Exploitation of Machine Learning to predict airport Runway Utilisation relative to known precursors and abnormality

Herrema, F.F.

DOI
10.4233/uuid:e96d5b2c-f0ce-4cbe-b1c4-bf5c408ad546

Publication date
2020

Document Version
Final published version

Citation (APA)
Herrema, F. F. (2020). *Exploitation of Machine Learning to predict airport Runway Utilisation relative to known precursors and abnormality*. [Dissertation (TU Delft), Delft University of Technology]. https://doi.org/10.4233/uuid:e96d5b2c-f0ce-4cbe-b1c4-bf5c408ad546

Exploitation of Machine Learning to predict airport Runway Utilisation relative to known precursors and abnormality

Exploitation of Machine Learning to predict airport Runway Utilisation relative to known precursors and abnormality

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus Prof.dr.ir. T.H.J.J. van der Hagen,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op
woensdag 4 maart 2020 om 12:30

door

Floris Friso HERREMA

Master of Science in Aerospace Engineering, Technical University of Delft
Geboren te Steinheim, Duitsland.

Dit proefschrift is goedgekeurd door de

promotor: Prof.dr. R. Curran
copromotor: Dr.ir. B.F. Lopes dos Santos

Samenstelling promotiecommissie:

Prof. Dr-Ing. D. Kugler
Prof. Dr-Ing. H. Fricke
Prof. dr. B.A. van de Walle
Prof. dr. ir. G.J.P.M. Houben

Onafhankelijke leden:

Ir. V. Treve

Reserve lid:

Prof.dr.ir. J.M. Hoekstra

Science is a wonderful thing
if one does not have to earn one's living at it.
Albert Einstein

Acknowledgements

This thesis is the result of a 48-month-long PhD thesis study performed at the Airport Research Unit of EUROCONTROL. The research assignment was done by EUROCONTROL under the H2020 Project SafeClouds and the Air Transport and Operations Department of the Aerospace Engineering faculty at Delft University of Technology. This study was executed in association with various airports, Air Navigation Service Providers, pilots and air traffic controllers.

I would like to thank several people who contributed to this thesis. First, I would like to thank my promotor, Ricky Curran for guiding and supporting me over the years. You have set an example of excellence as a researcher, mentor, instructor and friend. Also, I would like to thank my co-promotors Bruno Lopes dos Santos, Dries Visser and Sander Hartjes, for their guidance, support, and for providing me with the freedom to define and carry out my PhD research project.

Second, I would like to thank my thesis committee members for their guidance throughout this process; your discussion, ideas and feedback have been invaluable. I am extremely grateful to all of you.

Third, I am particularly thankful to my amazing family for their love, support and constant encouragement over the years. I would like to thank my wife Louise, son Lucas and my parents Klaas and Manon. You are the salt of the earth and I could not have done this without you.

Fourth, I am grateful to my sister, Annemarthe Herrema, and dedicate this thesis to her. It has been two years since you passed away but I still carry your lessons within me every day.

Finally, I would like to thank Austrocontrol for providing access to the aircraft final approach radar and runway data. Particularly, VIE ATCO Philipp Wächter, runway throughput expert Vincent Treve and head of airport research at EUROCONTROL Bob Graham for their valuable insights. I also wishes to acknowledge the support and expertise of the technical teams at SafeClouds that made this work possible.

Herrema, F.F., Delft, 10 March 2020

Summary

There is currently no supplementary operational system that assists the Air Traffic Control Officer (ATCO) in attaining *accurate*, *fast*, *intuitive* and *interpretable* predictions of Aircraft Safety Performance (ASP) enablers through suitable *visualisation* on the runway or on final approach. Thus, this study intends to develop an arrival ATCO support decision tool named the **Runway Utilisation (RU)** support tool.

The prediction of the ASP enablers is significant for ensuring a safer and efficient runway throughput in real-time operations. The RU support tool can become a realistic decision-making tool for the ATCO if the following hypothesis can be proved valid for each ASP enabler:

Machine learning can be used to effectively identify ASP patterns, risks and precursors, resulting in the extraction of RU requirements for tactical and strategic decision-making.

The RU support tool contributes to improving safety and making better separation decisions for aircraft pairs by addressing ASP enablers as defined by ATCOs. Three independent ATCOs from different hub airports were interviewed in this study. The following ASP enablers were considered to be the most significant due to their subsequent impact on runway throughput operations:

(1) Time to Fly (T2F) and True Airspeed (TAS), leading to a better characterisation of large spacings or infringements;
(2) (abnormal) Taxi-Out Times (TXOT);
(3) (abnormal) Arrival Runway Occupancy Times (AROT) and
(4) Procedural and Non-Procedural Runway Exit Use (NREX)

These ASP enablers depend on two key precursors. The first precursor is the impact of the prevailing meteorological and airport conditions, while the second precursor concerns runway congestion and decay of the wake turbulence. The following conclusions can hence be drawn in relation to the four ASP enablers along with their subsequent impact on runway throughput operations.

(1) Considering the most important prediction variable – ground speed at 10NM – might lead to certain operational issues. In order to be able to predict the T2F in real time, an ATCO has to wait until the aircraft is at 10NM. The T2F helps the ATCO calculate the compression on final approach using, for example, the Time Based Separation (TBS) concept. The dynamic TBS for the follower aircraft must be known before 10NM. Therefore, it is suggested to predict the Ground Speed (GS) at 10NM of the previous aircraft (based on the historical flight information of that time period). Moreover, the computational time (10 seconds) might be too large for real-time operations. We conclude that our hypothesis is not true for the ASP enabler T2F and TAS. Tactical predictions should be produced faster, and tactical and strategic decisions should be made before 10NM in this case.

(2) From our prototype, we can conclude that machine learning (ML) is feasible for extracting precursors and patterns that support the controller when it comes to tactical and strategic decision-making. TXOT precursors were mainly observed when the unimpeded time was larger than 22 minutes and the congestion level was greater than 32 movements per hour. The downside of our model is that we did not have access to a Real Time Simulation (RTS) that could validate this ASP enabler in a Charles de Gaulle (CDG) environment. Furthermore, the computational time (80 seconds) was too large for testing the model in a real-time operational environment.

(3) It can be noted that we can use Classification and Regression Tree (CART) to extract abnormal AROT patterns, risks and precursors for tactical and strategic decision-making. Therefore, the AROT conclusions for the RU requirements, ATCO operational needs and operational feasibility are addressed in Section 8.1.

(4) We concluded that we can use Gradient Boosting to extract NREX patterns, risks and precursors for tactical and strategic decision-making. The risks and the most important NREX precursors were identified for cases in which the throughput was lower than 28 landings per hour, the Cloud layer was less than 8750m, the Groundspeed at 2NM was higher than 147kts, the WMAWindSpeed was lower than 29kts and the Groundspeed at 5NM was higher than 155kts. These precursors could be used during similar situations, thereby allowing the ATCO to anticipate a non-procedural exit (intuitive). NREX operational needs, operational feasibility and RU requirement conclusions are presented in Section 8.1.

AROT and NREX were selected, as they allow us to make *intuitive*, *interpretable*, *visual*, *quick* and *accurate* decisions through suitable visualisation. Therefore, we conducted an operational needs and operational feasibility study to analyse the manner in which our RU support tool (AROT and NREX) can be used by ATCOs in their decision-making and planning in order to ensure safety and efficiency (*accurate*, *fast*, *intuitive* and *interpretable*) of airport operations through suitable *visualisation*. The feasibility study was conducted with an ATCO RTS tool.

Therefore, based on the findings from the validation activity, the validation was completed and the ML RU tool was reported to meet controllers' operational needs and provide certain safety benefits. The impact of an ML RU controller support tool on controllers' work and runway operations requires further investigation in follow-on validation activities. Potential benefits and impacts relating to the ML RU controller support tool that require more detailed investigation in upcoming validation activities are outlined in Section 8.3.

Finally, the ATCOs concluded that the RTS was successful in predicting both AROT and NREX. They observed improved operations in certain weather conditions, including an increased runway throughput and potential for a greater level of safety. In conclusion, the result of the present research study presents a new RU tool that enables the provision of unique interpretable and intuitive information from AROT and NREX patterns on final approach and the runway. The Gradient Boosting technique proves ideal for the detection of patterns, risk and precursors. When predicting the NREX, 95 decision trees and 12 features were used. Consequently, tactical and/or strategic decisions can be supported using this approach.

Samenvatting

Momenteel bestaat er geen aanvullend operationeel systeem dat de Air Traffic Controller (ATCO) helpt bij het voorspellen van Aircraft Safety Performance (ASP) enablers op en rond de startbaan of bij de nadering van de startbaan, rekening houdend met vijf aanwijsbare vereisten: *nauwkeurige, snelle, intuïtieve en interpreteerbare* ASP-voorspellingen verstrekt door een geschikte *visualisatie*. Daarom zal deze studie een ATCO-beslissingsondersteunings hulpmiddel voor landende vliegtuigen ontwikkelen, RU genaamd, dat staat voor **Runway Utilisation** voorspellingstool.

Deze ASP voorspellingen kunnen zeer waardevol zijn voor een veiligere en efficiëntere doorvoer van vliegtuigen op en rond de landingsbaan tijdens de werkelijke inzet (real-time operations). De RU voorspellingstool kan een realistische besluitvormings onderdeel worden als de volgende hypothese geldig kan worden bewezen voor de hier later benoemde ASP-enablers:

Machine Learning kan worden gebruikt om effectief ASP-patronen, risico's en voorbodes te identificeren, resulterend in de extractie van RU voorwaarden voor tactische en strategische besluitvorming.

De RU voorspellingstool draagt bij aan het verbeteren van de veiligheid en het nemen van betere scheidingsafstandsbeslissingen voor vliegtuigparen door het nader bestuderen van ASP-enablers, zoals gedefinieerd door ATCO's. Drie onafhankelijke ATCO's van verschillende grote luchthavens werden geïnterviewd. De volgende ASP-enablers werden door hen als meest belangrijk beschouwd vanwege hun invloed op de doorvoer van landende vliegtuigen:

(1) Time to Fly (T2F) en True Airspeed (TAS) welke leiden tot een betere karakterisering van grote of te korte afstanden op het laatste stuk voor de landing;
(2) (abnormale) Taxi-Out Tijden (TXOT);
(3) (abnormale) Runway Occupancy Tijden (AROT) en
(4) Procedureel en niet-procedureel gebruik van de baanuitgang (NREX).

Deze ASP-enablers zijn afhankelijk van twee voorbodes. De eerste is de invloed van de heersende meteorologische en luchthavenomstandigheden, waarna de tweede de landingsbaanverzadiging en het verval van de turbulentie betreft. De volgende conclusies kunnen worden getrokken met betrekking tot de vier ASP-enablers, beoordeeld op hun invloed op de doorstroom van landende vliegtuigen op en rond de landingsbaan.

(1) Kijkend naar de belangrijkste voorspellingsvariabele - grondsnelheid bij 10NM - kan leiden tot bepaalde operationele problemen. Om de T2F in real-time te kunnen voorspellen, moet een ATCO wachten tot het vliegtuig zich op 10NM bevindt. De T2F ondersteunt de ATCO om de samendrukking afstand te berekenen met behulp van bijvoorbeeld het Time Based Separation (TBS) concept. De dynamische TBS voor het volgvliegtuig moet vóór 10NM bekend zijn. Daarom wordt voorgesteld om de grondsnelheid

(GS) te voorspellen op 10NM van het vorige vliegtuig (op basis van historische vluchtinformatie van die periode). Bovendien kan de rekentijd (10 seconden) te groot zijn voor real-time bewerkingen. We concluderen dat onze hypothese niet waar is voor de ASP enabler T2F en TAS. Tactische voorspellingen moeten sneller worden gemaakt en tactische en strategische beslissingen moeten in dit geval vóór 10 NM worden genomen.

(2) Uit ons prototype kunnen we concluderen dat Machine Learning (ML) haalbaar is voor het extraheren van voorbodes en patronen die de ATCO ondersteunen bij tactische en strategische besluitvorming. TXOT voorbodes werden vooral waargenomen wanneer de ongehinderde TXOT tijd groter is dan 22 minuten en het verzadigingsniveau groter is dan 32 bewegingen per uur. Een tekortkoming van ons model is dat we geen toegang hebben tot een Real Time Simulation (RTS) die deze ASP-enabler in een Charles De Gaulle (CDG) omgeving kunnen valideren. Bovendien is de rekentijd (80 seconden) te groot om het model in een real-time operationele omgeving te testen.

(3) Er kan worden geconcludeerd dat we de classificatie- en regressieboom (CART) kunnen gebruiken om abnormale AROT-patronen, risico's en voorbodes te extraheren voor tactische en strategische besluitvorming. Daarom worden de AROT-conclusies voor de RU voorwaarden, operationele behoeften en operationele haalbaarheid van ATCO behandeld in paragraaf 8.1.

(4) Geconcludeerd kan worden dat we Gradient Boosting kunnen gebruiken om NREX-patronen, risico's en voorbodes te extraheren voor tactische en strategische besluitvorming. Risico's en de belangrijkste NREX voorbodes werden geïdentificeerd voor gevallen waarin de doorvoer lager is dan 28 landingen per uur, de Cloud-laag kleiner is dan 8750m, de Grondsnelheid bij 2NM hoger is dan 147kts, WMAWindSnelheid lager is dan 29kts en de Grondsnelheid bij 5NM hoger is dan 155kts. Deze aanwijzingen kunnen worden gebruikt in vergelijkbare situaties waardoor de ATCO kan anticiperen op een niet-procedurele exit (intuïtief). NREX operationele behoeften, operationele haalbaarheid en RU voorwaarden conclusies worden gepresenteerd in paragraaf 8.1.

AROT en NREX werden geselecteerd omdat ze ons in staat stellen om intuïtieve, interpreteerbare, visuele, snelle en nauwkeurige beslissingen te nemen door geschikte visualisatie. Vervolgens hebben we een operationele behoeften- en operationele haalbaarheidsstudie uitgevoerd waarin we hebben geanalyseerd hoe onze real-time RU-voorspellingstool (AROT en NREX) door ATCO's kan worden gebruikt bij hun besluitvorming en planning om veiligheid en efficiëntie (nauwkeurig, snel, intuïtief en interpreteer baar) van luchthavenactiviteiten door geschikte visualisatie (hoofdstuk 7). De haalbaarheidsstudie werd uitgevoerd in een ATCO RTS-tool.

Uiteindelijk kunnen we op basis van de bevindingen van de validatieactiviteiten concluderen dat het ML RU-hulpmiddel voldoet aan de ATCO operationele behoeften en mogelijke veiligheidsvoordelen biedt. De impact van een ML RU-controller ondersteuningstool op de operationele werkuitvoering van de ATOC's moet verder worden onderzocht in een vervolg validatieactiviteit. Potentiële voordelen en effecten met betrekking tot het ondersteuningsinstrument worden beschreven in paragraaf 8.3 en zullen nader worden onderzocht in de komende validatieactiviteiten.

Tenslotte concludeerden de ATCO's dat de RTS succesvol was in het voorspellen van zowel de AROT als de NREX. Ze zagen verbeterde operaties in bepaalde weersomstandigheden, waaronder een verhoogde doorvoer van vliegtuigstromen op de landingsbaan en een potentieel verhogend veiligheidsniveau. We concluderen dat het resultaat van het huidige onderzoek een nieuw RU-tool is die het mogelijk maakt om unieke interpreteerbare en intuïtieve informatie te bieden uit AROT en NREX-patronen voor de uiteindelijke nadering naar en landing op de landingsbaan. De Gradient Boosting-techniek is ideaal voor het detecteren van patronen, risico's en voorbodes. Wanneer we de NREX voorspellen met deze techniek hebben we 95 classificatiebomen nodig en 12 features. Tactische en/ of strategische beslissingen worden ondersteund met behulp van deze aanpak.

Table of contents

Acknowledgements .... *vi*

Summary .... *vii*

Samenvatting .... *ix*

Table of contents .... *xii*

List of Figures .... *xvi*

List of Tables .... *xviii*

List of Definitions .... *xix*

1.0 Introduction *1*

1.1 Research context 1

1.2 Runway Utilisation prediction 2

1.2.1 Runway Utilisation prediction requirements 4

1.2.2 Validating ASP enablers 5

1.3 Hypothesis and objectives 7

1.4 Methodology and novelty 8

1.5 AI in aviation industry 10

1.6 Outline of the thesis 12

2.0 Background on Machine Learning and Big Data categories and techniques *17*

2.1 Machine Learning categories 17

2.2 Strategy supervised learning 18

2.3 Machine Learning techniques 19

2.4 Criteria for choosing feasible Machine Learning techniques 21

2.5 Big data techniques 22

3.0 Evaluation of feasible machine learning techniques for predicting the time to fly and aircraft speed profile on final approach *25*

3.1 Introduction 26

3.2 Data, Prediction Variables & Modelling 27

3.2.1 Aircraft Performance & Weather data 27
3.2.2 Prediction variables 27

3.3 Modelling of ML techniques 28

3.4 Context-Machine Learning 29
3.4.1 ML techniques classification 29
3.4.2 Regression techniques to be tested 29
3.4.3 Neural networks techniques to be tested 29

3.5 Pre-Processing 30
3.5.1 Compute T2F and TAS profiles 30
3.5.2 Feature selection 31
3.5.3 Assessibility of PCA 32
3.5.4 Construct the datasets 33
3.5.5 Stability of three different data parts-cross validation 33
3.5.6 Accuracy of data and outliers 34

3.6 Results 35
3.6.1 PCA groupings and RreliefF for A320 at Airport 1 35
3.6.2 Assess feasible ML techniques 37
3.6.3 Relation number of hidden neurons versus MSE 38

3.7 Case Study Results 39
3.7.1 Airport 1 and B738 40
3.7.2 Airport 2 and A320 40
3.7.3 Absolute error results 41

3.8 Conclusion 42

3.9 Recommendations 43

4.0 Taxi-Out Time prediction model at Charles de Gaulle airport 47

4.1 Introduction 48
4.1.1 Related Work 48
4.1.2 Aim 50
4.1.3 Structure 50

4.2 Data and Prediction Variables 50
4.2.1 Operational Taxi Data 50
4.2.2 Prediction Variables 51

4.3 Proposed Methodology 52
4.3.1 Compute the TXOT 52
4.3.2 TXOT Understanding 53
4.3.3 Data Preparation 54
4.3.4 Evaluation of Feasible Machine Learning Techniques 56

4.4 Results Regression Tree Modelling **58**

4.5 Prototype Model **62**

4.6 Discussion on TXOT Results **64**

4.7 Recommendations **65**

4.8 Conclusions **65**

5.0 Predicting Abnormal Runway Occupancy Times and observing related Precursors ***71***

5.1 Introduction **72**

5.1.1 Related Work 72

5.1.2 Aim 74

5.1.3 Structure 74

5.2 Data and Prediction variables **74**

5.3 Methodology **75**

5.3.1 Identification and Understanding of the AROT 76

5.3.2 Data Preparation 81

5.3.3 Combining Feasible ML Techniques 82

5.3.4 Assessing Combined ML Method 83

5.3.5 Observe Risk Precursors with Regression Tree 85

5.4 Real-Time model **91**

5.5 Conclusion **94**

6.0 A machine learning model to predict runway exit at Vienna airport 99

6.1 Introduction **100**

6.1.1 Related work and Interview with Vienna ATCO 101

6.1.2 Aim 105

6.1.3 Structure 105

6.2 Case Study **105**

6.3 Methodology **106**

6.3.1 Final approach and Runway data sources 106

6.3.2 Data preparation 109

6.3.3 Feature selection 111

6.3.4 Suitable ML classification techniques 112

6.3.5 Assessing GB method 113

6.3.6 Observe risk precursors with Gradient Boosting 116

6.4 Prototype model **118**
6.5 Conclusion **120**
6.6 Recommendations **121**
7.0 Validation Runway Utilisation tool ***125***
7.1 RU pre-validation steps **126**
7.2 Validation methodology **128**
7.2.1 RTS objectives 128
7.2.2 RTS scope 128
7.2.3 Solution description 129
7.3 RTS conduct **129**
7.3.1 Environment 129
7.3.2 Traffic 130
7.3.3 Wind Profile Modelling 131
7.3.4 Speed Profile Modelling 131
7.3.5 Separation Scheme 132
7.3.6 Arrival Runway Occupancy Time 132
7.3.7 Tower simulation platform 133
7.3.8 Example of the RU support tool for predicting NREX 133
7.3.9 Conduct of RTS 134
7.4 Results from the V1 real time simulation **136**
7.4.1 Operational needs 136
7.4.2 Operational feasibility and acceptability of the ML RU controller support tool 137
7.4.3 Controller information requirements for a ML RU support tool 139
8.0 Conclusion & Recommendations ***143***
8.1 Conclusion **144**
8.1.1 Objectives 1 and 2 144
8.1.2 Objective 3 147
8.2 Limitations **150**
8.3 Recommendations for further development **150**
About the author ***152***
Associated publications ***153***

List of Figures

Figure 1.1: The EUROCONTROL ORD tool which integrates different conventional constraints impacting runway throughput enhancement solutions. 6
Figure 1.2: Methodology for investigating the ASP enablers. 9
Figure 1.3: the left image illustrates normal practices for a Medium aircraft pair when an infringement is expected. As shown in the right image an alarm should be raised as the ITD could be reduced by 0.5NM between a Medium aircraft pair. 11
Figure 2.1: The supervised learning categories with their corresponding techniques. 18
Figure 2.2: Popular clustering algorithms. 21
Figure 3.1: Example TAS, GS and HW profile versus distance to threshold. 31
Figure 3.2: Normalized feature selection using RreliefF algorithm. 31
Figure 3.3: A bi-plot in two dimensions, to find the relation among different variables. 33
Figure 3.4: MSE versus amount of epochs for 5000 A320 flights. 34
Figure 3.5: R values for test, training, validation and all for an A320 predictive TAS model. 35
Figure 3.6: TAS MLP MSE vs number of Neurons. 38
Figure 3.7: Mean and sigma TAS vs amount of flights for low and strong wind at 1 NM from threshold. 39
Figure 3.8: Maximum absolute error per 0.5NM segment. 41
Figure 4.1: TXOT versus the congestion level for stand C-E32 and runway 26R. 53
Figure 4.2: Normalized feature selection using RreliefF algorithm (see Table 1 for feature/variable definitions). 55
Figure 4.3: Normalized feature selection using Sequentialfs algorithm (see Table 1 for feature/variable definitions). 55
Figure 4.4: Number of TXOT flights (instances) versus errors in minutes for 250,000 TXOT flights. 57
Figure 4.5: MSE in minutes versus amount of epochs for the NNLM technique and 250,000 TXOT flights. 57
Figure 4.6: RMSE versus tree depth for different leaf sizes and features. 61
Figure 4.7: Part of the regression tree with a tree depth of four. 61
Figure 4.8: Distributions of the 61 terminal leaves and the number of flights (PDF, probability distribution function). 62
Figure 4.9: Interface of application. 63
Figure 4.10: Predicted output after running the application for the first trial. 64
Figure 5.1: Example of AROT per aircraft type for runway 08R. 76
Figure 5.2: AROT versus time of the day and different ICAO aircraft categories (H, M, S) for runway 09L, 27R, 08R, and 26L. 78
Figure 5.3: Runway throughput levels versus AROT for different Medium aircraft types on runway 26R. 79
Figure 5.4: Number of flights versus the AROT for different runway exits for Super Heavy (S), Heavy (H), and Medium (M) categories. 81
Figure 5.5: Top 10 features for the AROT using the RreliefF technique for CDG. 82

Figure 5.6: MSE of AROT using the top 10 features. 84
Figure 5.7: Outliers example of abnormal AROT flights for training the model with 10 features. 85
Figure 5.8: Cross-validated error versus minimum leaf size. 87
Figure 5.9: MSE versus tree depth for different leaf size and features. 87
Figure 5.10: Regression tree only for abnormal AROT at runway 08R. The tree shows “what-if” statements. If the statement is right, we go to the upper node; if the statement is wrong, we go to the lower node. 88
Figure 5.11: Interface of application. 93
Figure 5.12: Output after running the application for the first trial. 93
Figure 6.1: VIE Runway Design. The left green arrows in the lower left picture show the possible runway exit utilisations for RWY34. 104
Figure 6.2: Number of cases versus the AROT for RWY34 and runway exits for ICAO categories Heavy, Medium and Light during HIRO. 109
Figure 6.3: Normalized feature selection using *OOBPermutedVarDeltaError* algorithm (see Table 6.2 for feature/variable definitions). 112
Figure 6.4: Error versus minimum leaf size (*lmin*) for Medium aircraft types. 117
Figure 6.5: Predictive NREX visualisations in CAST for RWY34 at Vienna. This example shows that an arrival flight is likely to utilise at exit B2. 119
Figure 7.1: Vienna Runway layout 130
Figure 7.2: Aircraft speed profile model for arrivals 132
Figure 7.3: iTWP interface showing red arrow at RWY exit (B7) indicating the prediction that the approaching aircraft will take the procedural RWY exit for that aircraft (B7) is less 80%. 134
Figure 7.4: iTWP interface showing green arrow at RWY exit indicating the prediction that the approaching aircraft will take the procedural RWY exit for that aircraft (B4) is above 80%. 134

List of Tables

Table 1.1: RU requirements and identified approaches. 5
Table 3.1: Prediction and response variables. 28
Table 3.2: PCA and RreliefF outcome for different amount of flights. 36
Table 3.3: MSE and sigma results for 4 different groups. 36
Table 3.4: Assess feasible ML techniques on different performance indicators. 37
Table 3.5: Applying top 3 feasible techniques on first case study. 40
Table 3.6: Applying top 3 feasible techniques on second case study. 41
Table 4.1: Prediction and target variables. 51
Table 4.2: Comparison of PRU UTXOT with our UTXOT results. 54
Table 4.3: Four feasible ML techniques assessed on their performance indicators for runway 08L 1400 hrs, and 1800 to 2000 hrs. 60
Table 4.4: Average error differences per trial and for different time prediction windows. 65
Table 4.5: RMSE TXOT prediction within approximately 3 and 5 min for six different models. 65
Table 5.1: Prediction and target variables. 75
Table 5.2: Seventeen different abnormal AROT categories with their related precursors. 88
Table 5.3: Average error differences per trial and for the time prediction window 0–30 min and 30–90 min 0–30 min 30–90 min. 94
Table 6.1: Procedural AIP exit[4] at RWY34 for VIE airport. 103
Table 6.2: Prediction variable 1 till 14 and target variable 15. 107
Table 6.3: Forecast and computational time performance for the GB technique. 114
Table 6.4: True classes versus the predicted classes for RWY34 and six different classifications during HIRO. 115
Table 6.5: Example of precursors for a Medium_Non-Procedural flight. 118
Table 7.1: Traffic distribution 131
Table 7.2: Wind characteristics applied in the Safe Clouds RTS. 131
Table 7.3: Average AROT in the SafeClouds RTS. 132
Table 7.4: Procedural AIP exit at RWY34 for VIE airport with associated non-procedural exit. 133
Table 7.5: RTS schedule 135

List of Definitions

Term	Definition
Abnormal behaviour	Abnormal behavior is any behavior that deviates from what is considered normal. The abnormal behavior threshold value for each (Aircraft Safety Parameter) ASP enabler is shown at the ASP risk definition.
Accurate ASP enablers predictions	An ASP enabler statement about what will happen in the future. The following Mean Squared Error (MSE) or accuracy values have been applied in this study to judge if a model is accurate: • Time to Fly (T2F) MSE < 15 seconds • Taxi-Out Time (TXOT) MSE < 2 minutes • Arrival Runway Occupancy Time (AROT) MSE < 6 seconds • Runway Exit Use (NREX) accuracy > 80%
AROT	The Arrival Runway Occupancy Time. The time when the aircraft is above runway threshold, until the time the aircraft is clear of the runway.
ASP enablers	Aircraft behaviour that impacts runway throughput and safety, as defined by ATCO; the parameters considered in this study are AROT, TXOT, NREX, T2F and TAS.
ASP pattern	Pattern is an underlying structure that organizes structures in a consistent, regular manner. AROT, NREX and TXOT patterns are defined by the regression or classification tree, which gives, per terminal leaf, a distribution describing the relation between an ASP enabler and the precursor features. These patterns and precursors can be transferred into "what-if" statements by analysing the relations between the ASP enabler and the precursors. A pattern for the ASP enablers T2F and TAS are defined as the normal distributions (from –2 to +2 sigma) of T2F and TAS for a given flight and for a range covering the last 10NM of the final approach.
ASP precursor	A precursor is one that precedes and indicates the approach of another. A metrological or aircraft feature that precedes and indicates the approach of an ASP pattern or risk.
ASP risk	The likelihood that the potential for the accident or the incident will be realized[1]. An ASP risk is considered when the following threshold values are met, including an *accurate* prediction during HIRO: • The minimum time separation between an aircraft pair is smaller than the minimum ICAO Separation per aircraft

[1] International Civil Aviation Organization (ICAO) Doc 9859

	pair[2]. This separation scheme is translated back from distance to time (T2F) during low wind conditions (< 5kts headwind)[3] • The abnormal TXOT is higher than 25 minutes • The abnormal AROT is higher than 2σ standard deviation from the normal distribution mean • The aircraft misses its intended runway exit as defined by the Aeronautical Information Publication (Table 6.1)
ASP safety	The condition of being protected from or unlikely to cause risk. The predicted TXOT, AROT, NREX, T2F and TAS enable the ATCO to anticipate the prediction. This study contributes to the following aspects of safety: • AROT and NREX, to request the leader to expedite an exit earlier or request the follower to apply speed deceleration, which could lead to a potential avoidance of an infringement or go-around • TXOT, to have a more accurate holding point time estimate during mix-mode operations • T2F/TAS, to request the follower to apply speed deceleration, which could lead to a potential avoidance of an infringement or go-around
ATCO	Air Traffic Control Officer
Clear *visualisation*	The act of visualizing something. Showing accurate ASP enabler predictions next to the airplane and on the Human Machine Interface (HMI) final approach; showing the value only on the HMI when the following criteria are met: • *fast* and *accurate* ASP prediction • when an ASP enabler exceeds a threshold (ASP risk)
Effectively	In such a manner as to achieve a desired result. In this study it is the achievement of the Runway Utilisation (RU) requirements.
Fast ASP predictions	A prediction that is happening quickly. The quickest real time prediction should be the shortest time possible to take a useful ATCO action using it. A *fast* solution has less than five seconds between the prediction and the ATCO action.
Feasible machine techniques	Is capable of being done or carried out. The machine learning (ML) techniques proposed are based on the criteria in Section 2.4 and the potential of fitting the RU requirements.

