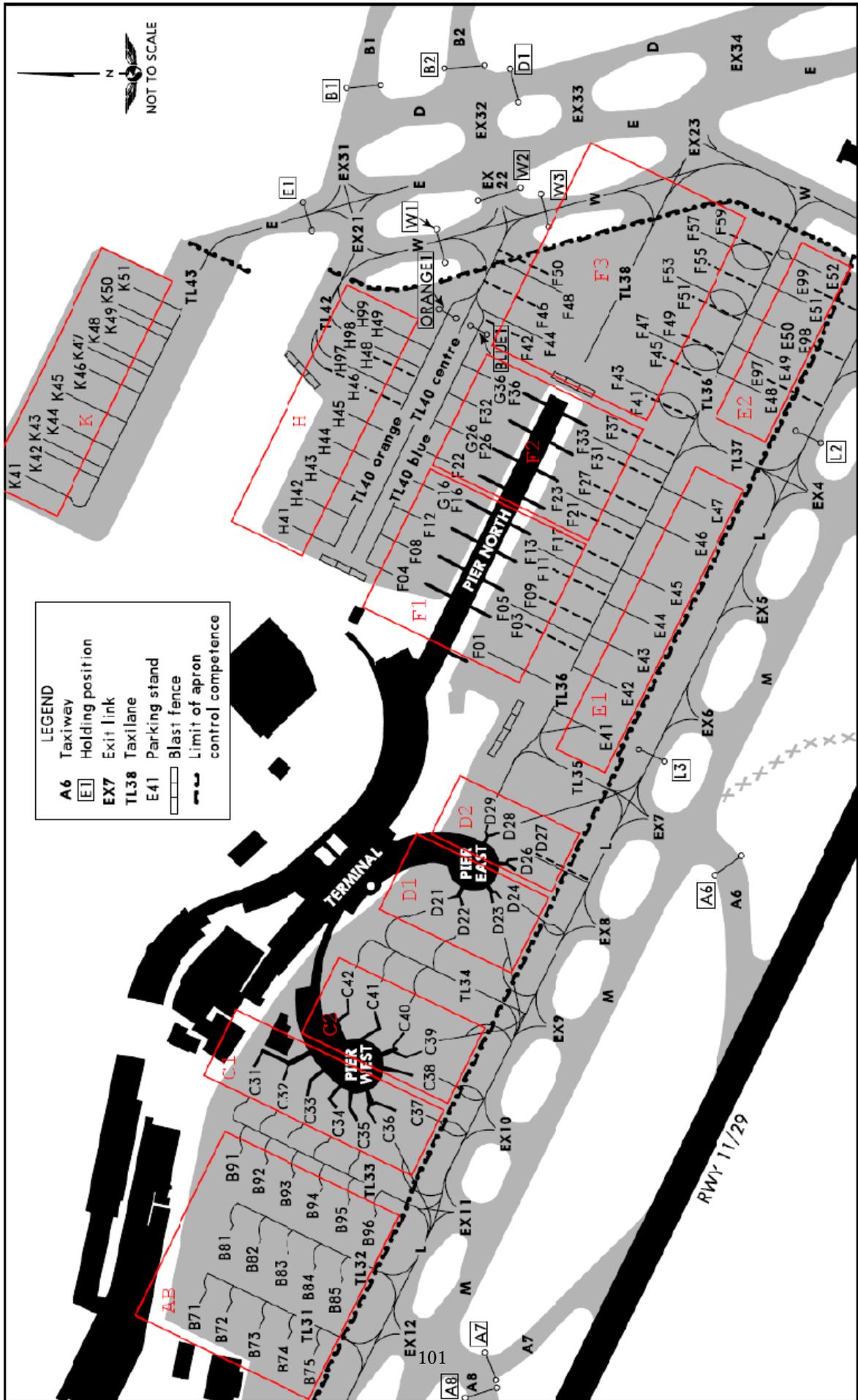


Figure B.2: Map of Vienna Airport: Gate groups - part 2. Obtained from [9]