[2] International Civil Aviation Organization (ICAO) Doc 4444

[3] Herrema, F., Curran, R., Zhao, W., Treve, V., & Graham, R. (2015). Time Based Separation: A study into runway compression and time based separation. In *15th AIAA Aviation Technology, Integration, and Operations Conference* (p. 2430).

Infringements	An aircraft pair on final approach where the minimum required distance is not maintained.
Intuitive and *interpretable* ASP predictions	To understand the driving precursors and explain these precursors during a similar situation; explaining and showing driving precursors when an ASP enabler exceeds a threshold (ASP risk); the threshold values during high intensity runway operations are presented in the ASP risk definition.
NREX	Procedural or non-procedural runway exit taken.
Operations	ATCO's performance of a practical work.
Real-time (operations)	Serve real-time applications that process data as it comes in. In this context, the ATCO support-decision tool aims to make *fast* ASP predictions for tactical decision-making. The data should be processed within five seconds such that it is immediately available virtually as feedback for the ATCO with regards to the process from which it is coming.
RU requirements	*Fast, accurate, intuitive, interpretable* predictions through feasible *visualisation*.
Runway throughput	It is a measure of the capacity of a runway. It defines the average movements (both arrival and departure) that can be performed in an hour's time.
RU support tool	A computer program application that analyses data and presents it so that users can make decisions more easily. In this study it is considered as the supplementary operational ATCO support decision tool that assists the ATCO in predicting ASP enablers in consideration of the identified Runway Utilisation (RU) requirements.
Strategic decision-making	It involves the span of the next one or two hours from the moment of prediction and can be used by the ATCO supervisor to decide on changing aircraft pairs of the final approach sequence.
T2F	Time to fly till runway threshold.
Tactical decision-making	It is performed over several seconds, enabling ATCOs to be warned about any impending runway capacity issues.
TAS	True airspeed on final approach.
TXOT	The time elapsed between the off-block time and the take-off time.

1.0 Introduction

Many of today's major airports are often unable to handle the traffic demand. The busiest airports are already saturated, and there are political and environmental issues associated with further airport expansions. Considering the expected further growth in air traffic demand [1, 2] (doubling in the next 20 years), there is an urgent requirement for runway capacity improvements in a safe and environmentally responsible manner.

Safety and runway throughput capacity is, to a large extent, determined by the wake vortex separation criteria applied during instrument operations. Air Traffic Controllers (ATCOs) are responsible for applying these strict separation criteria. Similarly, they are responsible for the planning and spacing management to optimise the runway throughput. Separation management and standards are crucial for the efficient use of airspace resources and efficient airport operations [3].

1.1 Research context

Currently, at many airports, the runway throughput is the limiting factor for the overall capacity. Among the crucial parameters limiting the arrival flow at airports are the wake turbulence separation minima expressed in distance and the uncertain speed variations in speed profile between two successive arrivals on final approach [4]. For dealing with these limitations, ATCOs apply a buffer based on training and experience for ensuring minimum separation.

The size of the applied buffer does not only depend on the *speed* and *time to fly* profile but also on the *runway exit utilised* or the *arrival runway occupancy time* of the lead aircraft of two successive arrivals [5, 6]. In so-called mixed-mode operations, an ATCO may want to insert a departure flight between two successive arrivals when the gap between the two arrivals is sufficiently large to permit this. To be able to accomplish this, an ATCO needs an accurate estimate of the time at which the departing aircraft reaches the runway holding position. More accurate holding point time estimates can be realised through a better understanding and prediction of the *taxi-out* time. Research has shown [5] that the aforementioned experience can be quantified and thus predicted to facilitate optimum operations. This experience and terms highlighted in italics are covered by the name Aircraft Safety Parameter (ASP) enablers.

The existing arrival wake vortex separation minima, usually expressed as fixed distance values, are generally considered to be over-conservative [7]. At many hub airports these fixed separation values depend only on the aircraft weight categories as defined by the International Civil Aviation Organisation (ICAO): Heavy, Medium and Light.

Currently, different wake vortex separation rules are applied during the final approach that are typically expressed in terms of distance. In the coming years, distance-based separation is expected to be gradually replaced by time intervals and/or speed compensation at airports where strong wind conditions apply. The problem is that each aircraft of a (leader-follower) pair flying on final approach differs in speed, causing the separation between two succeeding aircraft to either increase or decrease. Decreasing separation leads to an inherent safety risk due to the phenomenon of wake turbulence.

To optimise runway throughput, it is not only necessary to refine the separation criteria, as is the objective of dynamic and flexible separation concepts, but also to better understand and avoid operational risks. Providing a support system to aid the controllers in their separation management and assurance tasks has the potential to yield significant benefits in terms of improved throughput efficiency. Such concepts are currently being developed in the SESAR (Single European air traffic management (ATM) Research) program. SESAR is a collaborative research programme oriented to completely overhaul European airspace and its ATM [8].

An example of a refined separation criteria is the new, more complex concept for reduced separation minima for aircraft pairs – the European Wake Vortex Re-categorisation (RECAT-EU). RECAT-EU offers a more refined categorisation of aircraft types than the traditional ICAO approach. It aims at safely increasing airport capacity by redefining wake turbulence categories and their associated separation minima. It divides the current Heavy and Medium categories into two sub-categories, e.g. creating a new Super Heavy category for the Airbus A380 [4].

The development and refinement of wake vortex separation rules is sensitive to the dynamic influences of wake behaviour [7]. For instance, ATCOs [9] suggested that fixed over-conservative separation values such as RECAT-EU could be refined by addressing additional ASP enablers for identifying precursors and avoiding accident or incident risk. Therefore, proper ASP prediction is required to further avoid incidents and reduce spacing uncertainty.

1.2 Runway Utilisation prediction

Taking ASP enablers and precursors into account might help ATCOs to make better separation assurance, resulting in potentially safer operations and probably also higher capacity in certain weather conditions. This could potentially provide significantly more efficient spacing criteria in lieu of the worst-case criteria currently used but without increasing the risk associated with wake encounters.

Currently, ATCOs make use of Arrival Manager (AMAN) and Departure Manager (DMAN) tools. AMAN systems provide automated sequencing support for approach and runway ATCOs, while continuously optimising arrival traffic sequences and runway slot times for landing aircraft. This is accomplished by a more efficient and predictable arrival management process that can assist in reducing low-level holdings and tactical intervention by the ATCO. AMAN considers the locally defined maximum landing rate (capacity), the required separation standards for aircraft in the touchdown zone and additional operational criteria. DMAN is an advanced controller tool for optimising runway throughput. To achieve optimal use of runway capacity and airspace capacity in the Terminal Management Area (TMA), a DMAN assists the ATCO in managing departure traffic by providing optimised take-off sequences when considering departure trajectories. AMAN and DMAN tools are essential controller aids that provide guidance and ensure the best use of the available runway capacity (i.e. maximum throughput). Both tools can provide the controller with advice on tactical or

strategic runway capacity decisions but without the associated precursors. Tactical decision-making is performed over a horizon of several seconds, enabling ATCOs to be warned about any impending runway capacity issues. Strategic decision-making involves a horizon of the next 1 or 2 hours from the moment of prediction and can be used for the ATCO supervisor to decide on changing aircraft pairs of the final approach sequence.
Based on the current AMAN/DMAN state-of-the-art and ATCO interviews [9], research requirements were identified as part of this work and are highlighted below in italics in brackets.

An additional support tool providing real-time alerts (*fast* predictions) is expected to be an advantage if not a necessity in a future environment of High Intensity Runway Operations (HIRO), in which the associated risk of a loss of separation between aircraft has a direct negative impact on safety and accident and incident avoidance. *Fast* predictions are only useful if the ATCO has enough time to take an action during real-time HIRO.

Therefore, ATCOs operating at hub airports are moving towards proactive risk management [10], which aims to identify, understand (*intuitive*) and predict ASP risk and precursors (*interpretable*) to mitigate the risks associated, thereby avoiding accidents or risking incidents during HIRO. *Intuitive* and *interpretable* decisions are useful when an ASP enabler exceeds a certain threshold. For example when a loss of separation between two A380s lower is than 120 seconds, resulting in a negative impact on safety. To increase the accuracy of the spacing between aircraft, the development of an ATCO support tool to alert the ATCO (clear *visualisation*) with (*accurate*) predicted ASP enablers that impact runway throughput and safety is considered as necessary [9].

Currently, there is no supplementary operational system that assists the ATCO in predicting ASP enablers on the runway or on final approach, considering the five identified requirements, highlighted in italics in brackets. Therefore, this study will develop an arrival ATCO support decision tool named RU which stands for **Runway Utilisation** support tool.

Considering the urgent requirement for a *fast*, *accurate*, *interpretable* and *intuitive* model to the creation of the RU support tool, we can now formulate the primary research question:

How to identify and analyse runway utilisation requirements, runway-throughput and safety to extract ASP patterns, risks and precursors on the runway and final approach in order to model and support tactical and strategical decision-making and alerting solutions?

1.2.1 Runway Utilisation prediction requirements

RU requirements can be formulated for working with the envisaged RU tool during real-time operations with the following four requirements being the most important ones:

1) Making *fast* ASP predictions, allowing ATCO to quickly decide and anticipate the leader or follower aircraft during real-time operations. The quickest real time prediction should be the shortest time possible to take a useful ATCO action using it. A *fast* solution [9] has less than five seconds between the prediction and the ATCO action.
2) Making *accurate* ASP predictions, allowing ATCO to rely on these enablers and making trustworthy decisions for the leader or follower aircraft. Accuracy is a measure of how well samples are classified to the correct category. Accuracy is one metric for evaluating classification models. In regression analysis, it is a measure of how well the model predicts the response variable. The List of Definitions show the accuracy or regression values per ASP enabler.
3) Being *intuitive* and *interpretable*, which allows ATCO to understand the driving precursors and understand and explain these precursors during a similar situation. Explaining and showing driving precursors when an ASP enabler exceeds a threshold (ASP risk) and when the model is *accurate*. The threshold values during high intensity runway operations can be found in the List of Definitions.
4) Clear *visualisation.* Showing accurate predictions next to the airplane and on the final approach Human Machine Interface (HMI). Showing ASP value only on the HMI when the following criteria are met: 1) *fast* and *accurate* ASP prediction and 2) when an ASP enabler exceeds a threshold (ASP risk).

Different approaches can be considered to address the aforementioned requirements and to develop the RU tool. Predicting ASP enablers and aiming at the RU requirements are the ingredients when developing the RU tool. The ASP enablers are predicted based on historical runway and final approach aircraft performance data under wind uncertainty (Chapter 3–6). The data that is utilised within this study has unknown ASP patterns and precursors [11], making it a stochastic problem [12]. Therefore, three approaches were selected based on the data used within this study for predicting ASP enablers. The following stochastic approaches are proposed; data mining [13], optimisation methods [14] and artificial intelligence [15].

Table 1.1 shows the RU requirements on the upper row and suitable approaches on the first column.

Table 1.1: RU requirements and identified approaches.

	Fast	*Accurate*	*Interpretable* and *intuitive*	Clear *visualisation*
Data mining methods				
Optimisation methods				
Artificial Intelligence methods				

The following explains the relevance of each and the colouring in Table 1.1.

<u>Data mining</u> refers to the process of discovering patterns in large historical data sets [13]. We use data mining analysis for knowledge discovery. It is expected that by analysing historical weather, airport and aircraft performance data, a better understanding and prediction of ASP enablers will be realised. However, data mining cannot merge and clean data sets [16]. Furthermore, it does not permit *interpretable* and *fast* predictions [16].

<u>Optimisation methods</u> aim to generate *fast* and *accurate* solutions as the goal of optimisation methods is to find an optimal or near-optimal solution with low computational effort [17]. The effort of an optimisation method can be measured as the computational time and computer memory required by the method. For many optimisation methods, there is a trade-off between solution quality (*accurate*) and effort (*fast)* as with increasing effort, the solution quality increases [17, 18]. The most commonly used optimisation methods are decision rules and Heuristics which provide *interpretable* and clear *visualisation* results [17].

<u>Artificial Intelligence (AI)</u> is the simulation of human intelligence processes by machines, especially computer systems. These processes include: the acquisition of information and rules for using the information (learning), using rules to reach approximate or definite conclusions (reasoning) and using or creating rules when additional data is fed to the model (self-correction). The rapid evolution and adoption of AI analyses in various industries (such as aviation) has led to more efficient AI analytical methods for improving efficiency in operations [19]. The data used to design the RU tool is stochastic, making it suitable for AI. AI is the only method that covers all RU requirements: clear *visualisation*, *accurate*, *fast*, *intuitive* and *interpretable* [20] predictions.

1.2.2 Validating ASP enablers

The RU support tool developed in this thesis needs to be tested and validated on operational needs, operational feasibility/ acceptability and RU requirements before it can deployed at hub airports.

The ATCO Real Time Simulation (RTS) and ORD (Optimised Runway Delivery) tool developed by EUROCONTROL is used to validate the predicted ASP enablers. The EUROCONTROL tool is the first ATCO system vision in Europe [5] that integrates and tests the operational feasibility of different runway throughput enhancement solutions. It provides a dynamic application for separation and safety indicators that enable consistent and efficient delivery of the required separation or spacing between arrival pairs on final approach for the runway landing threshold. Different throughput solutions such as Time-Based Separation

(TBS), have already been tested in the ORD tool before being deployed and operational at London Heathrow and Vienna airports [21].

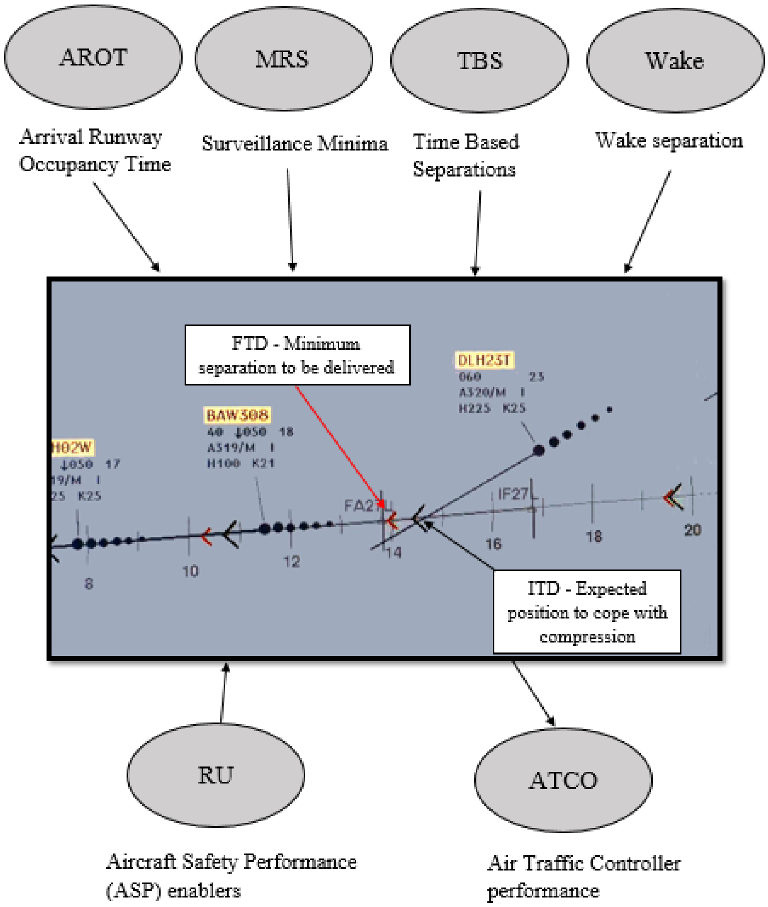


Figure 1.1: The EUROCONTROL ORD tool which integrates different conventional constraints impacting runway throughput enhancement solutions.

Figure 1.1 presents the system layout for the ORD tool. The flight schedule and the initial landing sequence is provided to the ORD tool based upon the order of the predicted landing times. The tool computes the Target Distance Indicators (TDIs) for each aircraft pair when the required spacing criteria are met and displayed on both the Approach and Tower ATCO working positions. The TDIs comprise a Final Target Distance indicator (FTD) and an Initial Target Distance indicator (ITD). The FTD calculation represents the minimum required separation or spacing depending on the most constraining factor (e.g. TBS and Wake separation, Surveillance Minima (MRS), Arrival Runway Occupancy Time (AROT), or a gap inserted by the final approach ATCO) to be applied at the point of separation delivery, in this case, the runway threshold. Currently, the employed fixed over-conservative separation values do not consider ASP enablers as defined by ATCOs [9], thereby lacking ASP predictability.

The necessity for developing the RU tool and validating accurate ASP enablers using an RTS tool (such as ORD) allows ATCOs to anticipate increased air traffic demand and events [9] that impact the runway throughput or safety concerns.

1.3 Hypothesis and objectives

We identified several requirements in Section 1.2 that need to be fulfilled to integrate multiple operational improvements through a single RU support tool. The ATCO support tool can become a realistic decision-making tool if the following hypothesis can be proved valid for each ASP enabler:

ML can be used to effectively identify ASP patterns, risks and precursors resulting in the extraction of RU requirements for tactical and strategic decision-making.

Before we can address the hypothesis for each ASP enabler, key objective 1 must first be addressed:

1. To assess ML techniques, their different features and the amount of data to identify ASP enablers

Here we identify applicable and feasible ML techniques and lessons learned with regard to collecting, combining, processing and analysing data in ATM and other domains. Through assessment and modelling, many core ML techniques, different data samples and sources need to be covered. It is crucial to understand which ML techniques are feasible for addressing different ASP risks (e.g. unnecessarily large spacing on final approach, missed runway exits, abnormal AROTs or TXOTs), using different data set sizes and sources. Different features impacting the runway were used and different ML techniques were assessed with respect to their forecast performance, computational time and amount of data required for an accurate prediction.

After assessing objective 1, the hypothesis is addressed (objective 2), which will be presented in Chapter 8.1. For each ASP enabler, it was determined whether the RU requirements highlighted in Section 1.2 can be extracted using ML.

2. To address the testing of the hypotheses for each ASP enabler identified

Literature and interviews indicate that identifying patterns, risks and precursors from historical and real-time information has never before been implemented mathematically in a support decision tool for the ATCO. Therefore, this work will focus on tactical and strategic predictions and decision-making.

- Tactical prediction is performed over a horizon of several seconds from the moment of prediction, enabling tactical operation tools to be fed and ATCOs to be warned about certain events along with their associated precursors.
- Strategic prediction involves a horizon of 1 or 2 hours and can be used either as a strategic operational tool for the ATCO supervisor to decide on the final approach sequence, or can be used as a strategic input for the RTS ORD tool.

Objective 3 can only be addressed if the hypothesis is true for that corresponding ASP enabler. Key objective 3 can now be formulated:

3. To validate the RU requirements and operational feasibility and acceptability of the RU tool in the RTS simulator

The following validation process will need to be adaptable to meet the requirements (identified in Section 1.2) for the RU support tool:

- Ensuring *accurate*, *fast*, *intuitive* and *interpretable* ASP predictions provided through suitable *visualisation*. These predictions are valuable assets for ensuring a safer and efficient runway throughput.
- Assessing the performance of the RU system through the following set of indicators: operational needs and operational feasibility and acceptability.

1.4 Methodology and novelty

This work contributes to improving safety and making better separation decisions for aircraft pairs by addressing ASP enablers as identified in [5] and defined by ATCOs [9]. Three independent ATCOs from different hub airports were interviewed. The following ASP enablers were considered most significant due to their subsequent impact on runway throughput operations:

(1) Time to Fly (T2F) and True Airspeed (TAS) leading to a better characterisation of large spacing's or infringements (Chapter 3);
(2) (abnormal) Taxi-Out Times (TXOT – Chapter 4);
(3) (abnormal) Arrival Runway Occupancy Times (AROT – Chapter 5) and
(4) Procedural and non-procedural runway exit used (NREX – Chapter 6).

These ASP enablers depend on two key precursors. The first precursor is the impact of the prevailing meteorological and airport conditions [19, 30], while the second precursor involves the runway congestion and decay of the wake turbulence [31, 32].

The focus of the research methodology is to identify and understand RU requirements and ASP patterns, as well as to show ASP precursors and risks impacting runway throughput and safety. To identify these patterns, risks and precursors, feasible ML and BD techniques were used. The ASP enablers were assessed through the three objectives highlighted in Section 1.3. An overview of the methodology is illustrated in Figure 1.2 below;

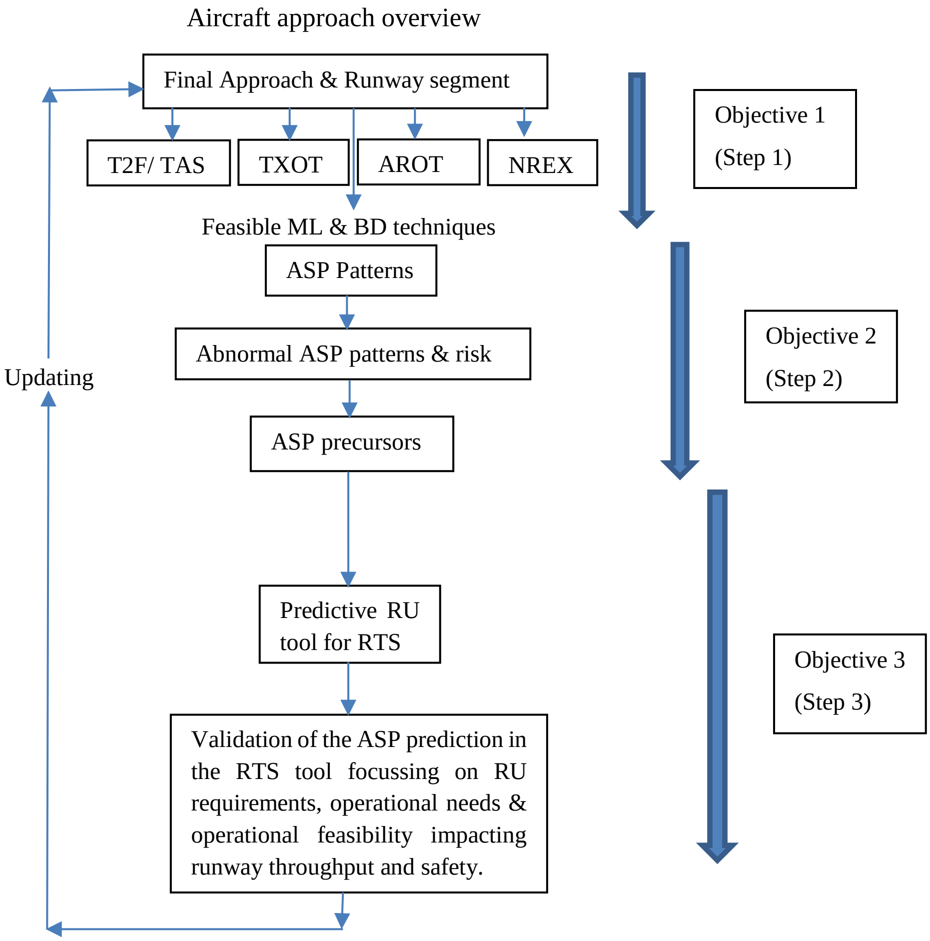


Figure 1.2: Methodology for investigating the ASP enablers.

In this dissertation, we apply the methodology highlighted in Figure 1.2 to investigate the ASP enablers: NREX, TXOTs, AROTs, T2F and TAS. The methodology begins with Step 1 in which we review and assess predicted ASP enablers regarding data, problems and ML techniques. In Step 2, ASP patterns, abnormal behaviour, risk and precursors are identified using feasible ML and BD techniques. In Step 3, a Validation on RU requirements, operational needs and operational feasibility is executed using EUROCONTROLs ATC RTS tool. Finally, we close the loop and return to Step 1 by adding additional data and updating ASP patterns and risks.

Novelty

This dissertation significantly advances the current practice of ATCO decision-making support on final approach. It specifically addresses the suitability of ML for improving ASP predictability using historical data and precursors. No known work has previously undertaken the task of developing an RU support tool for ATCO. The uniqueness of the ML field with the use of ASP precursors, abnormal historical data and applying real-time support tools to the problem resulted in a novel solution tool. The novelty of this thesis can be summarised as follows:

1) The suitability of ML for improving ASP enabler predictability using historical data:

   By addressing key objective 1, a better understanding of the ML suitability will be obtained. ML techniques have been assessed for each ASP enabler with respect to their different features and the amount of data required.

2) The real-time feasibility of ML being used:

   The computational time is assessed for each ASP enabler (key objective 1). Only accurate predictions with low computational time can be validated in the RTS and used during real-time operations.

3) Using abnormal historical data and precursors to further improve predictability

   Novelty statement 3 is part of validating the hypothesis. Abnormal historical data and precursors are extracted using ML. For each ASP enabler, it is observed whether the predictability increases with the use of this data.

1.5 AI in aviation industry

In Section 1.2.1, it was explained why AI is the best approach for addressing the ATCO requirements: clear *visualisation*, *accurate*, *fast*, *intuitive* and *interpretable* predictions. The following Section elaborates on the feasibility of using AI methodology. The efficient deployment of the RU tool for ATCOs requires a reliable ASP prediction and clear visualisation of intuitive and interpretable precursors impacting the runway. This study will promote such a deployment through a better understanding of the mentioned ASP enablers, leading to safer and more efficient spacing.

AI can be divided into two sub-domains: Big Data (BD) and Machine Learning (ML). In this context, BD and ML can be used to identify patterns in previous data [22], leading to specific ASP enabler shortfalls on the runway. These ASP enablers can be predicted during real-time operations but also for strategic decision-making.

Due to the explosion in the capacity to acquire, store and analyse sizeable datasets with ML in recent years, analytical models are gradually being replaced by powerful data analysis solutions in most industries. This new technology yields notably reliable results in optimising scenarios where there are many factors that influence the value of an ASP enabler, such as the time to fly of an aircraft on final approach [23]. ML is especially beneficial when the relationship between these factors and the predicted ASP enablers is unknown and complex [23].

Over the past few years, the opportunity has arisen at different hub airports to assess an increasing amount of short-term evolution of meteorological parameters and historical aircraft and airport performance parameters to enhance the expected prediction of ASP enablers using ML. As ML techniques are standardly available in many BD libraries and because of significantly faster computational time and lower costs, there is a definite interest to assess historical data growth to provide accurate ASP predictions.

Such predictions would alert ATCO members regarding forthcoming ASP issues to transform these issues into understandable precursors for tactical or strategical decision-making and to propose specific solutions for these safety issues that impact runway throughput. An alarm flag should be raised when an ASP risk is expected.

The illustration below in Figure 1.3 shows an ATCO alert example of the T2F or the TAS of the follower. As the follower is flying faster (as the leader slows down earlier) and due to the different speed profile characteristics per aircraft type, an infringement for this aircraft pair is expected. For these situations the ATCO normally takes a too conservative ITD into account [9]. Therefore the tool predicts that an ITD of 0.5NM could be applied to allow safe distance separations, instead of the 1NM ITD initially applied.

The T2F and TAS are continuous variables that are computed depending on variables such as aircraft type, the airport and weather conditions. For this example, ML and BD can be used to identify patterns and to observe precursors leading to better T2F and TAS prediction. The ML techniques will be addressed by modelling the predicted T2F and TAS values under different weather conditions and at specific locations before the runway threshold.

For choosing an appropriate feasible ML technique for assessing ASP enablers such as T2F and TAS, first the ML category with which the technique has to comply should be identified. The different ML and BD categories and techniques are elaborated in Chapter 2.0.

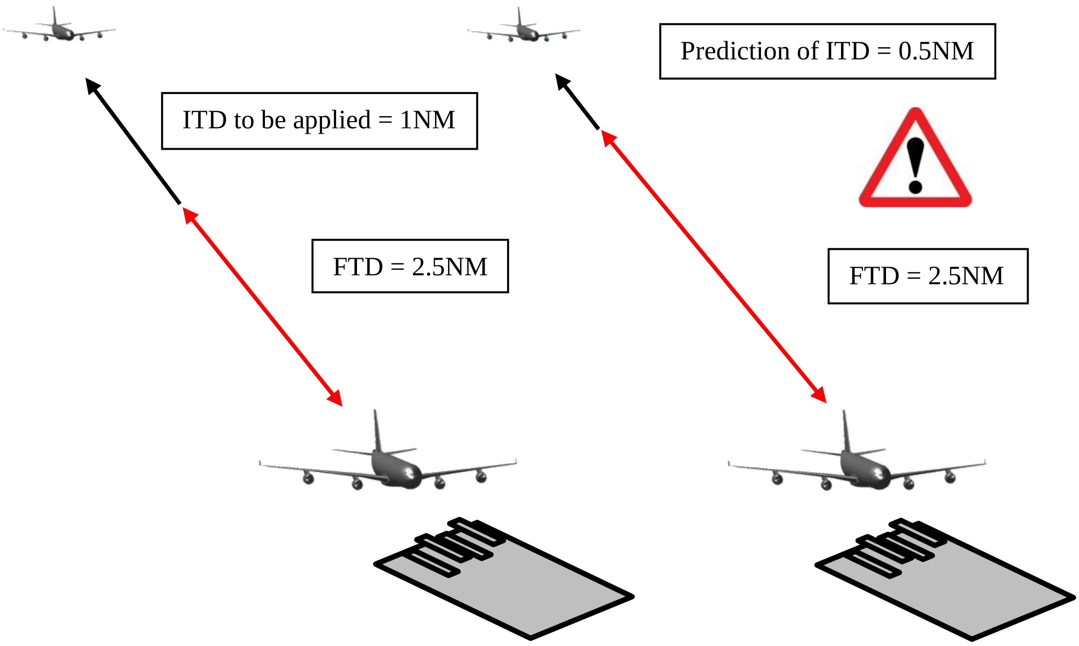


Figure 1.3: the left image illustrates normal practices for a Medium aircraft pair when an infringement is expected. As shown in the right image an alarm should be raised as the ITD could be reduced by 0.5NM between a Medium aircraft pair.

When considering an AI approach, first the current studies and work previously completed in aviation research were explored. AI techniques have been successfully applied in many aviation related domains. The authors of [24] proposed the use of the Tabu Search algorithm to solve the combinatorial Aircraft Conflict Detection and Resolution problem. They reported up to 23% improvement with respect to the Branch and Bound approach on data from Fiumicino airport, but their technique was unable to determine a conflict-free sequence in some situations. Their results were better on Malpensa airport data without any scheduling issue. In [25], the authors proposed the use of Ordinary Least Square (OLS) regression to extract wind parameters (direction and speed) from aircraft radar data. They assessed the accuracy of their technique using Meteo-France data compared to wind parameters extracted from two trajectory datasets (Mode-C radar data from the Paris area and Mode-S radar data from the Toulouse area). The study [26] used Random Forests to predict turbulences associated with thunderstorms, one of the most significant causes of weather delays. Their approach outperformed the basic storm distance and the Graphical Turbulence Guidance (GTG) product. More recently, [27] and [28] successfully used Gradient Boosting Machines (GBM) to predict the mass and air speed of aircraft during climbing. In [28], the researchers mixed predicted mass and (calibrated air speed, M) speed profiles in conjunction with the Base of Aircraft Data (BADA) performance model to predict the future trajectory (altitude) of an aircraft during climb within a 10-minute horizon. They claimed an improvement of at least 36% in the airspeed estimates using their GBM-based approach instead of the reference BADA profiles. Similarly, an improvement of at least 45% on the future altitude prediction task was reported. It is noteworthy that even with very accurate speed estimates, the altitude estimates might remain inaccurate. According to the authors, this might be due to errors in the weather model and/or the BADA performance model, particularly the max climb thrust setting approximation or inaccurate mass estimations. Another interesting example is the BagTrack project funded by The Danish Advanced Technology Foundation. This project aims to improve baggage handling quality using Radio Frequency Identification (RFID) baggage tracking data. In [29], the authors leveraged decision tree classifiers to identify potential issues in baggage management.

1.6 Outline of the thesis

To address the aim and research objectives and proof the hypothesis, an identification and mock-up of suitable ML techniques was performed as presented in Chapter 2 for finding RU requirements, runway throughput and ASP patterns on the runway and final approach. Chapter 3 serves as a background chapter on the topic. Chapter 4 is a journal article published in a peer-reviewed management journal. Chapters 5 and 6 were published in peer-reviewed journals. Chapter 7 is awaiting a decision for acceptance by a peer-reviewed journal. Each of the previously published or publication-pending articles has been reproduced here in their original format so that they can be read independently. The chapters are described in more detail below.

Chapter 2

Chapter 2 presents existing ML and BD algorithms. First, we introduce ML categories, strategies and techniques. Second, the criteria for selecting suitable ML techniques are presented. In Chapters 3 to 6, the suitable techniques identified will be assessed on their forecasting performance, computational time and the amount of data needed for delivering a reliable ASP prediction.

Chapter 3

In Chapter 2, valuable insights were obtained with regard to feasible ML techniques for ASP enablers. These insights were used in our first ML model (Chapter 3). Chapter 3 presents how the ML techniques might be used for predicting the T2F and TAS profiles on final approach. Different ML techniques were assessed on their forecasting performance, computational time and the amount of data needed for delivering a reliable prediction. These techniques were applied to the traffic of two different major European airports and were benchmarked against the ORD tool using a statistical approach for deriving the T2F and TAS. Consequently, the most efficient ML techniques were applied on the two case studies for predicting the T2F and TAS.

Chapter 4

Chapter 4 uses the ML techniques that were considered suitable along with the amount of data required. Chapter 4 focuses on how TXOT can be predicted by using the neural networks, regression tree, reinforcement learning and multilayer perceptron methods. These four methods were assessed based on their performance indicators and were applied to the Charles de Gaulle airport operational taxi data and benchmarked against real-life TXOT profiles. The root-mean-squared error metric was selected as the essential performance indicator. The regression tree appeared to be the most efficient method which was then applied in a case study for predicting the TXOT and finding the key related precursors extracted from the top 10 features. The TXOT prediction ASP enabler was used as a first input for the RU support tool.

Chapter 5

In Chapter 4 insights involved extracting patterns, risk and precursors using regression trees. The tree allows ATCOs to anticipate historical safety risks and how to avoid them in the future by understanding the extracted precursors. In Chapter 5, this knowledge is extended by using multiple trees in a real-time operational environment. A real-time ML model was developed in which the existing ML trees were combined for predicting the abnormal AROTs of unique radar data patterns. The regression tree, which was the best performing method, was used in this study to observe the key related precursors extracted from the top 10 features. The abnormal AROT prediction ASP enabler was used as a second input for the RU support tool.

Chapter 6
This chapter presents a ML method building upon experience with the previous methods investigated in Chapters 4 and 5. Particularly, Chapter 5 demonstrated the extraction of abnormal behaviour and patterns and how the same can be used for real-time tactical and strategical decision-making. This knowledge was used for extracting the NREX. As identified in Chapter 3, classification ensembles are particularly suitable for predicting this third ASP enabler**.** In Chapter 6, the existing classification ensembles are assessed on their forecast performance and computational time for predicting the NREX. Tests were conducted using runway and final approach radar data at Vienna airport.

Chapter 7
Finally, a validation exercise with EUROCONTROL and proof of concept was performed based on the use of feasible ML and BD technologies. The two best performing models – abnormal AROT from Chapter 5 and NREX from Chapter 6 – were validated using the EUROCONTROL ATC RTS tool. This ensured the best possible use of the existing safety data to enhance ASP risk identification and risk assessment at European level. The final RU support tool should include a software infrastructure that provides the computational power required to receive and analyse the data managed from other airports. The final tactical and strategical predictive RU tool updates patterns and can show alerting issues and support decision-making for Vienna airport.

References

[1] Jackson, M.R., Sharma, V., Haissig, C.M. and Elgersma, M., 2005. Airborne technology for distributed air traffic management. *European Journal of Control*, *11*(4–5), pp.464–477.

[2] Schnell, M., Epple, U., Shutin, D. and Schneckenburger, N., 2014. LDACS: Future aeronautical communications for air-traffic management. *IEEE Communications Magazine*, *52*(5), pp.104–110.

[3] Malik, W., Gupta, G. and Jung, Y., 2010, August. Managing departure aircraft release for efficient airport surface operations. In *AIAA Guidance, Navigation, and Control Conference* (p. 7696).

[4] EUROCONTROL. (2015). *RECAT-EU, optimising Europe's airports capacity.* Available at: https://www.eurocontrol.int/articles/recat-eu. Last accessed 24th Jan 2019.

[5] Herrema, F. F., Treve, V., and Curran, R. (2015, September). Typical additional spacing-buffer to apply at 4DME for delivering separation minima. In *Digital Avionics Systems Conference (DASC), 2015 IEEE/AIAA 34th* (pp. 1F1–1). IEEE.

[6] Ellejmi, M. Technical feasibility and impacts of reducing standard separation minima in final approach. In: *2018 Integrated Communications, Navigation, Surveillance Conference (ICNS)*. IEEE, 2018. p. 1–40

[7] Gerz, T., Holzäpfel, F. and Darracq, D., 2002. Commercial aircraft wake vortices. *Progress in Aerospace Sciences*, *38*(3), pp.181–208.

[8] Baumgartner, M. and Finger, M., 2014. The single European sky gridlock: A difficult 10-year reform process. *Utilities Policy*, *31*, pp. 289–301.

[9] Wachter, P. (2018) Interviewed by Floris Herrema for SafeClouds, 2 November. Available at: www.safeclouds.eu (H2020 deliverable) (Accessed: 17 January 2020).

[10] Karikawa, D., Aoyama, H., Takahashi, M. and Furuta, K., 2013. A visualization tool of en route air traffic control tasks for describing controller's proactive management of traffic situations. *Cognition, technology & work*, *15*(2), pp. 207–218.

[11] Nanduri, S.K.A., 2016. Anomaly detection in aircraft performance data.

[12] Matsuno, Y., Tsuchiya, T., Wei, J., Hwang, I., 2015. Stochastic optimal control for aircraft conflict resolution under wind uncertainty. *Aerospace Science and Technology*, *43*, pp. 77–88.

[13] Shapiro, A., Dentcheva, D. and Ruszczyński, A., 2009. *Lectures on stochastic programming: Modeling and theory*. Society for Industrial and Applied Mathematics.

[14] Chang, Y., 2010. *Stochastic programming approaches to air traffic flow management under the uncertainty of weather*. PhD Thesis. Georgia Institute of Technology.

[15] Defourny, B., 2010. *Machine learning solution methods for multistage stochastic programming*. PhD Thesis. Ph. D. thesis, Institut Montefiore, Université de Liege.

[16] Longadge, R. and Dongre, S., 2013. Class imbalance problem in data mining review. *arXiv preprint arXiv:1305.1707*.

[17] Rothlauf F., 2011. Optimization methods. In: *Design of Modern Heuristics. Natural Computing Series*. Springer, Berlin, Heidelberg.

[18] Fletcher, R., 2013. *Practical methods of optimization*. John Wiley & Sons.

[19] Herrema, F.F., Treve, V., Curran, R. and Visser, H.G., 2016, June. Evaluation of feasible machine learning techniques for predicting the time to fly and aircraft speed profile on final approach. In *International Conference on Research in Air Transportation* (Vol. 8, No. 4, pp. 4–8).

[20] Molnar, C., Giuseppe C., and Bernd B., 2019. Quantifying interpretability of arbitrary machine learning models through functional decomposition. *arXiv preprint arXiv:1904.03867*.

[21] Morris, C., Peters, J., and Choroba, P. 2013, June. Validation of the time-based separation concept at London Heathrow airport. In *10th USA/Europe ATM R&D Seminar, Chicago* (Vol. 10).

[22] Hasan, S., Shamsuddin, S. M., and Lopes, N., 2014. Machine learning big data framework and analytics for big data problems. *Int. J. Advance Soft Compu. Appl*, *6*(2).

[23] Nasrabadi, N. M., 2007. Pattern recognition and machine learning. *Journal of electronic imaging*, *16*(4), 049901.

[24] Pistelli, M., D'Ariano, A. and Pacciarelli, D., 2011. Optimization models and algorithms for air traffic control in the airspace of busy airports. Technical report. Universita degli Studi di Roma Tre.

[25] Hurter C., Alligier, R., Gianazza, D. and Puechmorel, S., 2014. Wind parameters extraction from aircraft trajectories'. In*: Computers, Environment and Urban Systems 47*, pp. 28–43.

[26] Williams, J. K., 2014. 'Using random forests to diagnose aviation turbulence'. en. In: *Machine Learning 95*(1) pp. 51–70. ISSN: 0885-6125, 1573-0565.

[27] Alligier R., Gianazza D., and Durand, N, 2015. Machine learning and mass estimation methods for ground-based aircraft climb prediction'. In: *IEEE Transactions on Intelligent Transportation Systems*, pp.1–12.

[28] Alligier, R., Gianazza, D., & Durand, N. (2015, June). Machine learning applied to airspeed prediction during climb.

[29] Ahmed, T., Calders, T., and Bach Pedersen, T., 2015. Mining risk factors in RFID baggage tracking data. In: *IEEE 16th International Conference on Mobile Data Management*. *1*, pp. 235–242.

[30] Herrema, F.F. Curran, R., Visser, H.G., Huet, D. and Lacote, R., 2018. Taxi-out time prediction model at Charles de Gaulle airport. *Journal of Aerospace Information Systems, 15*(3), pp.120–130.

[31] Herrema, F.F., Curran, R., Visser, H.G., Treve, V., and Desart, B., 2018. Predicting abnormal runway occupancy times and observing related precursors. *Journal of Aerospace Information Systems, 15*(1), pp. 10–21.

[32] Herrema, F.F., Curran, R., Hartjes, S., Ellejmi, M., Bancroft, S. and Schultz, M., 2019. A machine learning model to predict runway exit at Vienna airport. *Transportation Research Part E: Logistics and Transportation Review*, *131*, pp.329-342.

2.0 Background on Machine Learning and Big Data categories and techniques

2.1 Machine Learning categories

ML algorithms fall into the general categories of *regression*, *classification*, *clustering*, and *anomaly detection*. Each one is designed to address a different type of ML problem. These categories can be divided into two different branches. The first one is called unsupervised learning and the second 'supervised learning'.

- Unsupervised learning: the input data is totally unlabelled. There is no knowledge about the groupings of the data and the data is only based on input. An example is clustering algorithms.
- Supervised learning develops predictive models that predict an outcome or a class of a new data point. In this case there is some knowledge about the input data. For example, you might have some training data, where you have the known labels or the known classes of the data. Here the data is divided into input and output data. In this study there will be a primary focus on supervised learning. Supervised learning takes a known set of input data and known responses to the data, and seeks to build a predictor model that generates reasonable predictions for the response to new data. This is what we want to achieve during the prediction of ASP enablers. Section 2.2 elaborates on the proposed supervised learning strategy. Supervised learning can be divided into two different categories which are classification and regression. The different algorithm techniques per category can be found in Figure 2.1 shows also which techniques are most feasible for each ASP enabler, assessed in Chapter 3-6.

You can have trouble deciding whether you have a classification problem or a regression problem. In that case, create a regression model first, because they are often more computationally efficient. In the supervised learning set, one is given a training set $S = \{(x_i|y_i)\}\begin{cases} n \\ i = 1 \end{cases}$ made of independent and identically distributed random variables from a joint distribution $P(X,Y) = D$. A couple (x,y) is made of an input vector $x \in X$ and a label $y \in Y$. Depending on Y, two sub-settings appear:

- If Y is a discrete ensemble, the setting is called classification. If $card(Y) = 2$ (for example $y = \{-1, +1\}$), one say binary classification, otherwise it is called multi-class classification.
- If Y is continuous, the setting is called regression.

In the sequel, we will focus on the setting $y =$•

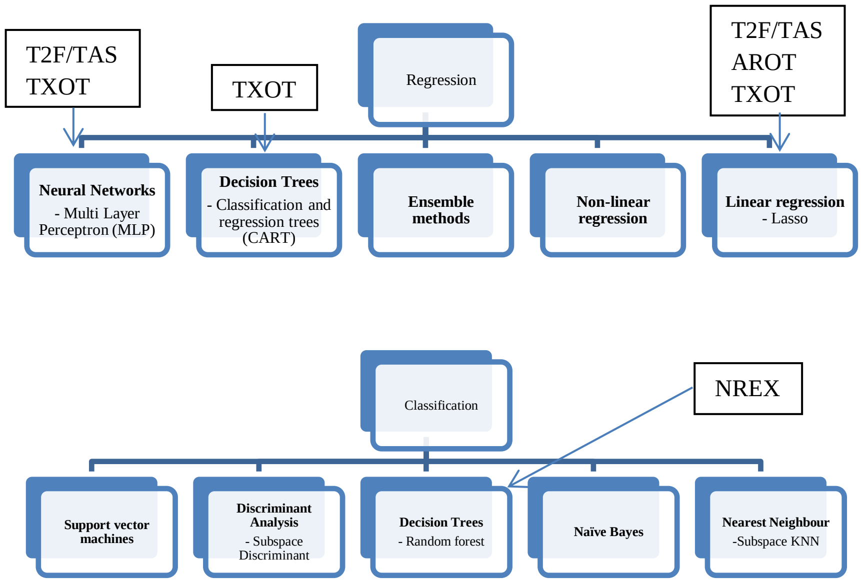


Figure 2.1: The supervised learning categories with their corresponding techniques.

2.2 Strategy supervised learning

The goal of supervised learning is to construct a function $h \in H$ that capture the dependencies between data x and labels y. The quality of the function h is measured using a loss function $l(h(x), y)$:• x •→• + quantifying the deviation between the estimated value and the true label. The best function is the one that minimizes the risk:

$$F_l(\mathrm{h}) = E_{(X,Y)\sim D}[l(h(X), Y)] = \int l(h(X), Y)dP(X, Y)$$

Since the joint probability distribution D is unknown, we need a mechanism to leverage the only available information: $\mathrm{S} = \{(x_i|y_i)\}\left\{_{i\,=\,1}^{\;n}\right.$

That mechanism is referred as empirical risk minimization [1]. We first define the empirical risk as

$$F_{empirical}(\mathrm{h, s}) = \frac{1}{n}\sum_{i=1}^{n} l(h(x_i), y_i)$$

The quantity $F_{emperical}(h, s)$ is an estimator of the true risk $F_l(h)$ based on the sample S such that $F_{empirical}(h, s)$ converges to $F_l(h)$ when the number of examples n in the training set is such that n → ∞ . Hence, minimizing $F_{empirical}(h, s)$ seems to be a good proxy to minimize $F_l(h)$ when $F_{empirical}(h, s)$ is a monotone function. But this is not always true since the quality of the estimator $F_{empirical}(h,s)$ also depends on the chosen family of functions H [2, 3]. Basically, the bigger H is, the more complex a function $h \in$ H is and the more important the probability that $F_{empirical}(h,s)$ diverges from $F_l(h)$ even if $n_l(h)$. A simplified version of the dependence between $F_{empirical}(h,s)$ and $F_l(h)$ can be stated as:

$$F_l(\mathrm{h}) \leq F_{empirical}(\mathrm{h}, \mathrm{s}) + \epsilon_{\lfloor H \rfloor / n}$$

with high probability. This simplified equation shows the so-called *bias-variance tradeoff*. When h is a too simple function, $\epsilon_{\lfloor H \rfloor / n}$ is low but $min_{h\in H}\ F_{empirical}(h,s)$ can be large (large bias, low variance). When h is a complex function, $min_{h\in H}\ F_{empirical}(h,s)$ might be very small but $\epsilon_{\lfloor H \rfloor / n}$ is potentially large (low bias, large variance) [4].

2.3 Machine Learning techniques

<u>Regression</u>; Regression for responses that are a real number, such as True Airspeed (TAS) and Time to Fly (T2F) to threshold. Common regression algorithms include:

- Linear regression; normally with multiple predictor variables. A data *model* explicitly describes a relationship between predictor and response variables. Linear regression fits a data model that is linear in the model coefficients. The most common type of linear regression is a least-squares fit, which can fit both lines and polynomials, among other linear models like the Lasso technique [5]. The most common loss used in regression problems is the *mean squared error* defined as $l_{mse}(h(x),y) = (y - h(x))^2$. Considering the data set $S = \{(x_i, y_i)\}._{i=1}^{n}$, one wishes to find a function h minimizing the empirical risk.

$$F_{empirical}(h) = \frac{1}{n}\sum_{i=1}^{m}(y_i - h(x_i))^2$$

Where:

$$X = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \text{ et } Y = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$$

- Nonlinear regression; Nonlinear regression with multiple predictor variables. Parametric nonlinear regression models the dependent variable (also called the response) as a function of a combination of nonlinear parameters and one or more independent variables (called predictors). The model can be univariate (single response variable) or multivariate (multiple response variables). The parameters can take the form of an exponential, trigonometric, power, or any other nonlinear function. To determine the nonlinear parameter estimates, an iterative algorithm is typically used. A special class of nonlinear models, called *generalized linear models*, uses linear methods.

- Generalized linear models; Regression models for limited responses. Linear regression models describe a linear relationship between a response and one or more predictive terms. Many times, however, a nonlinear relationship exists.
- Decision trees; A decision tree with binary splits for classification. An object of class ClassificationTree can predict responses for new data with the predict method. The object contains the data used for training, so can compute resubstituting predictions.
- Neural networks; The neural networks is a way to model any input to output relations based on some input output data when nothing is known about the model.

Classification; Classification for responses can have just a few known values, such as 'true' or 'false'. Classification algorithms apply to nominal, not ordinal response values, where the data can be separated into specific "classes" like the Runway exit utilised (NREX) (Chapter 6.0). Classification algorithms can also use the algorithms within regression, however in addition the following techniques can be used for classification purposes only.

- Support vector machines (SVM); Support vector machines for binary or multiclass classification. You can use a support vector machine (SVM) when your data has exactly two classes. An SVM classifies data by finding the best hyperplane that separates all data points of one class from those of the other class. The best hyperplane for an SVM means the one with the largest margin between the two classes. Margin means the maximal width of the slab parallel to the hyperplane that has no interior data points. The support vectors are the data points that are closest to the separating hyperplane; these points are on the boundary of the slab.

- Naïve Bayes classifier; Train naive Bayes classifiers. Naive Bayes models assume that observations have some multivariate distribution given class membership, but the predictor or features composing the observation are independent. This framework can accommodate a complete feature set such that an observation is a set of multinomial counts.

- Discriminant analysis; Linear and quadratic discriminant analysis classification. Discriminant analysis is a classification method. It assumes that different classes generate data based on different Gaussian distributions. To train a classifier, the fitting function estimates the parameters of a Gaussian distribution for each class. To predict the classes of new data, the trained classifier finds the class with the smallest misclassification cost.

- Nearest neighbors (*k*NN); Find nearest neighbors for classification. Categorizing query points based on their distance to points in a training dataset can be a simple yet effective way of classifying new points. You can use various metrics to determine the distance, described next.

Clustering; cluster analysis involves applying one or more clustering algorithms with the goal of finding hidden patterns or groupings in a dataset. Clustering algorithms form groupings or clusters in such a way that data within a cluster have a higher measure of similarity than data in any other cluster. The measure of similarity on which the clusters are modelled can be defined by Euclidean distance, probabilistic distance, or another metric. The distinguishing feature of each of these algorithms is the metric to measure similarity. Popular clustering algorithms (Figure 2.2) include:

- Hierarchical clustering: builds a multilevel hierarchy of clusters by creating a cluster tree.
- k-Means clustering: partitions data into k distinct clusters based on distance to the centroid of a cluster.
- Gaussian mixture models: models clusters as a mixture of multivariate normal density components.
- Self-organizing maps: uses neural networks that learn the topology and distribution of the data.

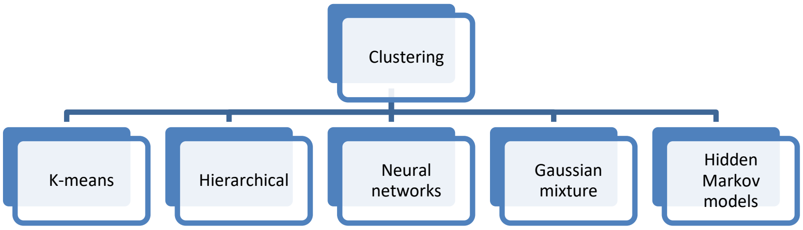


Figure 2.2: Popular clustering algorithms.

For a more detailed introduction to Machine Learning, please refer to [6] or [7].

2.4 Criteria for choosing feasible Machine Learning techniques

For choosing a feasible supervised learning technique for assessing *ASP enablers*, first it should be identified to which category it has to comply. The different categories and techniques are elaborated in Section 2.1.

As a next step, the designed classifier per prediction parameter will be tested on the following criteria; (1) *Accuracy* of the classifier, or the percentage of records that are classified correctly, should be as high as possible. This is very important, because the results of the Rule Discovery phase (which are influenced very much by this classification) should be very accurate. The objective is to train the classifier in such a way that it makes (as much as possible) the same decision as what the ATCO would do. (2) Amount of data needed; for different ML techniques and problems different amount of data is needed for having an *accurate*, *intuitive* and *interpretable* ML model. (3) Performance when combining ML techniques; The best three machine learning techniques are combined with each other and tested on their *accuracy*. (4) Clear decision process; For a good understanding of *intuitive* and *interpretable* predictions, the decision process should be very clear. As shown in the “Introduction to Machine Learning and Pattern Recognition” [6], neural networks and support vector machines both have an unclear decision process, because of their “ black box” nature. A Bayes classification is not easy to interpret, because it is based on a probability distribution. By contrast, decision tree classifiers and rule-based classifiers are very clear, because decision trees and rules are easily interpretable. Nearest neighbour classification is easy to interpret, because it is based on the largest vote among the nearest neighbours of the input vector. (5) Relatively easy to implement; Because preferably, the classifier has to be implemented in Visual Basic, it should be relatively easy to encode it. Neural networks are certainly the most difficult to implement, requiring specialized software or toolboxes. Support Vector machines and Bayes Classifiers are moderately easy to implement, using a model built with specialized software. Decision tree classifiers can be converted into a rule set and are therefore well suited for a software implementation. Implementing the output rule set of rule-based classifiers is the easiest way to implement a classifier. In the case of nearest neighbour classification, no model has to be built. The implementation therefore requires some work, but is straightforward. (6) Performance when applying feature analysis; Before the model will be trained, first the most important (group) features will be selected using Principal Component Analysis and RreliefF modelling (feature selection). The objective of feature selection is three-fold: improving the ASP prediction *accuracy*, providing faster and more effective predictors, and providing a better understanding of the underlying process that generated the data. (7) Relatively *fast*; Because it is the intention to process a large amount of data, classification speed should not be to slow. Nearest neighbour classification performs the worst: every time a record has to be classified, the whole set of data has to be scanned to find the nearest neighbours. For all the other techniques, the model is built only once, during the training phase. Classification of a new object is therefore much faster than for a nearest neighbour classifier. The speed of classification is for all these methods comparable.

The above criteria will be assessed on all ASP enablers in Chapter 3-6. Based on the outcome feasible techniques will be extracted and used for the RU validation use case in Chapter 7 for finding ASP patterns and abnormal behaviours.

2.5 Big data techniques

We will focus on storage and processing capabilities separately and then move on to data streams and how they solve issues related to the various distributed processing models we will describe. Within this study Amazon Elastic Map Reduce (EMR) is used which is described in detail in the next section.

Distributed Processing

EMR is the industry leading cloud-native big data platform, allowing to process vast amounts of data quickly, and cost-effectively at scale. Using open source tools like Apache Spark which is a unified analytics engine for Big Data (BD) processing, with built-in models for ML [8]. Within this study spark is coupled with the dynamic scalability of Amazon EC2 and scalable data storage of Amazon S3. EC2 forms a central part of Amazon's cloud computing platform, by allowing users to rent virtual computers on which to run their own computer applications. Data is scaled and stored in the S3 bucket.

<u>MapReduce:</u> A data processing job is described as two parallel operations: Map and Reduce. These can then be chained for more complex workloads. The MapReduce paradigm introduces a simple way of declaring processes while maintaining enough flexibility to describe more complex operations. It also allows for parallel execution. In particular, the map tasks are distributed across the data nodes and are performed locally on the data blocks. The result of such a map task is then partitioned across various reducers spread out across the cluster. The map task is the transformation of an input key value pair into a key value pair suitable for the reducer. Often times, the input key value pair is just a line number and the corresponding

content. The reducer on the other hand is an operation on the group of values that belongs to a certain key. Important to note is the arbitrary shuffle and sort step in between map and reduce tasks. A key limitation of the model is also the necessary synchronization after each map and each reduce step. Hadoop provides an API for declaring these MapReduce jobs and a framework that knows how to execute them in a distributed way. Furthermore, through its ecosystem, Hadoop also provides various abstractions on top of the MapReduce paradigm. So while MapReduce is a strong framework for doing parallel computations on top of large data sources, it is not suited for low-latency processing on data fractions.

Extended MapReduce

Because of its limitations, some frameworks like Apache Spark and Apache Flink, have adopted a more advanced version of the MapReduce paradigm. They provide a slightly richer API than traditional MapReduce and leverage memory to speed up the process. Since they are still batch processing frameworks, they remain unsuitable for near real-time processing on fractions of the data, but they will get the batch jobs done more quickly. The key difference is that these frameworks do not require data to be loaded from disk all the time. By keeping them in memory between jobs, they are able to outperform the traditional MapReduce jobs. In particular for iterative algorithms or multiple ad-hoc queries on the same data, they can achieve great performance gains. Furthermore, they are generally built on top of a DAG engine, an engine able to execute a job defined as a directed acyclic graph where the vertices represent certain operators and the edges define the data flow across those operators. This DAG engine allows the framework to translate the process into one single job rather than having to define multiple MapReduce jobs and chain them. This allows them to better understand the full work flow and thus make smarter choices in terms of which data sets should be cached. Because of their ability to handle iterations more efficiently, these frameworks are often the first choice when it comes to implementing distributed ML algorithms and often have a library in place with pre- built algorithms.

Data Streams mini batch processing In some ways derived from the extended MapReduce paradigm, mini batch processing or discretized stream processing arbitrarily splits a stream into an infinite set of finite mini-batches. Spark Streaming, part of the Apache Spark project, is an implementation of such a model. Through various optimizations, operations such as map and reduce can now be applied to a much smaller set of records, drastically decreasing the latency. While a valid option for some use cases, the extended MapReduce API is not flexible enough for all streaming use cases. It does provide some benefits over the traditional stream processing model discussed below though. As data comes in, mini batches are dynamically cached and processed on the worker nodes. This allows the framework to perform dynamic load balancing, limiting the impact of skew in workloads. The model also allows for faster recovery by executing failed jobs in parallel on multiple other nodes. This can be important of that failed job is delaying the entire processing job. In general, the similar development approach between extended MapReduce and discretized stream processing is also considered an added advantage when attempting to combine the two in what is called a Lambda architecture delivery [9].

References

[1] Vladimir Vapnik. "Principles of risk minimization for learning theory". In: Advances in neural information processing systems. 1992, pages 831–838.

[2] Vladimir N. Vapnik. "An Overview of Statistical Learning Theory." In: IEEE Transactions on Neural Networks 10.5 (1999), pages 988–999.

[3] Stéphane Boucheron, Olivier Bousquet, and Gábor Lugosi. "Theory of Classification: a Survey of Some Recent Advances". In: ESAIM: Probability and Statistics 9 (Nov. 2005), pages 323–375. ISSN: 1292-8100, 1262-3318.

[4] Avrim Blum, Adam Kalai, and John Langford. "Beating the hold-out: Bounds for k-fold and progressive cross-validation". In: Proceedings of the twelfth annual conference on Computational learning theory. ACM, 1999, pages 203–208.

[5] Robert Tibshirani. "Regression shrinkage and selection via the lasso". In: Journal of the Royal Statistical Society 58.1 (1996), pages 267–288.

[6] Bishop, C. M. (2006). Pattern recognition and machine learning. Springer.

[7] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. The Elements of Statistical Learning. Second Edition. Volume 1. Springer series in statistics Springer, Berlin, 2008.

[8] Zaharia, M., Chowdhury, M., Franklin, M. J., Shenker, S., & Stoica, I. (2010). Spark: Cluster computing with working sets. *HotCloud*, *10*(10-10), 95.

[9] SafeClouds, WP2.2 – Identifying Runway Utilisation hazards with Machine Learning. (2018). Available at: www.safeclouds.eu (Accessed: 17 January 2019).

3.0 Evaluation of feasible machine learning techniques for predicting the time to fly and aircraft speed profile on final approach

Over-conservative separation values expressed in distance or time are used by ATCOs at different hub airports. Currently they do not take into account ASP enablers as defined by ATCOs. In this Chapter the most feasible ML techniques are reviewed to model the ASP enablers Time to Fly (T2F) and True Airspeed (TAS) profile on final approach. Accurate predictions are used to refine the separation values. Chapter 3 consists of three parts. First, different ML techniques are assessed on their forecasting performance, computational time and the amount of data needed for delivering a reliable prediction. Second, the techniques are applied to the traffic of two major European airports. Finally, they are benchmarked against the ORD tool using a statistical approach for predicting the T2F and TAS. The most efficient ML techniques are applied on two case studies.

This chapter was previously published as:

Herrema, F.F., Curran, R., Treve, V. and Visser, H.G., 2016, June. Evaluation of feasible machine learning techniques for predicting the time to fly and aircraft speed profile on final approach. In International Conference on Research in Air Transportation (Vol. 8, No. 4, pp. 4-8).

Abstract
Currently, at many airports, the runway throughput is the limiting factor for the overall capacity. Among the most important constraining parameters is the separation minima expressed in distance. On the top of these minima, the difference of the leader and follower aircraft speed profiles imposes to consider buffer to cope with compression effect. Currently, Air Traffic Control Officers (ATCOs) take these buffers on the basis of their training and experience. However, this experience will not be sufficient to safety deploy advanced concepts, like pair-wise separations, that increase variability in the separations to be delivered and therefore in the compression buffer to be considered. Systematic analysis of years of radar tracks has allowed to better predict the buffers to apply by characterising the time to fly (T2F) given a separation distance and True Airspeed (TAS) profile as a function of meteorological parameters.

This chapter presents how Machine Learning (ML) techniques may be used for predicting the T2F and TAS profile on final approach. Different ML techniques will be assessed on their forecast performance, computational time and amount of data needed for delivering a reliable prediction. The techniques will be applied on two different major European airports traffic and will be benchmarked against Optimized Runway Delivery (ORD) study using a Model Based Approach (MBA) for deriving the T2F and TAS. As a result, the most efficient ML techniques will be applied on two case studies for predicting the T2F and TAS.

3.1 Introduction

ML can be used to identify patterns[4] and to observe 'what-if' scenarios in past data. These patterns can be transferred into 'what-if' statements by analysing relations between the response variables (T2F and TAS) and the prediction variables highlighted in Table 3.1. This analysis is needed to predict forthcoming operational risks during real time operations like loss of separation [1]. Such a prediction would feed 'what-if' tools at the airport to alert ATCO about impending aircraft behaviours.

Presently Distance Based Separation (DBS) or Time-Based Separation (TBS) rules are applied during final approach. As a next step the 'Dynamic pair-wise separation' concept is proposed to allow controllers to sequence arriving and/or departing aircraft using Time-Based, Weather Dependent and Pair-wise wake turbulence separations. The efficient deployment of such concept needs a reliable prediction of the T2F and TAS, which is mainly influenced by the aircraft type and wind profile. With this respect, it is envisaged to progressively move from a MBA to a ML approach for coping with the variability of aircraft speed behaviours. In this chapter, ML techniques will be assessed on their capabilities to produce fast and accurate predictions and their capabilities to test a large number of 'what-if' statements.

[4] Patterns in this context are defined as the normal distributions (from -2 to +2 sigma) of T2F and TAS for a given flight and for a range covering the last 10NM of the final approach

This chapter presents 10 feasible ML techniques; the amount of data needed, Principal Component Analysis (PCA) and feature analysis for predicting the T2F and TAS profile per wind-band on final approach based on 15 prediction variables. All relevant scenarios between combined data, ML techniques and problems will be assessed. As a result, the most efficient ML techniques will be applied on two case studies for predicting the T2F and TAS.

The structure of this chapter is as follows; Firstly, the methodology and data sources such as; aircraft data, wind and speed profiles are described. Secondly, the ML context is outlined. Thirdly, the pre-processing steps are elaborated. Fourthly the results of the two case studies and the respective prediction error are outlined in subchapter 3.7. Finally conclusions and recommendations are drawn.

3.2 Data, Prediction Variables & Modelling

In order to predict T2F and TAS profiles on final approach, two complementary sources of information are used; aircraft performance data and weather data.

3.2.1 Aircraft Performance & Weather data

Aircraft Performance is extracted from Radar data and has been provided by the Air Navigation Service Providers (ANSPs) for four airports. For one other airport, ADS-B data has been used. For each airport, the radar data cover 2 months of operations in 2012 or 2013. The ADS-B data covers two weeks of data in 2013. In total, the data comprises about 130,000 flights. For each radar point, the flight ID, aircraft type, actual time, lateral and longitudinal position coordinates, altitude and ground speed is recorded with a 4s refresh rate. The focus of the analysis is on approaching aircraft to single runways over the last 10NM. All variables can be found in Table 3.1.

The headwind (HW) profile and visibility measurements were gathered from Airport 1 and Airport 2 for respectively a period of 2 months and 3 years. The HW profile is defined by four measurements at heights of 10m, 500m, 1000m and 3000m. The HW profile is analysed per 10 minutes. Wind measurements are grouped into six different wind bands; 0-5kt TW (tailwind) and 0-5kt, 5-10kt, 10-15kt, 15-20kt and 20-25kt (headwind). The HW profile and visibility date and time have been included as input variables into Table 3.1.

3.2.2 Prediction variables

Table 3.1 gives all the 15 input prediction variables per 0.5NM segment. However, for some prediction variables the number is constant such as; recatEU, rwy and FAF or some are not considered such as gspass from 10NM till 0.5NM and AC from 19.5 till 10NM. The predictive response variables in our model are the T2F, y, and TAS, y’ and are outlined in Table 3.1 as number 16 and 17.

Table 3.1: Prediction and response variables.

Variable	Description	Variable	Description
1.Flightnr	Flight number	10.METARcwnd	Crosswind (kts)
2.Apt	Airport	11.METARvsby	Visibility (m)
3.actype	Aircraft type	12.ICAOcombi	ICAO combination
4.dpass	Distance from threshold (NM)	13.actypecombi	Aircraft type combination
5.hpass	Height from threshold (m)	14.ACin10NM	Number of aircraft between 0 and 10NM
6.gspass	Ground Speed (kts) from 10 till 19.5NM	15.FAF	Final approach fix
7.rwy	Runway (degrees)	16.T2F (response)	Time to fly (s)
8.recatEU	RECAT EU category	17.TAS (response)	True Airspeed (kts)
9.METAR hwind	Headwind (kts)		

3.3 Modelling of ML techniques

The ML techniques will be addressed by modelling the predicted T2F and TAS values under different weather conditions. The T2F is a continuous variable that is computed depending on variables such as aircraft type, airport and parameters related to weather conditions. The approach is based on learning a model per airport and aircraft type - A320 at Airport 1 and the B738 at Airport 2. The results for these two case studies will only be shown for the best 3 feasible ML techniques. Going one step further, we propose to approach the prediction as a multi-task learning problem. This approach can lead to a better model for the main task by exploiting the commonality among the tasks. In this research, this leads to the following consideration: instead of predicting T2F and TAS for each segment of 0.5NM from 0.5NM to 10NM individually, we propose to exploit multi-task learning by predicting the segments altogether. By solving the regression (subchapter 3.7) problem jointly for all these segments, we expect to improve the performance of the regression compared to the case where the segments are considered independently. The rationale behind this is that although the distribution of the T2F values depend on the segment, the behaviour of the aircraft on all the segments is subjected to the same conditions.

3.4 Context-Machine Learning

This section describes the feasible ML category and techniques for predicting the T2F and TAS profile on final approach.

3.4.1 ML techniques classification

ML techniques can be classified into different categories following three main strands; unsupervised learning, supervised learning and reinforcement learning. Supervised learning can be divided into two different subcategories which are classification and regression. In this study, there will be a primary focus on supervised regression learning since these are often computationally efficient for predicting the T2F and TAS (real numbers) whereas classification is often used for binary predictions such as go-arounds. For supervised regression learning, we propose two approaches that can be considered as baselines for this study [2] [13]. The first method is based on linear regression techniques and the second method on neural networks. These two methods can be divided into 10 sub techniques based on multi-task learning. Multi-task techniques are selected since we try to jointly fit the T2F and TAS for all segments from 0 till 19.5NM (0.5NM step). By definition, a multi-task learning approach learns a problem together with other related problems, all at the same time. Learning multiple related tasks simultaneously has been empirically [3, 4, 5, 6, 7, 8, 9, 10] as well as theoretically [3, 11, 12] shown to often significantly improve performance relative to learning each segment independently. The 10 feasible ML techniques are outlined in subsection 3.4.2 and 3.4.3.

3.4.2 Regression techniques to be tested

The regression techniques **fitglm**, **stepwiseglm**, **ridge regression** and **Lasso** are proposed for this problem. A variant of these techniques, called **Elastic net**, which combines the penalties of both methods and which is also a good candidate to tackle this particular prediction problem [14]. The Mean Square Error (MSE) serves as cost function for these algorithms. These techniques are applicable in the case of multi-task regression and are referred in the scientific literature as **multitask regularized regression**.

3.4.3 Neural networks techniques to be tested

Approaches based on neural networks are also proposed for the problem. We recommend the **Multi-Layer Perceptron** (MLP) as a baseline to tackle our case. The loss function used to train the network will be the MSE.

Neural networks are widely used in scenarios in multi-task learning, by making use of the fact that the underlying representation of the problem is inherently learnt during the training process. Depending on the amount of data at disposal, deeply connected neural network architectures will also be considered such as; **Auto Encoder, Boltzmann** and **Recurrent Neural Networks.**

3.5 Pre-Processing

This section describes the pre-processing steps to come up with a usable aircraft performance data set. This data set is needed to train a T2F and TAS prediction model. Each pre-processing step is detailed below:

A. **Compute T2F and TAS for each sample**; Before feasible ML techniques can be applied first the T2F and TAS profile are extracted for each segment of 0.5NM, 5kts wind band and aircraft type. Remove the samples where the T2F and TAS of one segment is more than 2 standard deviation away from the segment mean. This forms a matrix ***Y*** where each row represents a flight and each column a segment.
B. **Feature selection;** The RreliefF technique is applied before a model is learned.
C. **PCA:** finding out which features are important for best describing the variance in a data set.
D. **Construct the datasets**: based on different data sources and the Table 1 mentioned variables. Furthermore standardize feature matrix ***X***.
E. **Stability of three different data parts:** split the matrices *X and Y* in two subsets *Xtrain; Ytrain;* used to train the model and *Xtest;Ytest* used to evaluate the model accuracy. For those experiments the data is split into 70% of training data, 15% of test data and 15% of validation data (standard hold-out).
F. **Accuracy of data and outliers:** in the last pre-processing step the accuracy is measured and the outliers are shown.

3.5.1 Compute T2F and TAS profiles

The T2F is computed by the difference in time from a certain distance till threshold. The TAS is calculated by subtracting HW or adding TW of the wind profile from/to the GS profile. This study works with TAS since this gives a better indication of the speed compensations applied per aircraft type. 20.000 flights where extracted from Airport 1 to cover seasonal variations and to have a minimum of 50 measurements per aircraft type, wind-band and 0.5NM segment. Figure 3.1 shows an example of the TAS, GS and HW profile of an A318 in 10-15kts headwind as a function of distance from the threshold. The T2F and TAS results for 50 different aircraft types can be found in the report [15].

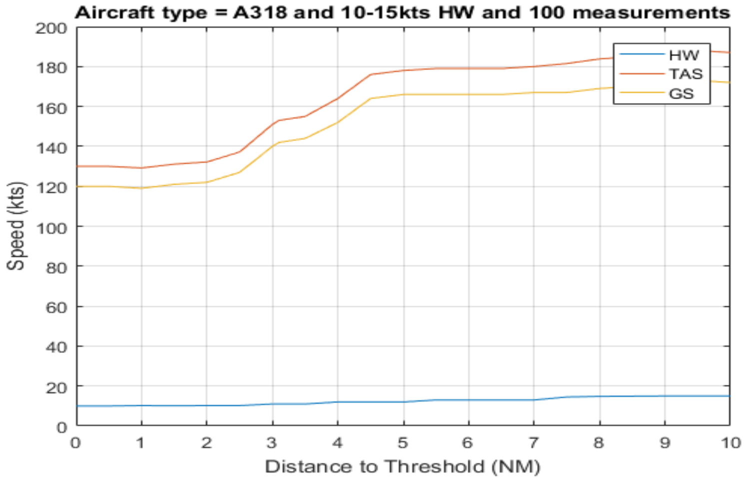


Figure 3.1: Example TAS, GS and HW profile versus distance to threshold.

3.5.2 Feature selection

Before the model will be trained, first the most important (group) features will be selected using PCA and RreliefF modelling (feature selection). The objective of feature selection is three-fold: improving the prediction performance of the predictors, providing faster and more effective predictors, and providing a better understanding of the underlying process that generated the data [16]. RreliefF has commonly been viewed as a feature selection method that is applied in a prepossessing step before the model is learned [17]. The standard RreliefF regression modelling technique has been extensively discussed in many papers [18]. The technique has been applied on 500 low wind (0-5kt) A320 flights for Airport 1 as showed in Figure 3.2.

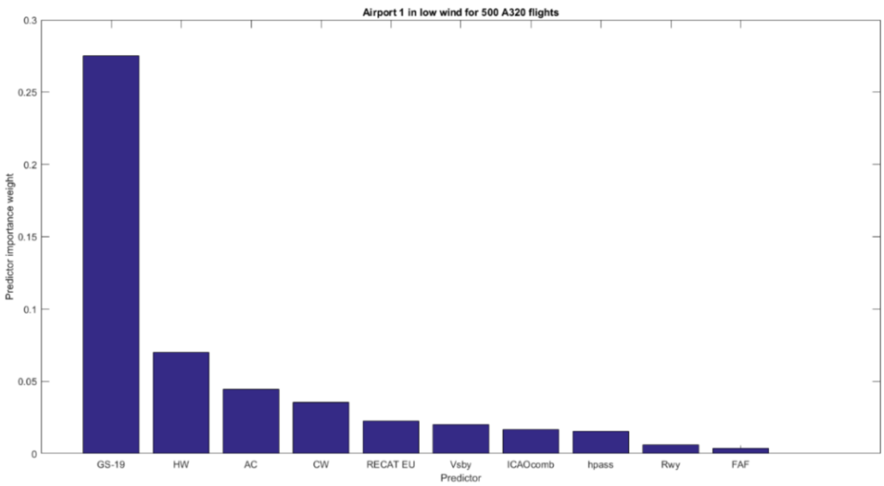


Figure 3.2: Normalized feature selection using RreliefF algorithm.

Figure 3.2 shows from left to right the most important succeeding features for Airport 1. The ground speed at 10NM (GS-19) seems to have the most impact on the T2F, followed by the headwind (HW) and Aircraft type (AC). Similar feature relationships are obtained for Airport 2 and different aircraft in low wind. According to the ORD study [19], the top 3 most important theoretical features match with the predicted RreliefF features. Table 3.2 in Section 3.6 compares the best (group) prediction features for different amount of flights for both PCA and RreliefF.

3.5.3 Assessibility of PCA

After applying RreliefF (feature selection), PCA will be applied. PCA is a procedure for identifying a smaller number of linearly uncorrelated variables called principal components. The goal of PCA is to show as much of the variability in the data as possible with the fewest number of principal components. The data have been divided into 15 different indicators of aircraft and weather behaviour at 2 different airports, which are showed in Table 3.1.

Figure 3.3 shows the top 10 feature selected variables, which are represented in a bi-plot by a vector, and the direction and length of the vector indicate how each variable contributes to the two principal components in the plot. In the new coordinate system, the first axis corresponds to the first principal component, which is the component that explains the greatest amount of the variance in the data, whereby it is obvious that component 2 explains the 2nd greatest amount of variance in the data, etc. In this example, the first principal component, on the horizontal axis, has positive coefficients for GS, Visibility, HW, AC, RECAT CAT, ICAO comb and FAF variables. That is why the seven vectors are directed into the right half of the plot. The largest coefficients in the first principal component are the second, third, fourth and seventh elements, corresponding to the variables HW, GS, RECAT CAT and FAF. The second principal component, on the vertical axis, has positive coefficients for the variables Runway, Height, Cwnd, AC, RECAT CAT, ICAO comb, Visibility, HW and negative coefficients for the GS and FAF variable.

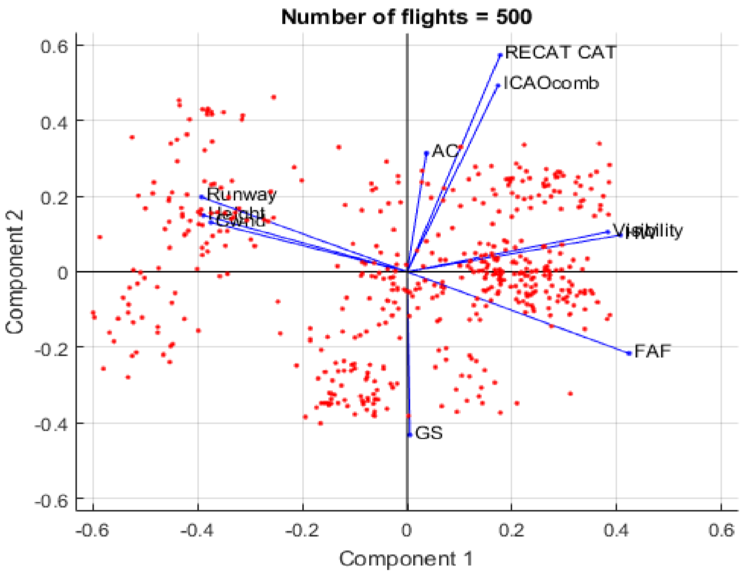


Figure 3.3: A bi-plot in two dimensions, to find the relation among different variables.

Since Figure 3.3 doesn't explain enough of the variance in the data of the first two principal components, Table 3.2 and Table 3.3 takes also into account component 3 and 4.

3.5.4 Construct the datasets

The first dataset includes the features flightnr, dpass, hpass, gspass, rwy, RECATEU, METARvsby, ICAOcombi, actypecombi, ACin10NM and FAF. Please note that we only consider the measurements from 19.5NM to 10NM. This forms the feature matrix X where each row is a flight and each column a feature. Another dataset is built from 10NM to 0.5NM with the same features plus the headwind at each segment determined as the difference between GS and TAS. When this is done the historical data will be divided into predictor variables and response variables. Finally, for each column X subtract the columns mean and divide by their standard deviation.

3.5.5 Stability of three different data parts-cross validation

To check the stability of different data parts, the data will be randomly divided into training, validation and testing subsets. Stability is defined as how the ML algorithm is perturbed by small changes to its input. A stable algorithm is one for which the prediction does not change much when the training data is modified slightly. It has been assumed that the default fractions in this study for training, testing and validation are 0.7, 0.15 and 0.15, respectively. The model is adjusted accordingly when training it. The validation is used to measure network generalization, and to halt training when generalization stops improving. To prove that a randomly selected data set is stable, epoch and validation checks are performed. Epoch indicates the amount of a single pass through the entire training set, followed by testing of the verification set. Thereafter we check convergence on the validation and at the end of the learning process the model is evaluated on the test set. The test has no effect on the training and therefore provides an independent measure of network performance during and after training. Figure 3.4 shows a final approach trained speed model by selecting 5000 A320 flights. By training the model according to the above described method, a good representation of real life flights will be given and unstable data parts are neglected.

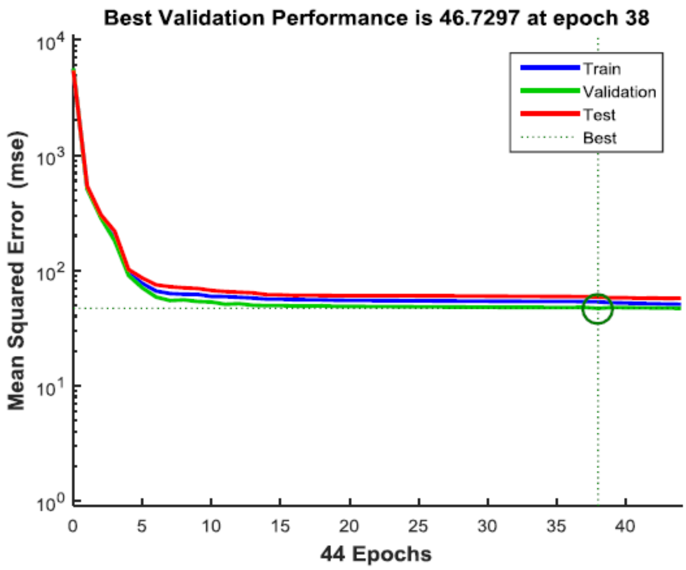


Figure 3.4: MSE versus amount of epochs for 5000 A320 flights.

3.5.6 Accuracy of data and outliers

Outliers exist when building predictive models. With outliers we mean when a data point or flight is not consistent with the other data points. One way to show this inconsistency is by plotting the regression for training, validation, test and all. Figure 3.5 shows such an example where the regression R values measure the correlation between outputs and targets. An R value of 1 means a close relationship, 0 a random relationship.

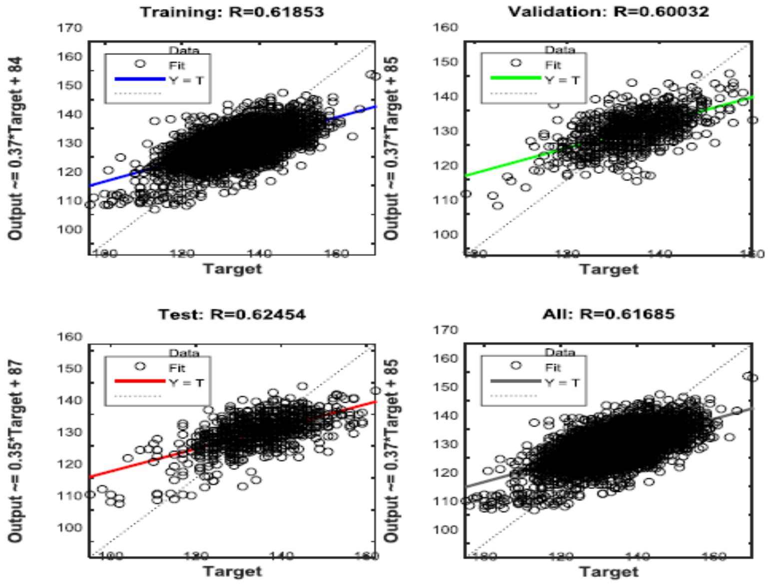


Figure 3.5: R values for test, training, validation and all for an A320 predictive TAS model.

Analysing these graphs shows that there are indeed many outliers. It will be obvious that by neglecting them in the target set, a better R value will be obtained for the predicted model. Doing this for the above example results in an overall R value of 0.69 instead of 0.61 presented in Figure 3.5.

3.6 Results

This chapter shows the results of the feasible ML techniques PCA and RreliefF for prediction of the T2F and TAS profile on final approach. The best technique will be assessed on the number of neurons and minimum amount of flights needed to come up with an accurate prediction model.

3.6.1 PCA groupings and RreliefF for A320 at Airport 1

Using PCA dimension reduction and/or feature selection will automatically not result in a better prediction model. It could happen that by excluding variables, you exclude automatically variables that are correlated with each other. It has been tested if by applying PCA and feature selection before training a ML model result in; (1) less time to compute, (2) a lower Mean Squared Error (MSE) and (3) an increased accuracy (lower sigma). Based on Table 3.3 and MBA experiences for different types and wind conditions, analysis are executed on the MSE by excluding expected correlations compared to including them. First Table 3.2 compares the important (group) prediction features for different amount of flights for both PCA and RreliefF.

Table 3.2: PCA and RreliefF outcome for different amount of flights.

Number of flights	**PCA**	**RreliefF**
50	9,6 (group 1) 3,10,11,5,8,12 (group 2) 7,15 (group 3)	6,3,9,10,8,12,11,15,7,5
100	3,9,6 (group 1) 10,12,8,11,5 (group 2) 15,7 (group 3)	9,6,3,12,10,8,11,15,5,7
300	6 (group 1) 9,3,8,11 (group2) 12,10,5,7 (group 3) 15 (group 4)	6,9,3,10,8,11,12,5,7,15
500	6 (group 1) 9,3,8,11 (group2) 12,10,5,7 (group 3) 15 (group 4)	6,9,3,10,8,11,12,5,7,15

First from this table it can be concluded that, based on PCA, 4 main groupings are correlated with each other. The numbers correspond to a certain prediction parameter and can be found in Table 3.1. After applying RreliefF we verified with PCA that above 400 flights, the prediction parameters influencing the response for the A320 flights at Airport 1 in low wind remain stable. At this stage the minimum amount of 400 should be inserted for designing an accurate prediction model. Table 3.3 shows for the MLP the TAS MSE and sigma results of the Table 3.2 mentioned groupings. These results are obtained by building a predictive model for 500 and 250 A320 flights in low wind at Airport 1 and by in-and excluding group correlations from Table 3.2. The MSE and sigma results are averaged per 0.5NM segment.

Table 3.3: MSE and sigma results for 4 different groups.

Group	MSE-500	Sigma-500	MSE-250	Sigma-250
Group 1	49.1	6.5	57.7	7.7
Group 1, 2	46.0	6.1	55.1	6.8
Group 1, 2, 3	48.0	6.3	57.2	7.5
Group 1, 2, 3, 4	48.3	6.3	59.5	7.4

Analysing the MSE using all decision parameters compared to the first three groups, first two groups and first group result in respectively; a 1% and 5% improvement and 2% reduction for the TAS MSE (Table 3.3). The same results are obtained for the T2F.

For this A320 flight case we conclude that after PCA we only apply the variables that are correlated with component 1 and 2. This can be explained by the fact that the first two principal components for PCA cover around 90% of the correlation for the response variables. Furthermore the time for learning the model stays the same and the sigma values are lower (increased accuracy) for 500 flights compared to 250.

3.6.2 Assess feasible ML techniques

In this subsection the 10 feasible ML techniques will be assessed on their training time (speed), number of parameters and performance indicators. The most important performance parameter to be minimized by the predictions of the models is the Root MSE (RMSE). The RMSE will be calculated from 8.5NM to 0.5NM according to Equation 1;

$$RMSE_{8.5-0.5NM} = \sqrt{\frac{1}{N}\sum_{i=0}^{N}(T2F_{8.5-0.5NM} - T2'F_{8.5-0.5NM})^2} \qquad \text{Equation 1}$$

Where $T2'F_{8.5-0.5NM}$ is the sum of the predicted T2F for each segment between 8.5NM and 0.5NM before runway. Table 3.4 shows the outcome for the 10 best feasible regression and neural network techniques. The technique with the highest grade receives 10 points whereby the lowest receives 1 point.

Table 3.4: Assess feasible ML techniques on different performance indicators.

Technique	**Performance indicators**				
	MSE Performance 3p	***Computational time 2p***	***Apply RreliefF and PCA 2p***	***Implementation clear decision process 2p [11]***	***Outcome***
Lasso	10	9	6	9	78
MLP	5	7	10	8	65
Elastic net	8	5	9	5	62
Ridge	7	8	7	4	59
Auto Enconder	9	4	8	4	59
Recurrent Neural Networks	4	10 (5 seconds)	2	8	52
Boltzmann	5	3	3	4	35
Regularized regression	2	2	2	2	18
Stepwiseglm	2	2	2	2	18
Fitglm	3	1	1	1	12

From Table 3.4 we conclude that MLP and Lasso performs best. Both techniques will be combined to design a third feasible technique – ensemble. The third model refers as ensemble which is simply the average of the predictions of the Lasso and MLP. Combining these techniques result in a more robust and accurate ML model [21].

3.6.3 Relation number of hidden neurons versus MSE

Figure 3.6 shows the MSE outcome versus the number of hidden Neurons for the best performed neural network technique. The MLP outcome has been analysed for 5000, 10000 and 30000 flights in low wind [20]

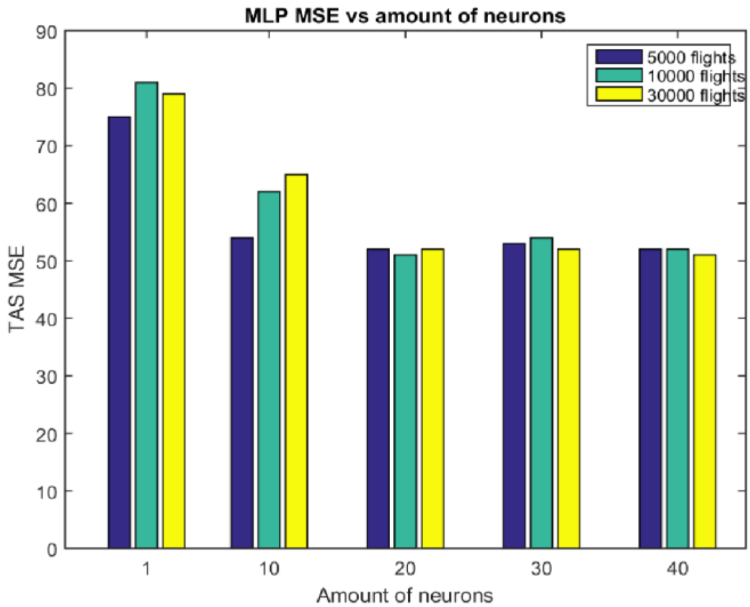


Figure 3.6: TAS MLP MSE vs number of Neurons.

It can be concluded by minimizing the validation MSE, the optimal amount of neurons lies between 20 and 23 for respectively 5000, 10000 and 30000 flights.

<u>Ensemble performance for different number of flights</u>

The ML model is programmed in such a way that it is able to calculate the MSE for different types, wind conditions and for 2 different airports. Figure 3.7 shows for the ensemble ML technique and MBA the MSE and sigma performances as function of the total number of flights for low (0-5kts) and strong wind (20-25kts) conditions at Airport 1.

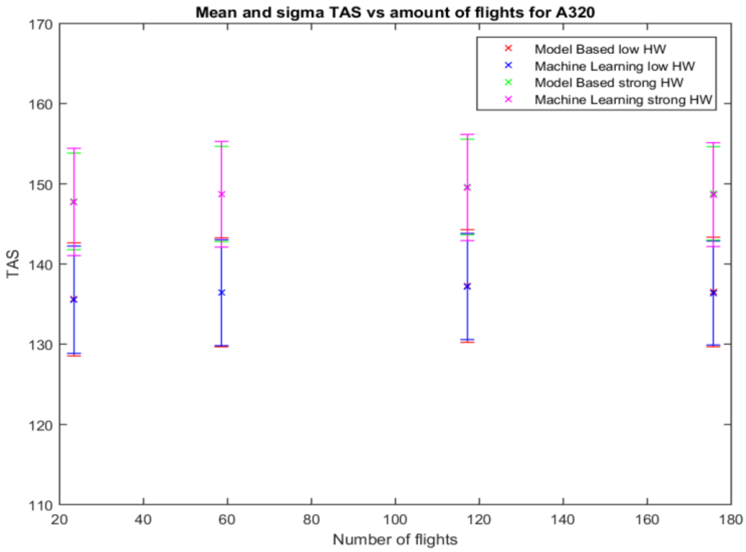


Figure 3.7: Mean and sigma TAS vs amount of flights for low and strong wind at 1 NM from threshold.

We conclude from Figure 3.7 that ensemble produces results comparable to MBA (differs between 1 and 2%) and that the standard deviation values are unaffected by sample size. Furthermore by analysing the MSE, we need 60 flights to build a ML model with accurate results - for the other aircraft types stable MSE values are obtained after learning the model with a minimum of 70 flights. The ensemble model is also validated with an additional data set from Airport 2 and shows comparable results.

Based on the results showed above, it can be concluded that no prediction should be made based on fewer than 60 flights per aircraft type and wind-band. Furthermore outliers like NaN and 0 values should be excluded from the sample data set for valid predictions. The dataset need to be carefully constructed and measured by analysing the R value for a correct output of the model.

3.7 Case Study Results

In this section we analyse two T2F case studies using the Lasso, MLP and Ensemble techniques. During the first case study Airport 1 and aircraft type B738 are analysed. Thereafter we analyse Airport 2 and aircraft type A320. The RMSE of the MBA is estimated using the mean of dataset 1 and dataset 2 (Section 3.5.4). We compute the RMSE from 8.5NM to 0.5NM according to Equation 1 and accordingly, the RMSE from 4.5NM to 0.5NM which is given by Equation 2:

$$RMSE_{4.5-0.5NM} = \sqrt{\frac{1}{N}\sum_{i=0}^{N}(T2F_{4.5-0.5NM} - T2'F_{4.5-0.5NM})^2} \qquad \text{Equation 2}$$

Where $T2'F_{4.5-0.5NM}$ is the sum of the predicted T2F for each segment between 4.5NM and 0.5NM before the runway.

3.7.1 Airport 1 and B738

For this airport and aircraft, the estimated errors of the MBA are 6.35 (RMSE 8.5-0.5NM) and 3.65 (RMSE 4.5-0.5NM). Table 3.5 reports the errors of the different models. Clearly the headwind at each segment helps a lot diminishing the RMSE. Compared to the RMSE of the MBA, none of the model trained on dataset without headwind at each segment achieves the same performance. However, when we also take into account this feature our best model (Lasso) improves by 19.7% over the MBA for the 8.5 to 0.5 NM T2F task and by 19.5% for the 4.5 to 0.5NM task. Averaging the two models lead to a better performance without the headwind but with the headwind, if suffers the bad accuracy of the MLP.

Table 3.5: Applying top 3 feasible techniques on first case study.

Model	Headwind	Size	RMSE 8.5-0.5NM	RMSE 4.5-0.5NM
Lasso	Yes	(1321,331)	**5.0**	**2.9**
Lasso	No	(1388,347)	8.0	**4.6**
MLP	Yes	(1321,331)	5.3	3.1
MLP	No	(1388, 347)	8.0	4.7
Ensemble	Yes	(1321,331)	5.1	3.0
Ensemble	No	(1388, 347)	**7.9**	4.6

The comparison is not exact but seems to be fair as we compute the RMSE on the same number of segments. Note also that the errors of our models are computed on unseen data. Finally, the bad accuracy of the MLP might be due to the lack of architecture optimization and/or the amount of data.

3.7.2 Airport 2 and A320

For this airport and aircraft, the estimated errors of the MBA is 4.82 (RMSE 8-0NM) and 3.65 (RMSE 4-0NM). The analysis is the same for this experiment: the headwind at each segment helps diminishing the RMSE. Compared to the RMSE of the MBA, none of the model trained on dataset without headwind at each segment achieves the same performance. When we also take into account the headwind, the MBA is still better than our best candidate by around 2.7% for the 8 to 0 NM T2F task and have the same performance for the 4 to 0 NM task (Table 3.6). However, the maximum error of our model is lower (on average, all headwind conditions) as it can be seen in Figure 3.8.

Table 3.6: Applying top 3 feasible techniques on second case study.

Model	Headwind	Size	RMSE 8.5-0.5NM	RMSE 4.5-0.5NM
Lasso	Yes	(6753,1689)	4.8	3.8
Lasso	No	(7100,1776)	**7.3**	**4.9**
MLP	Yes	(6753,1689)	4.9	3.8
MLP	No	(7100,1776)	7.4	4.9
Ensemble	Yes	(6753,1689)	**4.8**	**3.8**
Ensemble	No	(7100,1776)	7.3	**4.9**

3.7.3 Absolute error results

The absolute T2F error for the MBA is computed versus the Ensemble method. The comparison is not exact but seems to be fair as we compute the RMSE on the same number of segments (Figure 3.8). The same has been done for the TAS. Note also that the errors of our models are computed on unseen data.

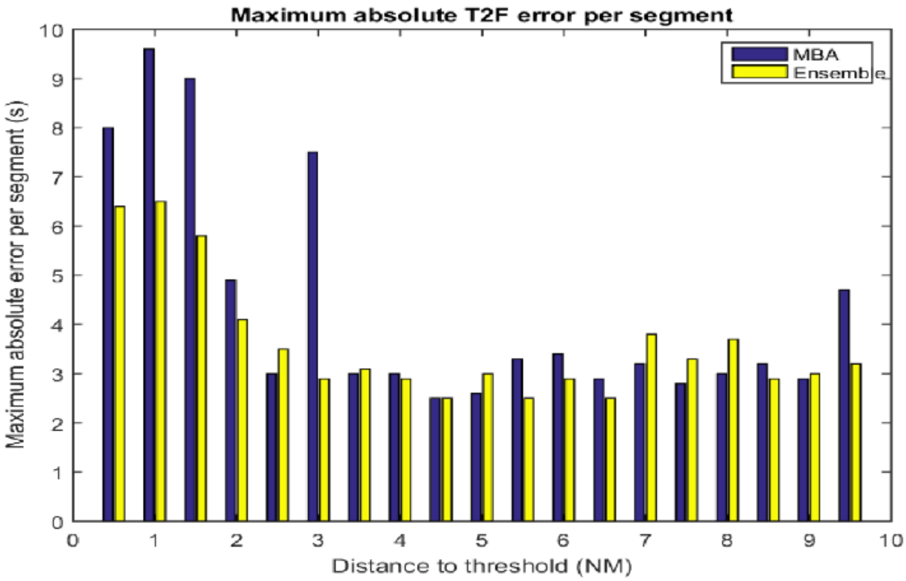


Figure 3.8: Maximum absolute error per 0.5NM segment.

Furthermore, the tool is able to calculate for the ML and MBA model for different flight cases (per aircraft type and Airport 1 and Airport 2), the MSE and standard deviation per aircraft type, wind-band and segment.

3.8 Conclusion

This study assessed feasible ML techniques on their performances for predicting the TAS and T2F. It can be concluded that by using the results of PCA and RreliefF before learning result in a lower MSE, lower sigma and same time compared to the results obtained without using these techniques. Our experiments show that PCA and RreliefF can discover strong dependencies between attributes, while in domains without such dependencies it performs the same as the MSE. It is also robust and noise tolerant.

Comparing the PCA and RreliefF MSE results using all the decision parameters compared to the first three groups, first two groups and first group result in respectively on average a 1% and 5% improvement and 2% reduction in MSE value for both T2F and TAS (Table 3.3).

From our experimental results we can conclude that learning multitask regularized regression with RreliefF is promising especially in combination with PCA. RreliefF's good performance and robustness indicate its appropriateness for feature selection.

Ground speed and other information at 10NM together with headwind information seem to capture a lot of the variation of the T2F and TAS in the last 10NM. According to Figure 3.2, the ground speed at 10NM is the most important feature whereby the headwind vector scores number two.

The multi task techniques Lasso and MLP turned out to be the best feasible and most accurate techniques for predicting the TAS and T2F from 8.5NM till 0.5NM and from 4.5 till 0.5NM. Combining these techniques result in a more robust and accurate ML model which is simply the average of the predictions of the Lasso and MLP - advanced model averaging techniques can be used to enhance the accuracy.

Stable MSE values are obtained when learning minimum 60 flights per aircraft type, wind band and distance from threshold. However when averaging the MSE per 0.5NM segment (10 till 0NM) we suggest a minimum of 400 flights per type and wind band.

Furthermore, outliers like NaN and 0 values will be excluded from the sample data set for analysing purposes. The dataset need to be carefully constructed and measured by analyzing the R value for a correct output of the model.

The ML techniques are more accurate and more robust to changes and they improve in overall over the accuracy of the MBA. We have seen that the standard error decreases with larger sample sizes since the estimate of the population mean improves.

It can be concluded that the optimal amount of neurons for MLP lies between 20 and 23 for respectively 5000, 10000 and 30000 flights in low wind. For high wind values the amount stays the same.

Table 3.5 and Table 3.6 shows that by learning a T2F ML model with HW, the MSE is significantly lower than without HW for both RMSE from 8.5-0.5NM and from 4.5 till 0.5NM. Furthermore the 4.5 till 0.5NM segment has a lower RMSE compared to RMSE from 8.5 till 0.5NM. Finally, the maximum error of our ensemble model is lower compared to MBA.

The results of this study are used as an input by SESAR and EUROCONTROL in the development of a new ATCO tool to predict aircraft speed performance. The Leading Optimized Runway Delivery (LORD) tool supports ATCO's to optimize the separation, the buffer and more efficiently and easily deal with the compression effect on the last part of the final approach.

The data supporting the above conclusions was obtained from 2 different airports. To improve verification the results were compared with data from Airport 2 and show significant similarities.

3.9 Recommendations

At this stage the ML tool is able to apply feature selection techniques and ensemble methods for calculating the MSE, standard deviation and amount of measurements for 30 aircraft types, wind-band and 2 different airports. For verification purposes more aircraft performance and weather data per airport should be considered where all airports count the same amount of flights during the same time period.

Looking at the most important prediction variable – GS at 10NM – might give some operational issues. For predicting the T2F in real life an ATCO has to wait till the aircraft is at 10NM. The T2F for an ATCO is interesting to calculate the compression on final approach using for example the TBS concept. The dynamic TBS for the follower aircraft needs to be known before 10NM. Therefore it is suggested to predict the GS at 10NM of the previous aircraft (based on historical flight information of that time period).

Learn new features such as sequential to visualize the main prediction variables that influence the T2F and TAS. Furthermore find a subspace that captures the variation of the data using PCA dimension reduction.

Learn one task at a time in order to see if the multi-task approach helps and validate that the multi-task approach lead to better results. Learn new ML techniques such as Support Vector Regression (SVR).

In this study the prediction parameters are used from the radar and METAR sets. As a next step the Flight Data Recorder variables will be included for the prediction of the responses, causalities and risks.

A more detailed analysis of the results is needed in order to emphasize the limits of the current approaches. Furthermore, an improved accuracy can be expected from fine tuning of the hyper parameters, network architecture optimization and multiple models averaging.

Acknowledgment

This work is supported by the EUROCONTROL Experimental Centre and the SESAR programme funded by EU, EUROCONTROL and the aviation industry. The authors would like to thank CDG and Vienna airport that provided access to the aircraft flight data. The author also wishes to acknowledge the work of the technical teams that made this work possible.

References

[1] Herrema, F. F., Treve, V., & Curran, R. (2015, September). Typical additional spacing-buffer to apply at 4DME for delivering separation minima. In 2015 IEEE/AIAA 34th Digital Avionics Systems Conference (DASC) (pp. 1F1-1). IEEE.

[2] Herrema, F.F. (2015). *Big data analysis and machine learning at airports to support decision making*. Brussels: p10-50.

[3] R.K. Ando and T. Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *J. Machine Learning Research*. 6: 1817–1853, 2005.

[4] B. Bakker and T. Heskes. Task clustering and gating for Bayesian multi–task learning. *J. of Machine Learning Research,* 4: 83–99, 2003.

[5] T. Evgeniou, C.A. Micchelli and M. Pontil. Learning multiple tasks with kernel methods. *J. Machine Learning Research*, 6: 615–637, 2005.

[6] T. Evgeniou, M. Pontil and O. Toubia. A convex optimization approach to modeling consumer heterogeneity in conjoint estimation. INSEAD N 2006/62/TOM/DS.

[7] T. Jebara. Multi-task feature and kernel selection for SVMs. Proc. of ICML 2004.

[8] A. Torralba, K. P. Murphy and W. T. Freeman. Sharing features: efficient boosting procedures for multiclass object detection. Proc. of CVPR'04, pages 762–769, 2004.

[9] K. Yu, V. Tresp and A. Schwaighofer. Learning Gaussian processes from multiple tasks. Proc. of ICML 2005.

[10] J. Zhang, Z. Ghahramani and Y. Yang. Learning Multiple Related Tasks using Latent Independent Component Analysis. NIPS 2006.

[11] J. Baxter. A model for inductive bias learning. *J. of Artificial Intelligence Research*, 12: 149–198, 2000.

[12] S. Ben-David and R. Schuller. Exploiting task relatedness for multiple task learning. Proceedings of Computational Learning Theory (COLT), 2003.

[13] Peel, T. (2016). Presentation of the models. In: Peel, T *PoC*. Brussels: d. p8.

[14] Hui Zou and Trevor Hastie. ``Regularization and variable selection via the Elastic Net''. In: Journal of the Royal Statistical Society, Series B 67 (2005), pages 301–320 (cited on page 10).

[15] Herrema, F.F. (2014). *Compression on final approach for optimized runway delivery*. Delft: R. Curran. p58-61.

[16] Serial metrics. (2013). *Machine Learning.* Available: http://serialmetrics.com/blog/what-is-machine-learning-feature-selection/. Last accessed 12-02-2016.

[17] MARKO ROBNIK S, & KONONENKO, I. (2013). *Theoretical and Empirical Analysis of ReliefF and RReliefF*. Available: http://link.springer.com/article/10.1023%2FA%3A1025667309714. Last accessed 12-02-2016.

[18] MARKO ROBNIK S, & KONONENKO, I.. (2003). An adaptation of Relief for attribute estimation in regression. (3), 9.

[19] EUROCONTROL (2015). *Optimized Runway Delivery*. Brussels: EUROCONTROL. 10.

[20] K. Gnana Sheela and S. N. Deepa. (2013). Review on Methods to Fix Number of Hidden Neurons in Neural Networks. *Mathematical Problems in Engineering*. 11 (2), p1-11.

[21] MatLab. (2014). *How can I know training data is enough for machine learning.* Available: http://stackoverflow.com/questions/24752941/how-can-i-know-training-data-is-enough-for-machine-learning. Last accessed 28th Feb 2016.

4.0 Taxi-Out Time prediction model at Charles de Gaulle airport

In the previous Chapter different ML techniques were assessed on their forecasting performance, computational time and the amount of data needed for delivering a T2F and TAS prediction. The regression technique MLP and Lasso allows the ATCO to refine separations for a wide range of environmental variables. However, the previous chapter is limited on fast and intuitive prediction. In this Chapter, a main focus will be given to Taxi-Out Time (TXOT) models that can be applied in real-time ATCO operations. The TXOT model is the second predicted ASP enabler for the RU support tool. We will start with how taxi-out time can be predicted by means of using the neural networks, regression tree, reinforcement learning, and multilayer perceptron methods. These four methods are assessed based on their performance indicators; applied to Charles de Gaulle operational taxi data and benchmarked against real-life taxi-out time profiles. The root-mean-squared error metric is chosen as the most important performance indicator. The regression tree is found out to be the most efficient method, which is then applied in a case study for predicting the TXOT and finding the key-related precursors extracted from the top 10 features.

This chapter was previously published as:

F.F. Herrema, R. Curran, H.G. Visser, D. Huet and R. Lacote. 2018. Taxi-Out Time Prediction Model at Charles de Gaulle Airport. Journal of Aerospace Information Systems, 15(3), pp.120-130.

Abstract
Taxi-out time predictions are a valuable asset in enabling efficient runway scheduling in real-time operations so as to reduce taxi-out times and fuel consumption on the airport surface. This paper will focus on how the neural networks, regression tree, reinforcement learning, and multilayer perceptron methods can be used for predicting taxi-out time. These four methods are assessed based on their performance indicators, applied on Charles de Gaulle operational taxi data. The root-mean-squared error metric is chosen as the most important performance indicator, which gives, for the applied regression tree method, on any given day, an average error of 1.6 min. The regression tree turns out to be the most efficient method, which is then subsequently applied in a case study for predicting the taxi-out time and finding the key-related precursors extracted from the top 10 features.

4.1 Introduction

Pressure on the taxiway requires clearly defined decision-support capabilities to maintain safety, reduce workload, and increase throughput and efficiency. In this context, the efficient deployment of a decision-support tool calls for a reliable prediction of taxi-out time (TXOT). In response to this need, this study presents a machine learning (ML) approach for coping with the variability of aircraft behaviors, and it is demonstrated how this approach contributes to the state of the art. The ML technique can be used to identify patterns and to observe precursors in past data. This study shows that the regression tree (RT) performs best by assessing the neural-network Levenberg–Marquardt (NNLM), RT, reinforcement learning (RL), and multilayer perceptron (MLP) techniques on their performance indicators. Therefore, a RT is derived based on different precursor features. Based on the state the airport is currently in, the distribution corresponding to the relevant leaf of the tree is used for TXOT prediction. Patterns in this context are defined by the RT, which gives per terminal leaf, a TXOT (Gumbel) distribution describing the relation between the TXOT and the precursor features. These patterns and precursors can be transferred into “what-if” statements by analysing relations between the target TXOT variable and the precursors. This analysis is needed to predict forthcoming operational risks during real-time operations, such as an increased TXOT, based on the identification of precursors that lead to operational issues and to reduction in efficiency. Such a prediction would feed what-if online tools at the airport to alert air traffic controllers (ATCOs) about impending aircraft behaviors and to produce both point forecasts and probabilistic forecasts in real time.

4.1.1 Related Work

Prior studies reported in the literature have attempted to predict TXOT and observe related risks precursors. We divide the literature in two groups. We start with the first group, which focuses on statistical detection methods for the prediction of TXOT. In [1], predictive queuing models were developed to estimate the TXOT from gates to the departure runways. The paper presented in [1] proposed simple statistical linear equations derived from regression analysis using the airport surface detection equipment (ASDE). NASA and American Airlines are jointly developing a decision-support tool that assists runway schedulers in making gate pushback decisions and helps to improve the overall efficiency of airport surface traffic [2]. Shumsky developed a model to predict TXOT using airline departure demand and departure runway selection as explanatory variables [3]. In [3], a queuing model was also developed for the runway service process. However, the proposed queuing model was based on cumulative behavior and did not reflect the stochastic nature of the process. Idris et al. [4] analysed the main causal factors that affect taxi times on factors such as queue size, runway configuration, weather, aircraft type, downstream traffic restrictions, and starting terminal. Based on this analysis, a statistical regression model to predict taxi times was developed. The work presented in [4] did not explicitly model the runway service process, and it required knowledge of the number of aircraft on the ground in order to predict TXOT. It was concluded that the take-off queue length was the most significant explanatory variable to predict TXOT. Basic statistical approaches have been introduced [5,6] to predict departure taxi times at Jackson Atlanta International Airport (denoted ATL) and New York's John F. Kennedy Airport (denoted JFK) using ASDE-X (model X) surface surveillance data. Also, linear regression models were used for modelling aircraft TXOT at Dallas/Fort Worth International Airport (denoted DFW) with several independent variables such as, airlines, taxi distance gate group, and the number of departures and arrivals [7,8]. The EUROCONTROL Performance Review Unit (PRU) developed a method [9] that predicted the TXOT based on the unimpeded TXOT (UTXOT). Because the study presented herein is supported by the EUROCONTROL PRU, we use their method [9] to characterize and predict the TXOT profiles. The method will be explained and used in Section 4.3. It is noted that the approaches considered in this first group did not assess the predicted TXOT target variable on many aircraft operational taxi variables from runway schedulers, such as the number of departures and the time of the day. To end up with a selected number of features (assumed here to be the top 10), we apply off-the-shelf feature selection techniques.

The second group of literature consists of feasible ML methods to predict aircraft performances on the taxiway system and on final approach. In previous papers [10,11], ML techniques like stochastic gradient boosting, neural networks (NNs), principal component regression, multiple linear regression (MLR), and least absolute shrinkage and selection operator (known as LASSO) were assessed in relation to their forecasting performance, computational time, and amount of data needed for delivering a reliable time to fly, as well as aircraft final approach mass and speed prediction. For the target variable TXOT, NN and RL algorithms were assessed based on their TXOT prediction accuracy at several major airports in the United States [12–16], such as Dallas Fort Worth Airport (DFW). In addition, regression methods, including MLR, least median squared linear regression, support vector regression, model trees, and fuzzy-rule-based systems were also applied to several European airports to predict TXOT [17,18]. Each ML method was independently applied to limited test data under different conditions, and therefore the prediction performance varied with the prediction model. Furthermore, state-of-the-art predictions of TXOT have relied on the ML techniques, viz., NNs, linear regression models, support vector machines, k-nearest neighbors, and random forest. These methods have, for example, been applied on limited traffic data for Charlotte Airport (denoted CLT), and their prediction performances were evaluated by comparing the predicted TXOT with the simulated taxi times, as well as with the predictions from fast-time simulations [19]. Based on a literature review, the following shortcomings in TXOT prediction methodology were identified; first, extracting top TXOT features using many aircraft operational taxi variables and feature selection techniques has, as yet, not been considered. Second, applying top features before training the model has not been explored. Third, as yet, no algorithm has been developed that is capable of real-time computation and providing the distributions associated with the individual TXOT predictions to synthesize the what-if statements. In this study we apply the state of-the-art ML algorithms NNLM and RL, in additional to the RT and MLP methods, which are all capable of synthesizing the what-if statements and are expected to overcome the noted lacking aspects from literature. In [20], the characteristics of these techniques were reviewed, it was explained how they were feasible for real-time and what-if statements computations.

4.1.2 Aim

The main purpose of this study is fourfold. First, we will analyse and review five years of historical operational taxi data, allowing us to better characterize and predict TXOT as a function of different runway–stand (RWY-STD) combinations and congestion levels. Second, we will identify key processes that might benefit from data-driven ML predictions. Third, we will extract the most efficient ML method based on their performance indicators; finally, we will improve TXOT prediction quality and demonstrate this by comparing our model with state-of-the art predictions. The key objective of this study is to develop a real-time model that forecasts the TXOT and congestion levels for different RWY-STD combinations using machine learning techniques. This model enables us to identify the key precursors impacting TXOT.

4.1.3 Structure

The structure of this paper is as follows: first, the data and prediction variables are described in Section 4.2. Next, the employed methodology is outlined in Section 4.3, and it includes the description of the TXOT behavior and data preparation. Next, the ML techniques are applied and evaluated, and the results are presented in terms of performance indicators, identified precursors, and a prototype model. In Section 4.5, a comparison of the prototype model with baseline models is performed. Finally, recommendations and conclusions are drawn in Section 4.7 and 4.8.

4.2 Data and Prediction Variables

To predict TXOT profiles and extract what-if scenarios, operational taxi data are used.

4.2.1 Operational Taxi Data

Aircraft operational taxi data are extracted from recorded runway scheduler data that have been provided by Charles de Gaulle (denoted CDG) airport. The data cover five years of taxi-out and taxi-in records from 2011, up to and including 2015. In total, the data comprise records of about 1,000,000 arrival and departure flights. The datasets are stored in comma separated values (CSV) formats and are thereafter saved in separate MATLAB files (.mat).

4.2.2 Prediction Variables

Table 1 lists all the 42 input (potential) prediction variables related to the target variable TXOTs that have been identified; TXOT is listed as entry number 43 in Table 4.1.

Table 4.1: Prediction and target variables.

Variables	Description	Variables	Description
1. AOBT	Actual Off Block Time	**23. Day**	Day of the year
2. ASAT	Actual Start Up Time	**24. Month**	Month of the year
3. ATOT	Actual Take Off Time	**25. FlightNumber**	Flight Number
4. Year	Year	**26. DepartureStand**	Departure stand
5. DeIcingStand	De-icing stand	**27. QFU**	Runway Orientation
6. CTOT	Calculated Take Off Time	**28. SOBT**	Scheduled Off Block Time
7. Caractredevol	Commercial or private flight	**29. TOBT**	Target Off Block Time
8. CodeIATA	IATA code company	**30. TSAT**	Target Start Up Arrival Time
9. CodeAirportICAO	Airport destination ICAO code	**31. Terminal**	Terminal departure
10. CodeAirportIATA	Airport IATA code	**32. TimeATOT**	Date and Actual Take Off Time
11. Airline	Airline	**33. TimeSchema**	Schema time
12. DataPointSpeed	Point where speed is measured	**34. TimeReal**	Actual time
13. DateReal	Actual Date	**35. AircraftType**	Aircraft type
14. DateSchema	Schema Date	**36. AircraftTypeICAO**	Aircraft type ICAO
15. DepArr	Departure or arrival flight	**37. GateConnectionType**	Connection Type at the gate
16. EOBT/EIBT	Estimated Off Block Time/ Estimated In Block Time	**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 off-block and

			estimated take-off) of the respective flight.
17. ETOT	Estimated Take Off Time	**39. NumberOfDepartures**	Number of departures in last 20 minutes observed when a flight is at ATOT on a specific runway.
18. DeIcingStatus	De-icing status	**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.
19. ScheduleBloc	Actual In Block Time (AIBT)	**41. SaturationLevel**	For explanation see section III.A
20. SOBT/SIBT	SOBT or Scheduled In Block Time (SIBT) - depending if it is a departure or arrival	**42. StandsGate Availability**	When a flight is at AOBT the number of not used stands are counted
21. ATCcallsign	ATC call sign	**43. TXOT**	Taxi-Out Time
22. RegistrationCode	Registration code		

4.3 Proposed Methodology

This study proposes a methodology comprising five steps. The method is based on the Statistical Package for the Social Science ML method [21]. This methodology describes the steps to come up with a usable predictability model. Each step is described in the following:

4.3.1 Compute the TXOT

Before feasible ML techniques can be applied, the TXOT response is extracted. One of the purposes of the TXOT indicator is to provide an accurate prediction of the average outbound queuing time during times that the airport is congested. Taking into account the timestamp data available, TXOT is defined as the time elapsed between actual offblock time (AOBT), from a specific stand, and the actual takeoff time (ATOT), on a specific runway [22]. This time envelope covers both systemic durations (e.g., time spans for certain procedures, queuing at runway to ensure flight demand) as well as additional time aspects linked to the actual progress of the operations. Therefore, in this step, we first plot the TXOT versus the congestion level for all 1730 RWY-STD combinations at CDG. The congestion level is defined as the estimated number of movements (i.e., arrivals and departures) at CDG within the estimated taxi-out transit time (i.e., time between offblock and takeoff) of the respective flight. The number of movements and the taxi-out transit time are estimated because we start the prediction from the AOBT, for which we use the variables of estimated offblock time (EOBT), estimated takeoff time (ETOT), and estimated in-block time (EIBT). Furthermore, we only include the number of movements of the dependent runways from the TXOT flight considered. Figure 4.1 shows an example for the TXOT from stand C-E32 to runway 26R. Note that the TXOT and congestion levels are rounded off in, respectively, minutes and the number (#) of movements. We observe from Figure 4.1 that, as soon as a certain level of congestion is reached, the TXOT increases linearly (purple line denotes linearly fit) with the congestion level. At a low level of congestion, the actual taxi-out time tends to be constant (horizontal red line). This constant actual TXOT is considered to be the UTXOT (explained in Section 4.3.2) required by any flight to taxi out and take off. The intersection between the red and purple lines corresponds to the saturation level. Beyond this saturation point, the TXOT is directly proportional to the number of movements.

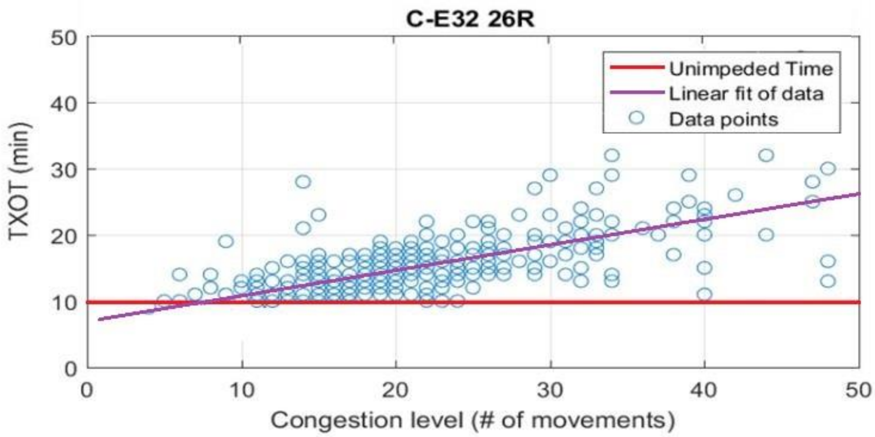


Figure 4.1: TXOT versus the congestion level for stand C-E32 and runway 26R.

4.3.2 TXOT Understanding

An analysis is conducted in order to extract additional prediction variables that affect TXOT behavior. We observed from Section 4.3.1 and the EUROCONTROL PRU methodology [9] that the following prediction variables are highly influencing the TXOT: "congestion level," "number of departures in last 20 min," "saturation level," "unimpeded TXOT per runway stand and stand group," and "stands gate availability." These variables are not included in the received historical operational taxi data, and are therefore calculated and included into Table 4.1 as additional prediction variables (numbers 38–42). The duration of the TXOT is captured using a statistical analysis for periods of low traffic [9], referred to as UTXOT. The TXOT can be predicted more accurately by first calculating UTXOT and the additional taxi-out time based on historical operational taxi data. The indicator is first calculated at a disaggregated level, i.e., for a comparable grouping of flights characterized by the same combination of RWY-STD. Each grouping of flights has an unimpeded reference associated with it. Thereafter, we benchmark our UTXOTs per RWY-STD combination with the UTXOT results of the PRU. Table 4.2 shows four examples where we benchmark the UTXOTs per runway–stand group from our model with the PRU model. For example, stand group A1 includes the stands A01 up to A18, and stand group A2 includes stands A30 up to A38. We observe that the results from the PRU are similar, except for stand group B1.

Table 4.2: Comparison of PRU UTXOT with our UTXOT results.

Runway	**Stand group**	**UTXOT calculated with our model**	**UTXOT PRU**
08L	A1	12:06 min	12:00 min
08L	A2	11:29 min	11.30 min
08L	B1	10:22 min	10:00 min
08L	B2	10:31 min	10.30 min

4.3.3 Data Preparation

The data preparation phase covers all activities required to set up the final dataset from the initial raw aircraft operational taxi data; also, the taxi data are merged and cleaned in this step. The tasks include feature selection for identifying which features are important for best describing the variance in a dataset. For feature selection, the RreliefF and Sequentialfs technique is applied. In previous papers, these techniques were introduced [10,11]. The objective of feature selection is threefold: improving the prediction performance of the predictors, providing faster computational performance and more effective predictors, and providing a better understanding of the underlying process that generated the data [23]. RreliefF and Sequentialfs have commonly been viewed as feature selection methods that are applied in a preprocessing step before the model is learned [24]. The standard RreliefF regression modeling technique has been extensively discussed in several papers [23–26]. The technique has been applied on 500,000 TXOT flights for CDG, as shown in Figure 4.2. The results for the Sequentialfs technique are shown in Figure 4.3.

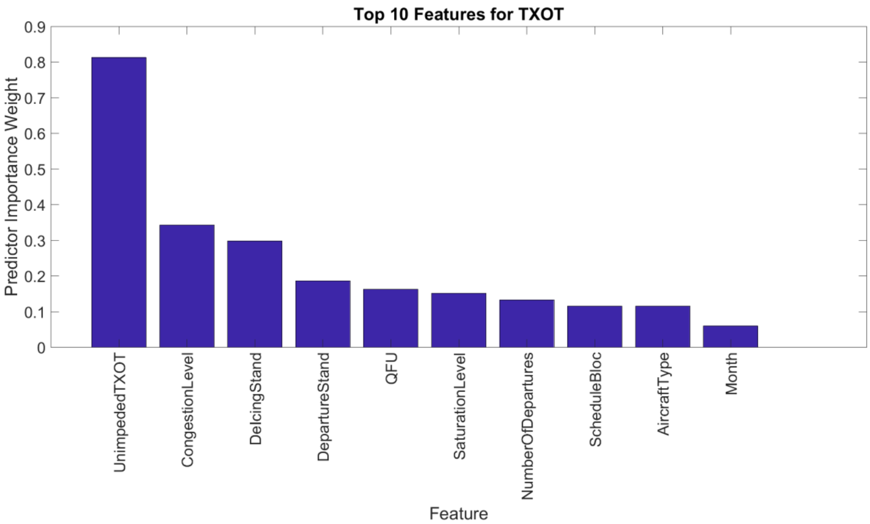


Figure 4.2: Normalized feature selection using RreliefF algorithm (see Table 1 for feature/variable definitions).

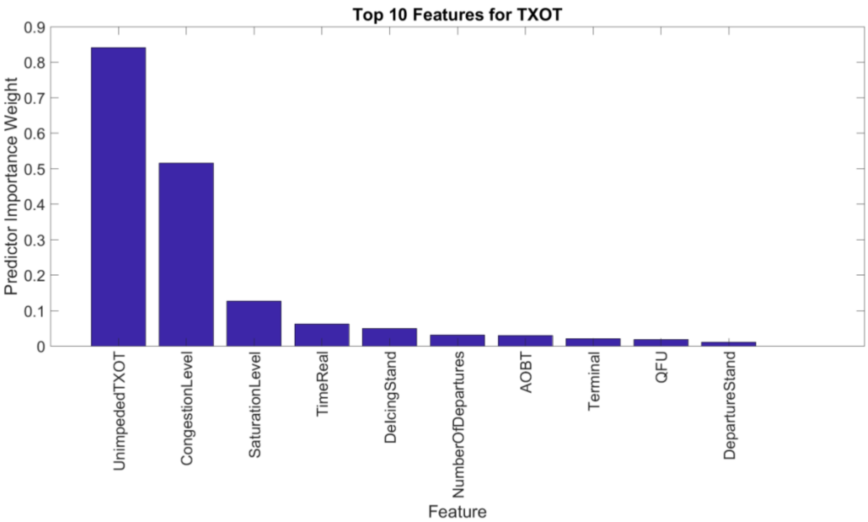


Figure 4.3: Normalized feature selection using Sequentialfs algorithm (see Table 1 for feature/variable definitions).

Applying the "intersection" method on the features selected from Figure 4.2 and Figure 4.3 results in the following 10 most important features: "unimpeded TXOT," congestion level, "Saturation level, number of departures in the last 20 min," "deicing stand," "month," "time real," "departure stand," "QFU," and "AOBT." These 10 features are included in the ML model that we have developed. The main reason for opting for fewer variables has already been explained by the feature selection objective described earlier. Next to this, there are more advantages associated to learning a ML model with only top features instead of including all prediction variables, as pointed out in [26].

4.3.4 Evaluation of Feasible Machine Learning Techniques

The MLP, RT, RL, and NNLM modelling techniques are applied and assessed in relation to their performance indicators in order to come up with an accurate TXOT ML model. Furthermore, we construct the datasets and find the stability of three different data parts: 1) Based on different data sources and the variables listed in Table 4.1, we standardize the feature matrix X. 2) We split matrices X and Y into two subsets (Xtrain, Ytrain) used to train the model, and (Xtest, Ytest) are used to evaluate the model accuracy. 3) The default ratios (splitting the data) for training, testing, and validation are analysed. Before the forecasting performance, computational time, and minimum amount of data needed for each applied ML technique are analysed, the stability of three different data parts is checked first (cross validation). To check the stability of different data parts, the data will be randomly divided into training, validation, and testing subsets. It is assumed that the default ratios in this study for training, testing, and validation are 0.70, 0.15, and 0.15, respectively. The model is adjusted accordingly during training. The validation is used to measure network generalization, as well as to halt training when generalization stops improving. To prove that a randomly selected dataset is stable, epoch and validation checks are performed. The number of epochs indicates the number of single passes through the entire training set. Thereafter, we check convergence on the validation set and, at the end of the learning process, the model is evaluated on the test set. The test has no effect on the training, and therefore provides an independent measure of network performance during and after training. Figure 4.4 and Figure 4.5 show an example of a trained TXOT LMNN model selecting 250,000 CDG flights from all stands to all runways. We learn the model with all prediction variables listed in Table 4.1. It has been tested that similar mean squared error (MSE) results are obtained using the top 10 features from the previous section. However, by excluding 32 variables, the model is trained three times faster and is more robust when inserting new data with similar structure. Also, the ML TXOT prediction error results show similar statistical TXOT prediction errors [17]. Assessing the NNLM technique result in a root MSE (RMSE) of 1.97 min (MSE 3.89 min) for 79% of the predicted TXOT flights, whereas an approximately 5 min RMSE is obtained for 98% of the cases.

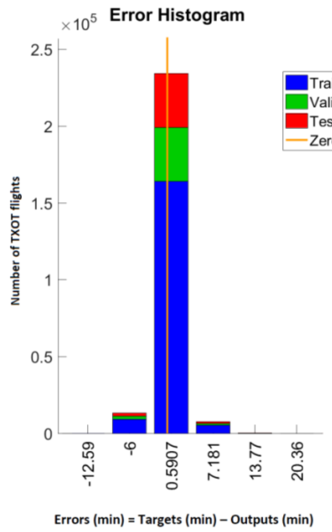


Figure 4.4: Number of TXOT flights (instances) versus errors in minutes for 250,000 TXOT flights.

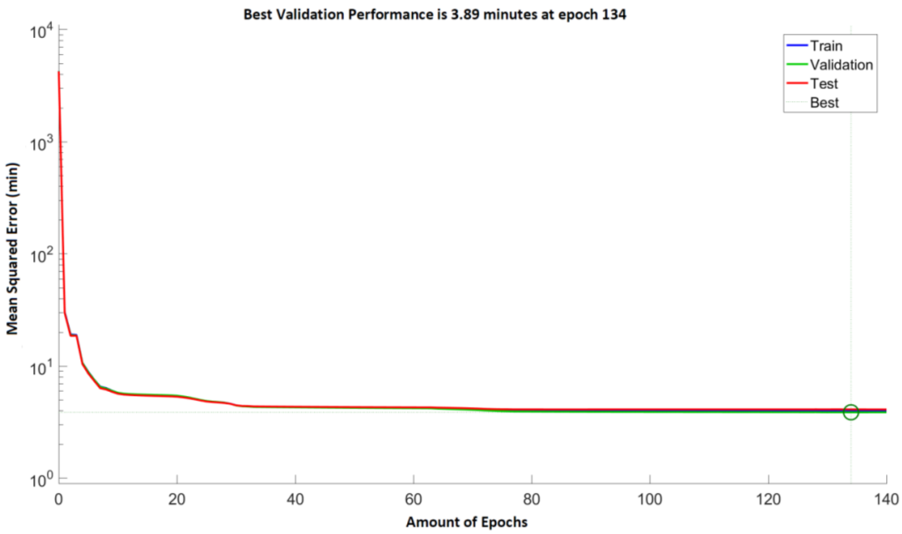


Figure 4.5: MSE in minutes versus amount of epochs for the NNLM technique and 250,000 TXOT flights.

Outliers

Outliers exist when building predictive models. Outliers are unusual data points (TXOT flight) that are far removed from the other data points. One way to show this inconsistency is by plotting the linear regression lines for the training, validation, and test sets, as well as for the complete set (all). The correlation coefficient values R represents a measure of the correlation between the predicted outputs and the TXOT target. An R- value of close to one means a close correlation and an R- value below 0.2 is defined as an outlier. It is evident that, by neglecting outliers in the target set, a better R value will be obtained for the prediction model. However, the TXOT model conceived herein takes all the outliers into account.

Results of Neural-Network Levenberg–Marquardt, Multilayer Perceptron, Regression Tree, and Reinforcement Learning Modeling

The same procedure has been done for the MLP, RT, and RL techniques. For the various techniques, Table 4.3 shows the minimum RMSE of TXOT in minutes, the computational time in minutes, and the minimum amount of data needed to obtain these results. We observed that the RT and RL techniques performed best in terms of the RMSE. However, due to a lower computational time and amount of data needed, we selected the RT as the most efficient method for TXOT prediction. The sequel to this paper will exclusively focus on the results obtained using the RT approach.

4.4 Results Regression Tree Modelling

Based on the results listed in Table 4.3, the regression tree modeling technique, by learning the tree based on the top 10 features extracted in the previous section, is further explored. The purpose of building a RT is to extract a set of if-then-else (what-if statements) split conditions in order to identify the main precursors that are mostly influencing the TXOT. After having built this tree, we start at the root node and ask a series of questions about the predictors. In each subsequent node, the tree selects the variable and the split point to achieve the minimum MSE between predictions and actual TXOT. This process will continue until a stopping rule is applied. Each of the terminal leaves represents one of the partitions of the input space. To provide a model that can generate accurate predictions that are not overcomplicated, we need to find the optimal tuning parameters for the tree. In this study, we use two parameters. The first parameter is the minimum leaf size lmin, for which we need enough data points in each terminal node to create a distribution. The parameter minimum leaf size can be used to stop the splitting process when the number of instances in a leaf is too small. In addition, if the tree contains too many variables, it is hard to interpret. The second tuning parameter for the tree is the maximum tree depth dmax. Avery 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 10 feature variables. The maximum tree depth can restrict the number of layers of a tree. Cross validation is used to select the minimum leaf size lmin, and MSE is used to select the maximum tree depth dmax. So, in our case, the tree is fit for a range of values of the two parameters based on three-quarters data. Thereafter, the MSE of the predictions is computed based on the remaining one-quarter. This is done for each quarter of the data, and the four MSE values are averaged. The set of parameters that gives the lowest MSE will be selected. As shown in Figure 4.6, we first train the trees with all 42 variables and different settings of dmax and lmin. We observe that the MSE drops as the tree depth increases from one to seven, regardless of the leaf size. Once the tree depth reaches the value of six, the MSE does not change significantly. On the other hand, a tree with 4000 minimum leaf sizes performs slightly better than the trees with 5000 and 6000 minimum leaf sizes. We also explore setting the minimum leaf size to less than 4000, but the model does not appear to improve much. Moreover, if we further reduce the leaf size, we may not have enough instances in the leaves to fit a distribution. Thereafter, we also train a model with the top 10 features found in Section 4.3.3, where we fit a tree to the entire dataset with the maximum tree depth and minimum leaf size set to 6 and 4000, respectively. We then sort the predictors based on their feature importance and select the first 10 as the final predictors. Thereafter, we retrain the tree with these 10 variables. We change the values of dmax and lmin, and we repeat the cross-validation process described previously. The tree with lmin equal to 4000 still performs slightly better than the others, and the MSE does not change significantly for tree depth values above six. Thus, our final model has 10 predictors and is fitted with a maximum tree depth and minimum leaf size set to 6 and 4000, respectively. By learning the tree, a mean and distribution are extracted per decision node. This is needed to observe precursors and understand what is likely to happen for the TXOT. Our model divides all the flights into 61 segments. In other words, the RT shown in Figure 4.7 has 61 terminal nodes and is learned with data from stand groups A, B, and C to runway 08L. Here, we will identify the most important predictors as the major factors that play key roles in influencing TXOT. The five most important factors are as follows: 1) The first factor is whether or not the flight experiences unimpeded conditions, which is the most important predictor in our model. The key decision value in our model is 21.5 min (node 1 in Figure 4.7). In the real-time prediction tool that has been conceived, a flight is considered unimpeded if the actually observed TXOT remains sufficiently low for that specific RWY-STD. 2) The second factor is the congestion level, which is mainly influenced by the time of the day and RWY STD; it is estimated in real time using the EOBT, ETOT, and EIBT within the estimated taxi-out transit time. 3) The third factor is the saturation level [9], which is predicted based on the top 10 features and estimated congestion level. 4) The fourth factor includes the time and month. We observe that, during three time windows of the day, the probability of experiencing congested conditions is significantly higher as compared to other time windows. The key decision values range from 0800 to 0930 hrs, 1230 to 13:00 and 17:00 to 18:00.

Table 4.3: Four feasible ML techniques assessed on their performance indicators for runway 08L 1400 hrs, and 1800 to 2000 hrs.

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 min	01:06 min	Yes	Yes	70,000
Regresion Tree	01:36 min	01:20 min	Yes	Yes	70,000
MLP	01:42 min	01:50 min	Yes	Yes	110,000
Reinforcement learning	01:36 min	01:30 min	Yes	Yes	150,000

Furthermore, cold conditions or winter conditions are negatively influencing the TXOT. Therefore, our decision variable of the de-icing stand can also be found in the top 10 features. 5) The fifth factor is the number of departures in the last 20 min, which is measured before the ATOT and estimated using the ETOT. The decision value lies at 15 flights.

Figure 4.7 shows what-if statements. If the statement (<) is true, we go to the left node at the next level; if the statement is false, we go to the right node at the next level. We also fit a parametric distribution to each terminal leaf. The probability distributions we consider include the Gumbel, Gamma, and F distributions. Equation (1) shows the Gumbel distribution, which provides the best fit over the terminal leaves:

$$f(x) = \frac{1}{\beta} e^{-\left(\frac{x-\mu}{\beta} + e^{-\frac{x-\mu}{\beta}}\right)} \qquad \text{Equation 1}$$

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

Figure 4.8 shows the Gumbel distributions fitted to the 61 terminal leaves. The blue bars represent the histogram of the 250,000 TXOT flight records in our training set. We note that the distribution of all the TXOT values (in the training, validation, and test sets) are more spread out than the distributions of the terminal leaves. The shapes of the terminal leaves' distributions are quite different from each other. In general, the distributions with lower medians are less spread out. This indicates that, in these segments, the uncertainties of the TXOT flights are low. If there are a lot of TXOT flights in these segments at the airport, the managers should have more confidence in making adjustments to their plans.

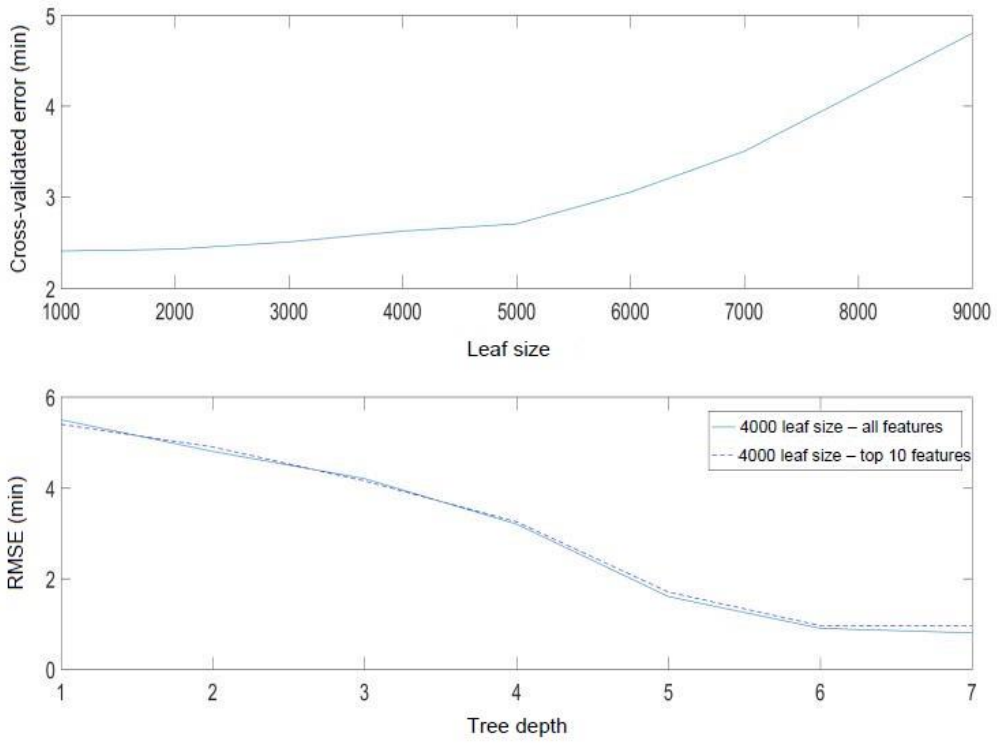


Figure 4.6: RMSE versus tree depth for different leaf sizes and features.

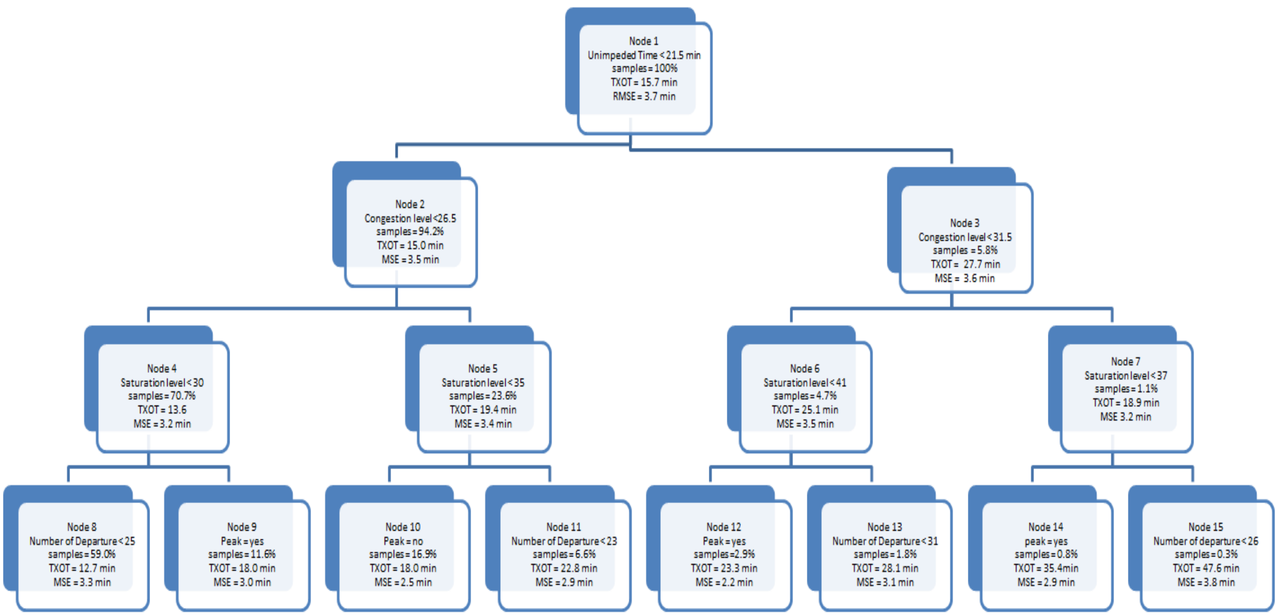


Figure 4.7: Part of the regression tree with a tree depth of four.

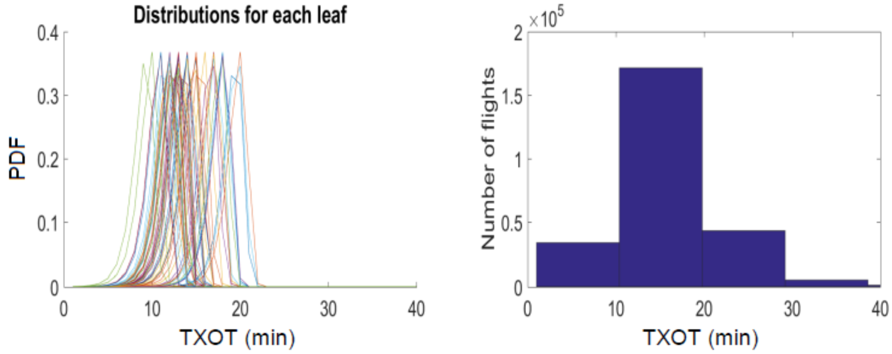


Figure 4.8: Distributions of the 61 terminal leaves and the number of flights (PDF, probability distribution function).

4.5 Prototype Model

Based on what we have learned in the previous steps and the data availability, a prototype model has been developed using the RT method (best performance) to forecast TXOT at CDG. We run our model, given the input variables from our case study, which are runway 08L, all stands, a forecast window of 120 min, 1500 simulations, and a forecast resolution of 5 min. The model is built on 500,000 flights collected over three years of operational taxi data. To generate real-time predictions based on the proposed model, an application based on MATLAB is developed. The output from running the application includes the mean and quantiles of flights. The aim is to generate TXOT forecasts for each flight and of the number of aircraft assigned to a given runway per time window. Suppose we are at time instance h and try to make predictions for the next x min. Given real-time flight information before AOBT, our RT tree model will determine to which end leaf the flight corresponds. For example, if a flight departs at terminal 2 and plans to take runway 08L with an ETOT of 0850 hrs, then this flight will fall into leaf 10. Thus, the median of the TXOT is 17.2 min and the distribution of the connection time can be described by a Gumbel distribution with $\mu = 17.9$ and $\beta = 2.3$. Next, we produce the distribution of the number of TXOT aircraft during a certain time interval [h1; h2], where $h < h1 < h2 < h + x$. This distribution is obtained by aggregating all the distributions of the flights that taxied out in the last 2 h or will depart in the next x min. The procedure of generating this distribution is summarized in two steps: 1) Suppose there are n flights that taxied out in the last 2 h or will taxi out in the next x min. We sample one TXOT from each of the n flights' distributions, and we calculate the time when the aircraft concerned arrive at the runway. We then count how many flights taxied out between the time interval [h1; h2], and we record this number as y1. 2) Repeat step 1m times and construct an empirical distribution using y1, y2..ym. Then, the qth quantile of the number of flights taxied out between the time interval [h1; h2] can be approximated by the qth quantile of y1, y2..ym. In a live trial, we produced the distributions of the TXOT of the flights who have departed in the last 2 h or will depart in the next x min. The operational datasets can be accessed in real time with exception of the actual variables AOBT, ASAT, and ATOT (see Table 4.1). For this trial, we generate real-time data for a selected historical data set of two years, in which we exclude the actual data variables. To conveniently generate predictions in real time, we develop a standalone MATLAB compiler that can work in most operation systems (Windows, Linux, Mac, etc.).

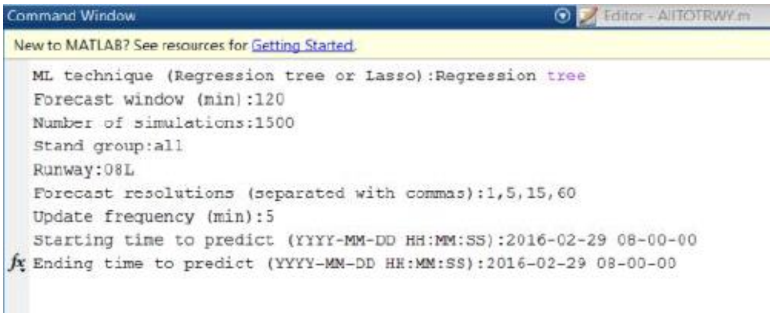


Figure 4.9: Interface of application.

Figure 4.9 shows the interface of the application. This application allows users to set the forecasting window x, the number of simulations m, the update frequency min, the forecast resolution r, the runway at CDG (RWY), the stand group, the starting time of the first forecasting window h, the machine learning technique (ML), and the ending time of the last forecasting window. The default settings of the first three parameters are 120min, 1500 simulations, and 5min, respectively. We update the predictions every 1 min because it takes slightly less than 1 min to produce the forecasts for the upcoming 2 h. The default resolutions are 1, 5, 15, and 60 min; and the predictions are saved at different resolutions in different CSV files. The starting time defaults to the current time if the user does not specify one. The ending time will be 24 h after the starting time. As shown in Figure 4.10, the predictions for this case study are generated on a rolling basis from all stands to runway 08L. Suppose the trial started at 0800 hrs. We first collected data of the flights that departed at CDG after 0600 hrs or would depart before 1000 hrs, and then we generated forecasts for the next 2 h (0800–1000 hrs). Thirty minutes later (0830 hrs), the second trial started. Similarly, we only considered the flights that departed at CDG after 0600 hrs or would arrive before 1030 hrs, and we generated forecasts for the time interval between 0830 and 1030 hrs. The numerical values attached to the dots in Figure 4.10 show the difference between the predicted TXOT and their real values. The difference is measured in minutes per stand group. Each stand group attached under the dots represents a number of different stands. For example, stand group A1 includes the stands A01 up to A18. It has been observed that, for this case study and for each prediction trial, the first 30 min have a significantly lower error as compared to the prediction for the remaining 60 min timeframe. Table 4.4 summarizes these differences in percentages for each trial. We observed an increase in error because of the longer look ahead time, which was not due to the increased number of flights.

4.6 Discussion on TXOT Results

To the best of our knowledge, there is no existing TXOT prediction function to compare against. In Table 4.5, we built a lookup table of baseline models (average TXOT between 8 and 14 min) to compare our CDG TXOT predictions against. The results of our case study include a final model that has 10 predictors and is fitted with a maximum tree depth and minimum leaf size set to 6 and 4000, respectively (Figure 4.6). The percentage of predictions that fall within approximately 3 and 5 min is used as metric to compare the TXOT prediction. Table 4.5 shows the RMSE TXOT prediction results for six different models. The models do not take into account 1) the layout of the airport (particularly by not considering the factors associated to the distances and the turning angles), 2) the time period and amount of data needed to learn, 3) the computational time needed for a prediction, and 4) the instances at which the prediction is performed. An average of 95.7% of predictions was found for Detroit International Airport (denoted as DTW), and an average of 93.8% was found for Tampa International Airport (denoted as TPA) for an approximately 3 min accuracy. The results for John F. Kennedy International Airport (JFK) were not very consistent and much less promising, showing an approximately 5 min prediction accuracy between 20.7 and 100% for different days and parts of the day. Additionally, [17] predicted 96.1% of the TXOT at Stockholm-Arlanda airport within 3 min of the actual time and 99.2% within approximately 5 min. In contrast, our RT model found an average of an approximately 3 min accuracy of 94% for CDG and 99% within 5 min, considering both departures and arrivals simultaneously. Based on these results, it can be concluded that the Arlanda and Zurich airport cases performed slightly better than our model in terms of the average RMSE. However, the CDG model proposed herein is more complex in terms of operations, and over 250,000 movements were used to learn the model; whereas the other models only took a day or a week of operation to learn the model.

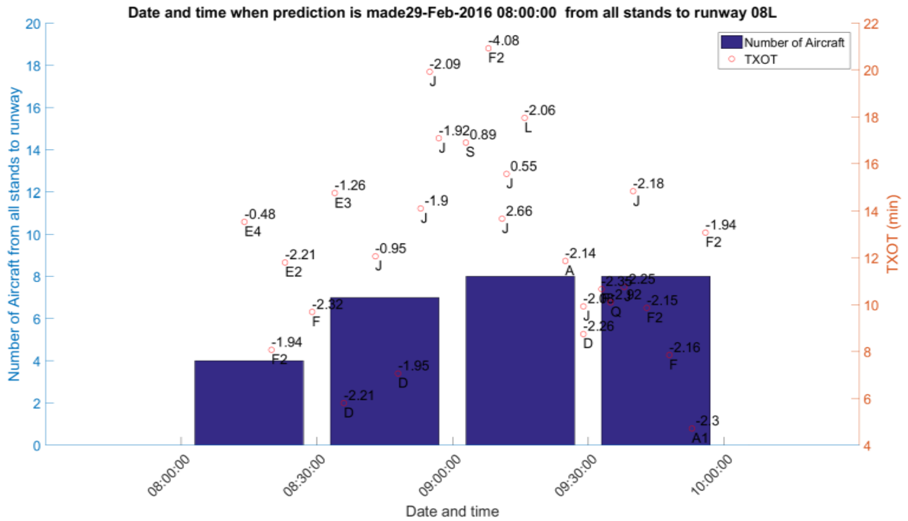


Figure 4.10: Predicted output after running the application for the first trial.

Table 4.4: Average error differences per trial and for different time prediction windows.

Time prediction window	**0 - 30 minutes**	**30 - 90 minutes**
Trial 1	6%	10%
Trial 2	4%	8%
Trial 3	4%	9%
Trial 4	2%	6%

Table 4.5: RMSE TXOT prediction within approximately 3 and 5 min for six different models.

	Within 3min	Within 5min	Median TXOT (min)
Charles de Gaulle Airport	94.2%	99.0%	17
Tampa International Airport [15]	89.9%–95.7%	-	12
Stockholm-Arlanda Airport [17]	96.1%	99.2%	11
John F. Kennedy International Airport [6, 17]	-	20.7-100%	37
Detroit International Airport [16]	89.9%–97.1%	-	15
Zurich Airport [17]	95.6%	99.4%	14

Furthermore, our model is the first TXOT model that illustrates how shared data and advanced analytics can be used to the benefit of ATCOs and runway schedulers.

4.7 Recommendations

Airport operational rules, regulations, and standards can vary significantly over a time period of five years. Therefore, in future development of the conceived TXOT model, the utility of a faster trend-tracking model will be explored by reviewing the prediction accuracy of the model on a day-to-day basis as compared to a static model based on the full five years of data. Additionally, weather events, maintenance downtime, and runway/taxiway repair will also be included in our TXOT model. Although the model has been developed for a TXOT prediction problem, we believe the methodology proposed in this study can be easily applied to other runway processes as well, such as the prediction of runway occupancy times. This topic will be explored in future research.

4.8 Conclusions

This study demonstrates the use of machine learning (ML) techniques to forecast the taxi-out time (TXOT) per flight. First, the TXOT aircraft behaviors, the data sources, the key features, and the ML outcome are reviewed. Based on the availability of data and the importance of the

TXOT prediction, activities that will benefit from making greater use of data are then identified, where the focus is on the TXOT from all stands to runway 08L. A predictive model was built for TXOT flights using the neural-network Levenberg Marquardt, multilayer perceptron, reinforcement learning, and regression tree techniques. Then, an approach was developed to generate distributions associated to each TXOT flight and the number of aircraft to a specific runway (08L) within a timeframe of 30 min. An application was also developed for CDG to produce these forecasts. Finally, a live trial at CDG on 14 November will be run, for which the accuracy of the model will be assessed and improvements will be made. To this end, a feasibility study will be conducted where an analysis will be made of how the prediction tool can be used by air traffic controllers in their decision making and planning to ensure resilience, safety, and efficiency of air traffic control operations locally in a sector, which will also take into account coordination with other sectors. The following 10 features were included in the ML model after the intersection method was applied on our feature selection results: unimpeded TXOT, congestion level, saturation level, number of departures in the last 20 min, deicing stand, month, time real, departure stand, QFU, and actual offblock time (AOBT). It was observed that the RT technique performed best; for this regression technique, a maximum tree depth and minimum leaf size was adopted of 6 and 4000, respectively. Furthermore, learning the model based on 10 features as compared to learning the model with all features resulted in a model that was more robust for new similar structured data, and which had faster computational times and similar MSEs. Also, reviewing the state-of-the-art statistical TXOT prediction errors showed similar ML TXOT prediction errors. There were some advantages associated to using the current model. First, the ML technique that was used to build the model was fast, intuitive, and interpretable. It could help the airport managers to understand the driving features of the TXOTs per runway–stand. Second, the model has been built based on a large historical dataset (500,000 flight records). More than 40 variables were available for selection as predictors. These variables also enabled one to build new features using domain knowledge of the data. Third, the model could update the predictions in real time. The application developed for CDG Airport allowed easy extraction of real-time data. The forecasting procedure was effective, and the predictions could be generated in a short amount of time. The model was the first to provide TXOT forecasts for each flight to a specific runway. The TXOT forecasts for a flight movement might help ATCOs to make better decisions, to predict whether the flights will experience additional TXOT, and to anticipate in advance on the AOBT by knowing when departure queuing starts. If an ATCO could retrieve this information far in advance, he or she might be able to generate more stable and accurate TXOTs and an EOBT used in Airport Collaborative Decision Making and Airports Operations Centre.

Acknowledgments

This work is supported by the EUROCONTROL Experimental Centre and the Single European Sky ATM Research Programme funded by European Union, EUROCONTROL, and the aviation industry. The authors would like to thank Charles de Gaulle Airport, which provided access to the aircraft runway data, and Jacco Hoekstra for his valuable insights. The authors also wish to acknowledge the support and expertise of the technical teams that made this work possible.

References

[1] Simaiakis, I., and Balakrishnan, H., "Queuing Models of Airport Departure Processes for Emissions Reduction," AIAA Guidance, Navigation, and Control (GNC) Conference, AIAA Paper 2009-5650, Aug. 2009.

[2] "SARDA: An Integrated Concept for Airport Surface Operations Management," NASA, 2013, https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20140013221.pdf [retrieved 9 Oct. 2017].

[3] Shumsky, R. A., "Dynamic Statistical Models for the Prediction of Aircraft Take-Off Times," Ph.D. Thesis, Massachusetts Inst. of Technology, Cambridge, MA, 1995.

[4] Idris, H., Clarke, J., Bhuva, R., and Kang, L., "Queuing Model for Taxi-Out Time Estimation," Air Traffic Control Quarterly, Vol. 10, No. 1, 2002, pp. 1–22.

[5] Legge, J., and Levy, B., "Departure Taxi Time Predictions Using ASDE-X Surveillance Data," The 26th Congress of International Council of the Aeronautical Sciences (ICAS) and 8th AIAA ATIO, AIAA Paper 2008-8958, Sept. 2008.

[6] Srivastava, A., "Improving Departure Taxi Time Predictions Using ASDE-X Surveillance Data," Proceedings of the IEEE/AIAA 30th Digital Avionics Systems Conference (DASC), IEEE/AIAA Paper 2011-6095989, Seattle, WA, Oct. 2011.

[7] Jordan, R., Ishutkina, M., and Reynolds, T., "A Statistical Learning Approach to the Modeling of Aircraft Taxi Time," Proceedings of the IEEE/AIAA 29th Digital Avionics Systems Conference (DASC), IEEE/AIAA Paper 2010-5655532, Salt Lake City, UT, Oct. 2010.

[8] Kistler, M. S., and Gupta, G., "Relationship Between Airport Efficiency and Surface Traffic," 9th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference, AIAA Paper 2009-7078, 2009.

[9] "Mathematical Model," Additional Taxi-Out Time Performance Indicator Document, edited by Capelleras, L., EUROCONTROL–Performance Review Unit, Brussels, 2015, pp. 10–27.

[10] Herrema, F. F., Treve,V., Curran, R., and Visser, H. G., “Evaluation of Feasible Machine Learning Techniques for Predicting the Time to Fly and Aircraft Speed Profile on Final Approach,” International Conference on Research in Air Transportation, Vol. 8, No. 4, TU Delft, Delft, The Netherlands, 2016, pp. 4–8.

[11] Alligier, R., Gianazza, D., and Durand, N., “Machine Learning and Mass Estimation Methods for Ground-Based Aircraft Climb Prediction,” IEEE Transactions on Intelligent Transportation Systems, Vol. 16, No. 6, 2015, pp. 3138–3149.

[12] Chatterji, G. B., and Zheng, Y., “Wheels-Off Time Prediction Using Surface Traffic Metrics,” The 12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference, AIAA Paper 2012-5699, Sept. 2012.

[13] Balakrishna, P., Ganesan, R., Sherry, L., and Levy, B. S., “Estimating Taxi-Out Times with a Reinforcement Learning Algorithm,” Proceedings of the 27th Digital Avionics Systems Conference (DASC), IEEE/AIAA, St. Paul, MN, Oct. 2008, pp. 3.D.3-1–3.D.3-12.

[14] Balakrishna, P., Ganesan, R., and Sherry, L., “Application of Reinforcement Learning Algorithms for Predicting Taxi-Out Times,” The 8th USA/Europe Air Traffic Management (ATM) Research and Development (R&D) Seminars, IEEE/AIAA Paper 2009-369723, Napa, CA, July 2009.

[15] 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.

[16] Balakrishna, P., Ganesan, R., and Sherry, L., “Airport Taxi-Out Prediction Using Approximate Dynamic Programming: An Intelligence-Based Paradigm,” Transportation Research Record, Journal of the Transportation Research Board of the National Academies, Washington, D.C., 2008, pp. 54–61.

[17] Ravizza, S., Atkin, J. A. D., 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. doi:10.1057/jors.2012.123

[18] Ravizza, S., Chen, J., Atkin, J. A. D., Stewart, P., and Burke, E. K., “Aircraft Taxi Time Prediction: Comparisons and Insights,” Applied Soft Computing, Vol. 14, Jan. 2014, pp. 397–406. doi:10.1016/j.asoc.2013.10.004

[19] Lee, H., Malik, W., Zhang, B., Nagarajan, B., and Jung, Y. C., “Taxi Time Prediction at Charlotte Airport Using Fast-Time Simulation and Machine Learning Techniques,” The 15th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference, AIAA Paper 2015-2272, June 2015.

[20] Blockeel, H., Machine Learning and Inductive Inference, Acco Uitgeverij, Leuven, Belgium, 2016, pp. 100–140.

[21] Chapman, P., “CRISP-DM 1.0: Step-by-Step Data Mining Guide,” SPSS, New York, 2000, pp. 50–76, https://www.the-modeling-agency.com/crisp-dm.pdf [retrieved 2018].

[22] "Actual Taxi-Out Time," EUROCONTROL ATM Lexicon, EUROCONTROL, Brussels, 2010, http://www.eurocontrol.int/lexicon/lexicon/en/index.php/Actual_Taxi-Out_Time [retrieved 21 Aug. 2016].

[23] "What is Feature Selection in the Context of Machine Learning," Serial Metrics: Machine Learning Blog, Serial Metrics, San Francisco, CA, 2013, http://serialmetrics.com/blog/what-is-machine-learning-feature-selection/ [retrieved 12 Feb. 2016].

[24] Robnik-Šikonja, M., and Kononenko, I., "Theoretical and Empirical Analysis of ReliefF and RReliefF," Machine Learning, Vol. 53, Nos. 1–2, Oct. 2003, pp. 23–69, http://link.springer.com/article/10.1023%2FA%3A1025667309714 [retrieved 12 Feb. 2016].

[25] Robnik Šikonja, M., and Kononenko, I., "An Adaptation of Relief for Attribute Estimation in Regression," Machine Learning: Proceedings of the Fourteenth International Conference (ICML'97), edited by D. H. Fisher, Morgan Kaufmann, 2003, pp. 296–304.

[26] Karagiannopoulos, M., Anyfantis, D., Kotsiantis, S. B., and Pintelas, P. E., "Feature Selection for Regression Problems," Proceedings of HERCMA'07, Athens Univ. of Economics and Business, Sept. 2007.

5.0 Predicting Abnormal Runway Occupancy Times and observing related Precursors

This Chapter builds on Chapter 4 by using the same methodology but by combining different feasible ML techniques. The reason for selecting this technique is that faster and more accurate predictions are found. The methodology and ML technique is applied to the third ASP enabler, being abnormal AROT. The main purpose of this Chapter is twofold: first, to better characterize and predict AROT as a function of operational parameters from historical data; second, to identify and predict abnormal AROT flights with their associated risk precursors. The identification and prediction is done using a new data-driven method by combining three feasible ML techniques. The key objective of this study is to develop a real-time model that forecasts the AROT for different aircraft types and weather conditions using this new data-driven ML method for Charles de Gaulle and Vienna airports. This model should offer insight into the predictability of key precursors impacting AROT due to abnormality.

This chapter was previously published as:

F.F. Herrema, R. Curran, H.G. Visser, V. Treve and B. Desart. Predicting Abnormal Runway Occupancy Times and Observing Related Precursors, Journal of Aerospace Information Systems, Vol. 15, No. 1 (2018), pp. 10-21.

Abstract

Accidents on the runway triggered the development and implementation of mitigation strategies. Therefore, the airline industry is moving toward proactive risk management, which aims to identify and predict risk precursors and to mitigate risks before accidents occur. For certain predictions machine learning techniques can be used. Although many studies have explored and applied novel machine learning techniques on different radar and A-SMGCS data, the identification and prediction of abnormal runway occupancy times and the observation of related precursors are not well developed. In our previous papers, three existing methods were introduced, lasso, multi-layer perceptron, and neural networks, to predict the taxi-out time on the taxiway and the time to fly and true airspeed profile on final approach. This paper presents a new machine learning method where the existing machine learning techniques are combined for predicting the abnormal runway occupancy times of unique radar data patterns. Additionally, the regression tree method is used in this study to observe the key related precursors extracted from the top 10 features. Compared with existing methods, the new method no longer requires predefined criteria or domain knowledge. Tests were conducted using final approach radar data and A-SMGCS runway data consisting of 78,321 flights at Paris Charles de Gaulle airport and were benchmarked against 500,000 flights at Vienna airport.

5.1 Introduction

Machine learning (ML) can be used to identify patterns and to observe related risk precursors in historic data [1–4]. These patterns and precursors can be transferred into “what-if” statements by analysing relations between the arrival runway occupancy time (AROT) and the precursors [5]. This analysis is needed to predict forthcoming operational risks during real-time landing operations based on the observation of risk precursors. In this context, this study will focus on two different time window predictions. The first one makes predictions from moment t up to 1–5 min, which would feed tactical operational tools to alert air traffic controllers (ATCOs) on abnormal AROT and associated precursors. The second one will make predictions for the next 1–2 h, which may be either used as a strategic operational tool for the ATCO supervisor to decide on changing the sequence algorithm on final approach or can be used as input for the arrival and departure manager (AMAN/DMAN). Both predictions could lead to an improvement in runway safety and throughput.

5.1.1 Related Work

Prior studies have tried to tackle these patterns and observe related risk precursors. We divide the literature into three groups. We start with the first group, which focuses on abnormal events detection methods for aviation systems. The Morning Report software package was one of the earliest efforts made to detect abnormal events from routine flight data recorder (FDR) data [6]. The software models the time series data of selected flight parameters using a quadratic equation. Each flight is mapped into a point that is described by the coefficients of the quadratic equations in the feature space. Thereafter, for each flight an “atypical score” is measured using the distance between the point and the mean of the distribution in the feature space. Later studies apply data-mining techniques to detect abnormal events in data of aerospace systems [7–12]. One of these studies applies supervised learning software called Inductive Monitoring System (IMS) [10]. The IMS software method summarizes the data distributions of typical system behaviors from a historical training dataset, which is then compared with real-time operational data to detect abnormal behaviors. However, the limitation of the IMS is that it always needs a training data set for labeling the norms. Other studies used unsupervised learning techniques. These studies focus on discrete flight parameters for monitoring pilot operations, such as cockpit switch flips [7,8]. The techniques observe abnormal events in the switch operations based on the longest common subsequence measures. The study presented in [9] developed a statistical framework to incorporate both continuous and discrete flight parameters in FDR data. Das et al. developed Multiple Kernel Abnormal Detection (MKAD), which applies a one-class support vector machine for abnormal detection [12]. MKAD assumes one type of data pattern for normal operations, which is not always valid in real operations, because standards vary according to flight conditions. Most recently, Matthews et al. summarized the knowledge discovery pipeline for aviation data using the previously discussed algorithms [13]. The second group focuses on the AROT extraction by applying statistical methods. For example, Kumar et al. [14] conducted a study that applied statistical methods for the extraction and analysis of surface track data for AROT. The study [15] highlights the factors using statistical methods that influence the AROT such as aircraft weight, velocity, air carrier, and meteorological conditions. Based on the above research studies, Airbus developed a system that shows the cockpit operational need by calculating the AROT and braking distance required when the aircraft is at touchdown point. In [16] the authors show a method for assisting the AROT and braking of an aircraft on a runway. The method comprises six steps, which are implemented automatically into the Airbus brake-to-vacate system.

The third group of literature consists of abnormal event detection methods developed and applied outside the aviation domain. In general, those abnormal event detection techniques can solve problems for a domain-specific formulation. Some techniques are developed for intrusion detection in computer systems [17,18], fault detection in mechanical units and structures [19,20], and fraud detection related to phones, credit cards, insurance claims [21,22], and so on. Additionally, two groups of techniques are developed for time-series data depending on how dissimilarities are measured: data based and model based [23,24]. The former measures the dissimilarity based on data observations. The dissimilarity is measured by a variety of distance functions, such as Euclidean distance [25,26], dynamic time warping distance [27], probability-based distance [28], correlation-based distance [29]. Based on the literature review, the AROT prediction methodology is lacking in the following aspects: first, extracting top AROT features using statistical methods and feature selection techniques; second, applying top features before training the model and combining feasible state of the art ML algorithms to train the model; and third, developing algorithm that is capable of real-time computation and finally providing decision tree distributions associated with the individual abnormal AROT predictions to synthesize the what-if statements. Therefore, it is envisaged to progressively move from a statistical approach to a new ML approach for coping with the variability of AROT behaviors.

5.1.2 Aim

The main purpose of this study is twofold: first, to better characterize and predict AROT as a function of operational parameters from historical data; second, to identify and predict abnormal AROT flights with their associated risk precursors. The identification and prediction is done using a new data-driven method by combining three feasible ML techniques. The key objective of this study is to develop a real-time model that forecasts the AROT for different aircraft types and weather conditions using this new data-driven method for Charles de Gaulle (CDG) and Vienna (VIE) airports. This model should offer insight into the predictability of key precursors impacting AROT.

5.1.3 Structure

The structure of this paper is as follows: First, the data and prediction variables are described in Section 5.2. Second, the methodology is outlined in Section 5.3, introducing the AROT behavior and data preparation. Furthermore, the feasible ML techniques are combined to a new ML model, which is applied on a regression tree. Thereafter abnormal AROT results are predicted, and related precursors are observed. In Section 5.4 the real-time model is outlined, and finally conclusions are drawn in Section 5.5.

5.2 Data and Prediction variables

The AROT is a key driver of airport runway throughput, especially when low airborne separation minima are applied. Several factors, such as aircraft type, weather conditions, traffic demand, and ATCO workload, influence the AROT [14,15]. All factors are included in our model with the exception of the ATCO workload, which will be included in future studies. To predict AROT profiles and extract risk precursors, final approach and runway radar data are used.

Final Approach and Runway Radar Data

Radar, A-SMGCS, and weather data are extracted from runway schedulers and have been provided by CDG and VIE airports. The data set covers, respectively, 5 years from 2011 to 2015 and 3 years from 2013 to 2015 of final approach radar data and A-SMGCS runway data (Table 5.1). The weather data set covers 5 years of data from 2011 to 2015. In total, the data comprise about 78,321 and 500,000 arrival flights. The data sets are stored in CSV formats and are thereafter saved in separate MatLab and Python files.

Table 5.1: Prediction and target variables.

AROT variables	Description
1. Anne	Year
2. Caractredevol	Commercial or private flight
3. CodeIATA	IATA code company
4. CodeAeroportOACI	Airport origin ICAO code
5. CodeAeroportIATA	Airport origin IATA code
6. Compagnie	Airline
7. Crosswind	Crosswind vector
8. DateReal	Actual date
9. Deep landing	The runway length available beyond the touchdown point
10. IdentifiantvolATC	ATCO call sign
11. Long flare	Estimate the start of the flare until touchdown
12. Mois	Month
13. NumFlight	Flight Number
14. Postedestationnement	Gate arrival
15. QFU	Runway orientation and runway exit
16. Semaine	Week
17. Tailwind	Tail wind vector
18. Temp	Temperature
19. TimeReal	Actual time of the day
20. Typeavion	Aircraft type and ICAO category
21. Visibility	METAR visibility conditions
22. Arrival runway throughput	The amount of landings that is performed on the runway during the last 30 min
23. ACSpeedPoint	Speed of the aircraft at 2NM, and 1NM out, threshold and the runway exit point (RWEP)
24. ALDT	Actual Landing Time
25. AROT	Arrival Runway Occupancy Time

5.3 Methodology

For this study, we propose a methodology comprising five steps. The method is based on previous work [3,4,30] and the Statistical Package for the Social Science (SPSS) ML method [31]. This methodology describes the steps to come up with a usable predictability model that identifies and to predict abnormal AROT flights with their related risk precursors. Each step is detailed below.

5.3.1 Identification and Understanding of the AROT

The AROT is extracted by calculating the time between the aircraft crossing the threshold and its tail vacating the runway [5], using the variables ACSpeedPoint and ALDT, as defined in Table 5.1 a matrix Y is then formed, where each row represents a flight, columns 1 until 22 a prediction variable, and column 25 the AROT target variable (see Table 5.1). Based on the state-of-the-art AROT analyses and the available data in Table 5.1, this subparagraph will show the correlation between the most important prediction variables from the literature and the AROT target variable, using a statistical approach. In Section 5.3.2 ML feature selection techniques will be applied to verify if the top features are similar. We propose to first extract the AROT per aircraft type for 78,321 CDG flights for runway 09L, 27R, 08R, and 26L and for 500,000 VIE flights for runway 11L and 29R. This is done to cover seasonal variations and to have a minimum of 15 AROT measurements per aircraft type and runway. Figure 5.1 shows the AROT values for 49 aircraft types at runway 08R at CDG airport. The AROT results for the remaining CDG runways and 49 different aircraft types can be found in [30]. We observe from Figure 5.1 that the AROT differs per aircraft type. As a next step we plot in Figure 5.2 the AROT as function of the aircraft ICAO categories: “Heavy” (H), “Medium” (M), and “Small” (S), per CDG runway and time of the day. The error bars represent the standard deviation σ and -σ.

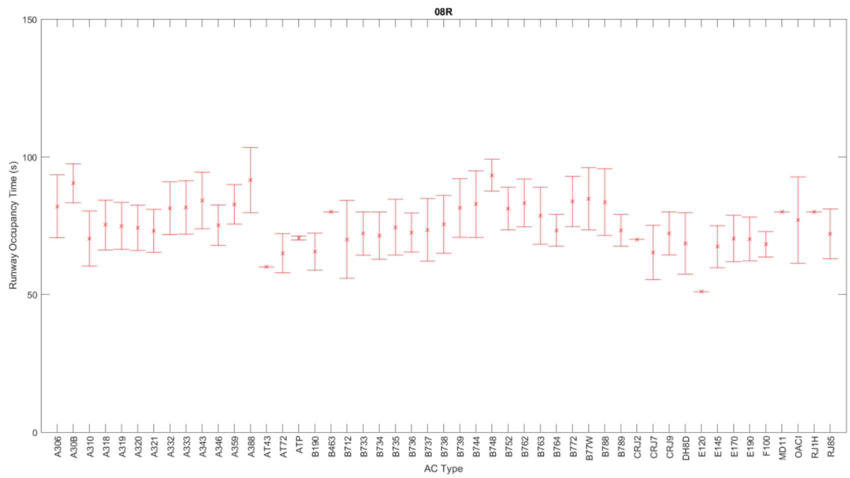
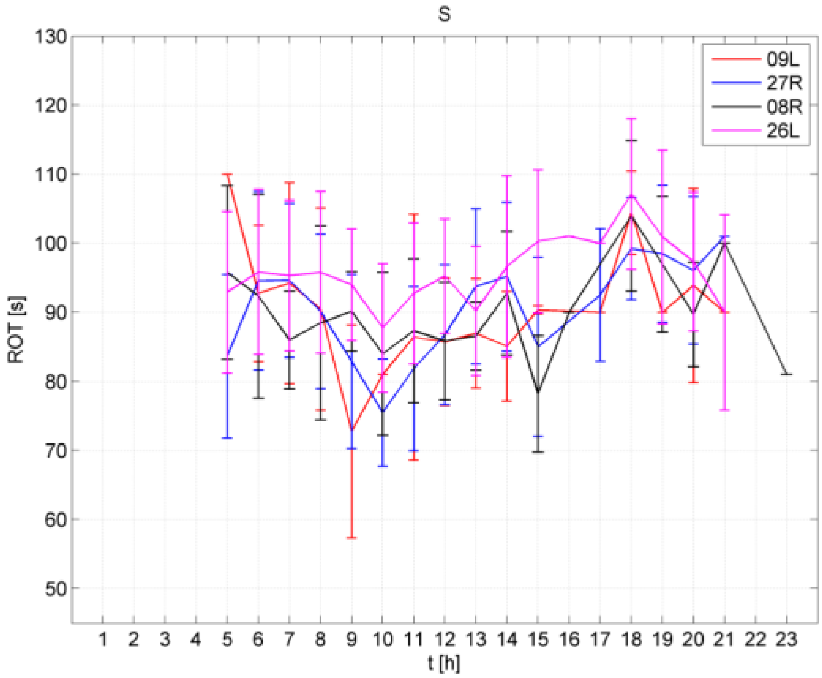
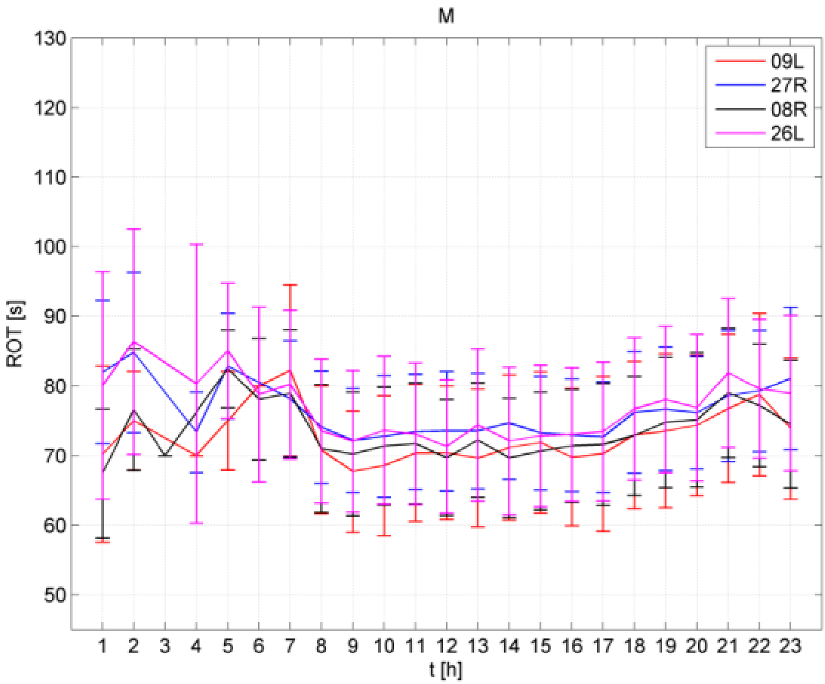


Figure 5.1: Example of AROT per aircraft type for runway 08R.



S
09L
27R
08R
26L
ROT [s]
t [h]



M
09L
27R
08R
26L
ROT [s]
t [h]

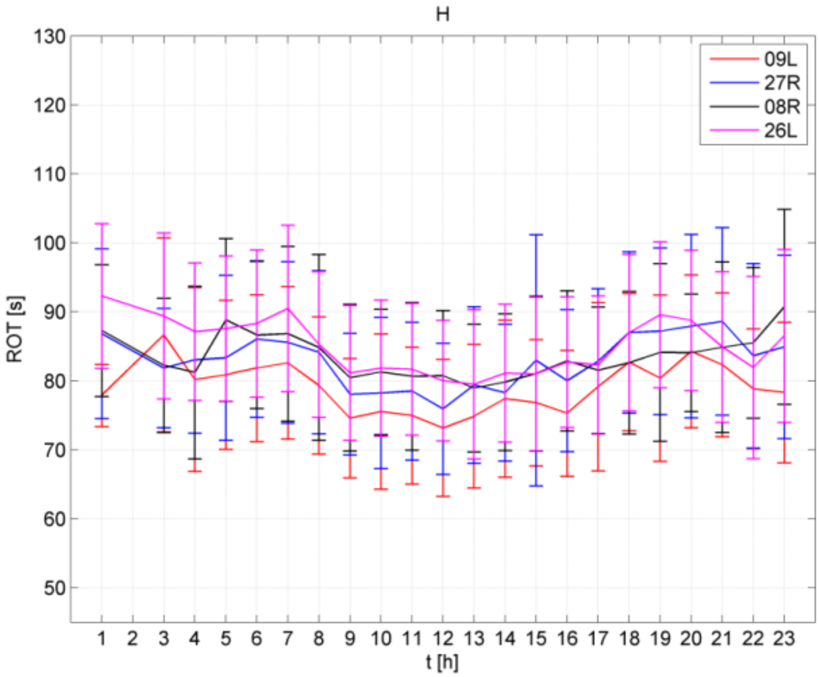


Figure 5.2: AROT versus time of the day and different ICAO aircraft categories (H, M, S) for runway 09L, 27R, 08R, and 26L.

Figure 5.2 shows that there is an effect of the hour of the day on the extracted AROT, especially during higher arrival demand (during peak hours), which is ranging at CDG airport from 07:00 till 08:00 a.m., from 09:30 till 10:30 a.m., from 12:00 till 13:00 a.m., and from 15:00 till 16:00 a.m.

Figure 5.3 shows an example for the "Medium" aircraft types. According to [15] the AROT depends mainly on the aircraft type/category and the brake policy by the airlines. Therefore, as a next step we plot in Figure 5.4 the AROT versus different CDG runway exits for the ICAO wake vortex category. All results can be found in report [30]. Because of the confidentiality agreement, results for the AROT per airline cannot be shown. We observe from Figure 5.4 that "Super Heavy" (S) and "Heavy" (H) aircraft categories have a significantly higher AROT in comparison to "Medium" (M) categories.

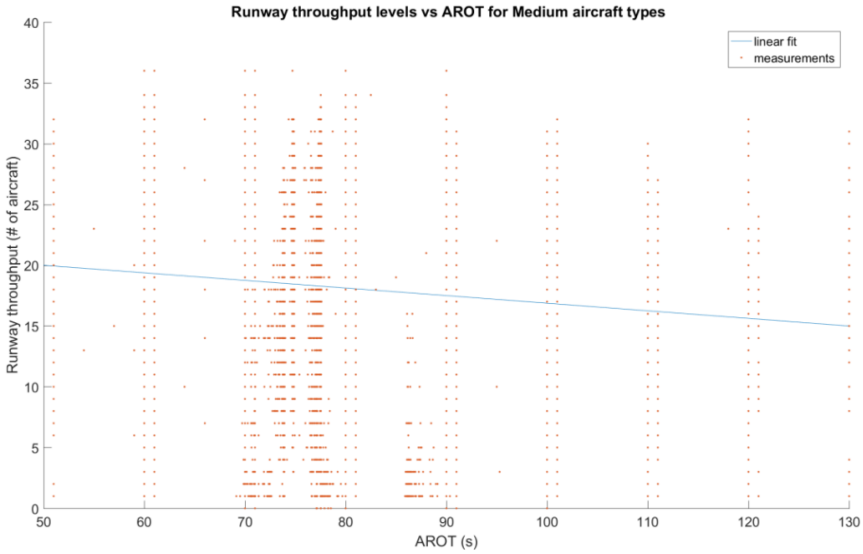
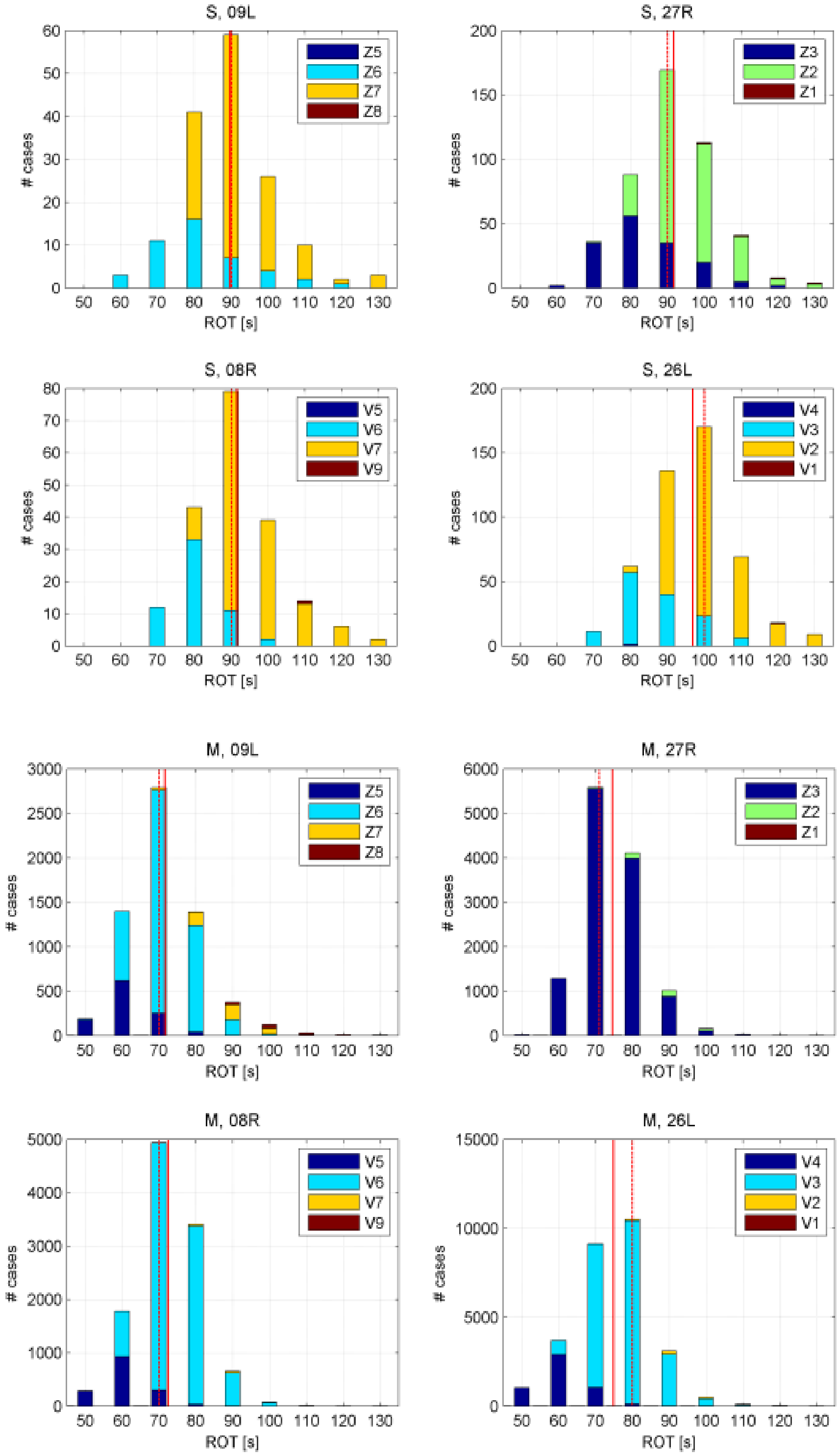


Figure 5.3: Runway throughput levels versus AROT for different Medium aircraft types on runway 26R.

Figure 5.3 shows indeed that the peak hours lead to lower AROT, and lower AROT can lead to higher arrival throughput. Plotting the arrival throughput versus the AROT confirms that a higher throughput leads to a decrease in AROT for all ICAO categories.


S, 09L
Z5
Z6
Z7
Z8
cases
ROT [s]
S, 27R
Z3
Z2
Z1
cases
ROT [s]
S, 08R
V5
V6
V7
V9
cases
ROT [s]
S, 26L
V4
V3
V2
V1
cases
ROT [s]
M, 09L
Z5
Z6
Z7
Z8
cases
ROT [s]
M, 27R
Z3
Z2
Z1
cases
ROT [s]
M, 08R
V5
V6
V7
V9
cases
ROT [s]
M, 26L
V4
V3
V2
V1
cases
ROT [s]

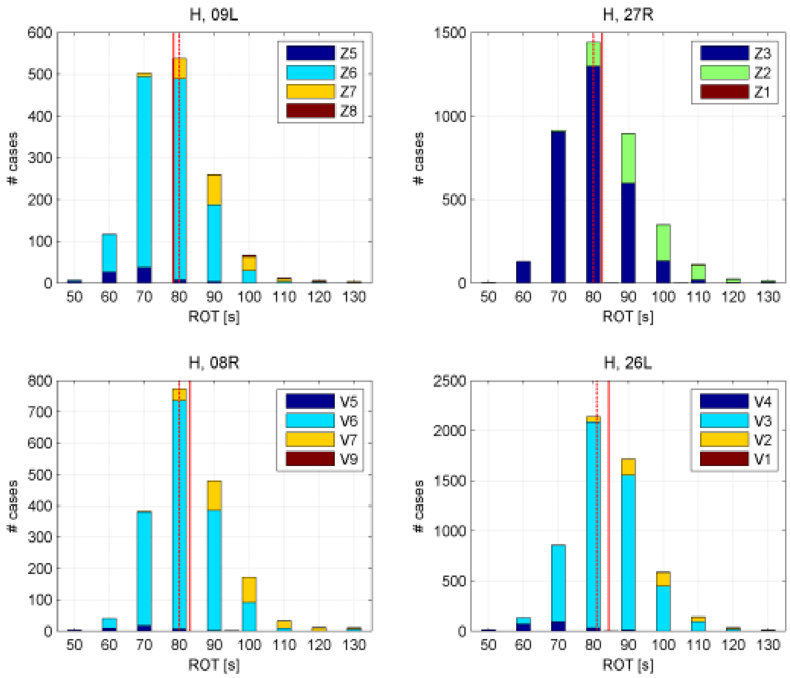


Figure 5.4: Number of flights versus the AROT for different runway exits for Super Heavy (S), Heavy (H), and Medium (M) categories.

5.3.2 Data Preparation

The data preparation phase covers all activities required to set up the final dataset from the initial raw aircraft operational runway and final approach radar data. Before the ML model will be trained with the prediction variables highlighted in Table 5.1, first the most important (group) features will be selected using RreliefF and Sequentialfs (feature selection) techniques. In previous papers we have introduced these techniques [3,4]. The RreliefF is only suitable for regression problems where the predicted value is continuous; therefore (nearest) hits and misses cannot be used. To solve this difficulty, instead of requiring the exact knowledge of whether two instances belong to the same class or not, a kind of probability that the predicted values of two instances are different is introduced. This probability can be modeled with the relative distance between the predicted (class) values of two instances. The function Sequentialfs provides a simple way (the default option) to decide how many features are needed. It stops when the first local minimum of the cross-validation MCE (misclassification error, i.e., the number of misclassified observations divided by the number of observations) is found. The objective of feature selection is threefold: improving the prediction performance of the predictors, providing faster and more effective predictors, and providing a better understanding of the underlying process that generated the data [32]. RreliefF and Sequentialfs have commonly been viewed as feature selection methods that are applied in a prepossessing step before the model is learned [33]. The standard RreliefF regression modelling technique has been extensively discussed in many papers [34]. In this study, the technique has been applied on 78,321 final approach flights for CDG as shown in Figure 5.5 and benchmarked against 500,000 VIE flights.

We observed for both airports that the QFU (runway orientation and runway exit), aircraft type, arrival runway throughput, visibility, wind vectors, and temperature are the most important features for predicting the AROT. The same results are obtained for the Sequentialfs technique. Those 10 features are considered for the remainder of this paper. Thereafter we construct the datasets and find the stability of three different data parts. Based on different data sources and the variables listed in Table 5.1, we standardize the top 10 feature matrix (X) for all the flights (Y). We have split the matrices X and Y into two subsets, Xtrain;Ytrain, and then used them to train the model and Xtest;Ytest have been used to evaluate the model accuracy. Finally we specify the default ratios (splitting the data) for training, testing, and validating into, respectively, 70%, 15%, and 15%.

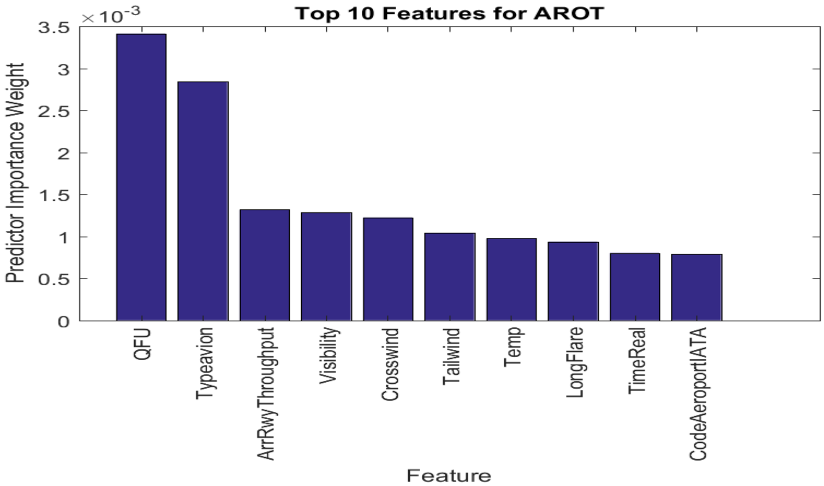


Figure 5.5: Top 10 features for the AROT using the RreliefF technique for CDG.

It can be concluded that the top three features correspond to [15], which uses statistical analyses to find the most correlated prediction variables. However, the time of the day is an exception, which is ranked 9 using ML feature selection and ranked 2 using statistical analyses. In future studies we will assess the influence of weather variables on the AROT using a statistical approach and ML approach.

5.3.3 Combining Feasible ML Techniques

For choosing suitable ML techniques for the AROT prediction we have to take into account the size, quality, and nature of the data. Even the most experienced data scientists cannot tell which algorithm will perform best before trying them [35]. In previous work [3,4], ML techniques were assessed on their capabilities to produce fast and accurate predictions and to test a number of what if statements. Next to these what-if statements this study will focus on the robustness of the model. A possibility to achieve this is by using the outcome of different feasible ML techniques [35–37]. Thereafter the regression tree was built to synthesize the what-if statements for our abnormal AROT predictions and to give a clear overview for the ATCO for what is likely to happen under certain situations. This study will use the outcome of four feasible ML techniques: lasso, multi-layer perception (MLP), neural networks (NN), and Regression Forest (RF). By doing so we take into account the characteristics of final approach and runway radar data. The method is based on expert studies [35–37] and will be explained below in 3 steps.

Novel Combined ML Method

1) First, we learn the AROT for 78,321 flights with 10 different features. Four models will be learned using the Lasso, MLP, NN, and RF techniques.

2) Second, we merge the AROT results of all 4 models to 1 final matrix for 78,321 flights.

3) Finally, we apply the Classification and Regression Tree (CART) technique to the final matrix obtained in the previous step.

5.3.4 Assessing Combined ML Method

Before we analyse the forecast performance, computational time, and minimum amount of data needed for the novel combined ML technique, we first check the stability of three different data parts known as cross validation. To check the stability of different data parts, the data will be randomly divided into training, validation, and testing subsets. It has been assumed that the default ratios in this study for training, testing, and validation are 0.70, 0.15, and 0.15, respectively. The model is adjusted accordingly during training. The validation is used to measure network generalization and to halt training when generalization stops improving. To prove that a randomly selected data set is stable, epoch and validation checks are performed. Epoch indicates the amount of a single pass through the entire training set for which all of the training vectors are used once to update the weights, followed by testing of the verification set. Thereafter we check convergence on the validation, and at the end of the learning process the model is evaluated on the test set. The test has no effect on the training and therefore provides an independent measure of network performance during and after training. Figure 5.6 shows a trained model by selecting 78,321 CDG final approach flights. We learned the model with the 10 most important prediction variables highlighted in Figure 5.5. It has also been tested that the same mean squared error (MSE) results are obtained using all features. However, by excluding 15 variables (including ACSpeedPoint, ALDT, and AROT) the model is trained two times faster. The MSE is calculated by comparing the predicted outputs with the real target values. Our model gives a worst-case MSE of 18.8 s.

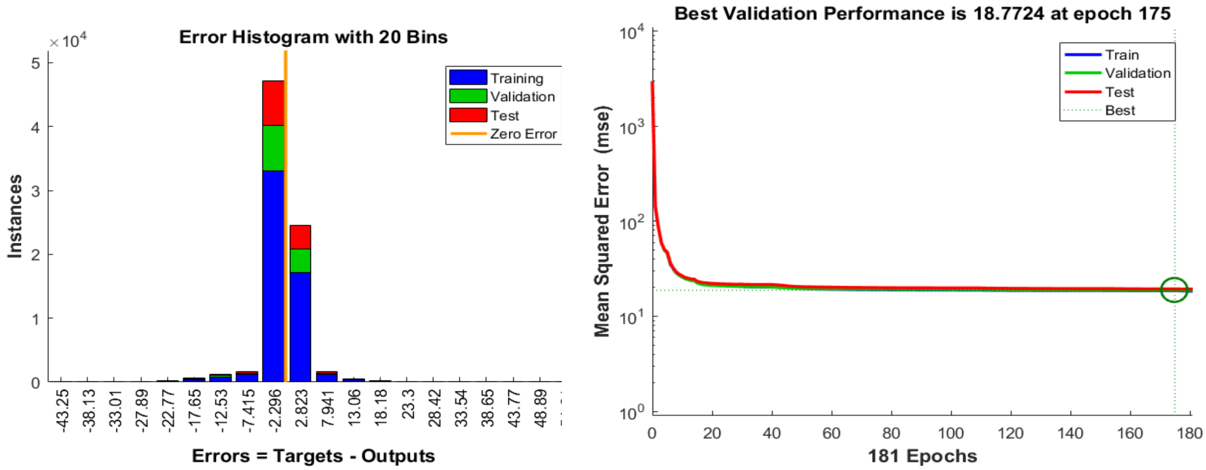


Figure 5.6: MSE of AROT using the top 10 features.

Abnormal AROT Flights and Observation of Outliers

We are interested in abnormal flights that stay too long on the runway. Therefore, in this study we consider an abnormal AROT if it is 2σ standard deviation from the normal distribution mean. The mean and 2σ values are calculated from the 78,321 CDG flights for which we assumed normal distribution. Only 8100 abnormal AROT flights are learned with our combined ML technique. As a next step, the inconsistency is measured between the outputs and the targets. One way to show this inconsistency is by plotting the regression for the training, validation, and test sets and for the complete set (all). Figure 5.7 shows an example where the regression R values measure the inconsistency between the predicted outputs and the targets. An R value of 1 means a close relationship, and 0 a random relationship. By analysing these R values we observe outliers. With outliers we mean occurrences where a data point is not consistent with the other data points. In this study we assume an outlier when it has an R value between 0 and 0.25. Analysing these graphs reveals that there are indeed outliers. It will be obvious that by excluding them in the target set, a better R value will be obtained for the predicted model. Doing this for the above example results in an overall R value of 0.54 instead of 0.31, as presented in Figure 5.7

The next and final step of our methodology (step E) will only take those abnormal AROT flights into account with an R value of $0.25 < R < 1$ for runway 08R at CDG.

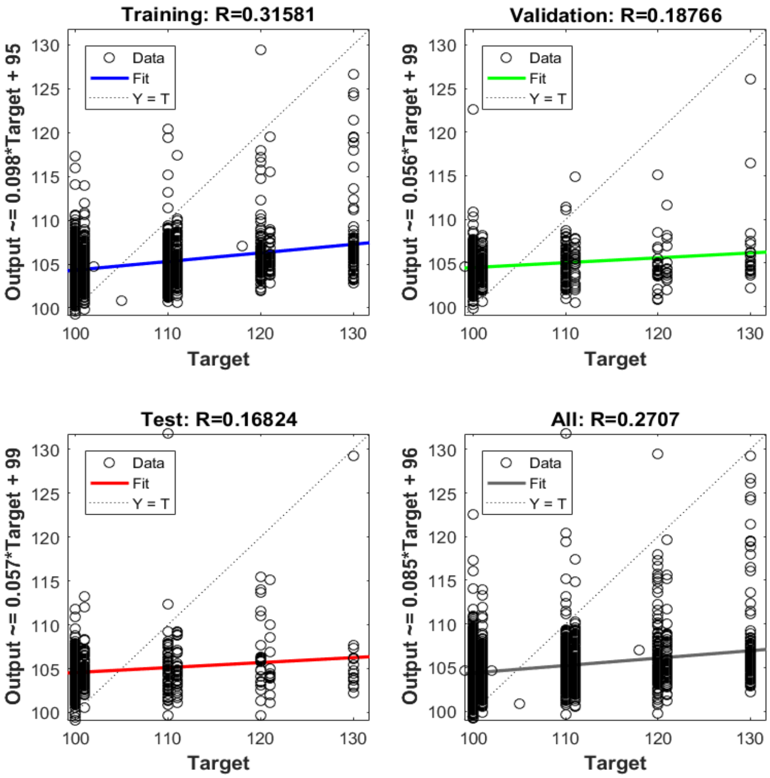


Figure 5.7: Outliers example of abnormal AROT flights for training the model with 10 features.

5.3.5 Observe Risk Precursors with Regression Tree

The purpose of building a regression tree is to extract a set of if-then-else (what-if statements) split conditions in order to extract the main risk precursors that most influence abnormal AROT flights. The observed flights from Figure 5.7 ($0.25 < R < 1$) for runway 08R are analysed and build into a single regression tree, which should give a good understanding of which features (top 10 from Figure 5.5) influence these abnormal AROT flights. By building this tree we start at the root node, and ask a sequence of questions about the predictors. In each iteration, the tree chooses the variable and the split point to achieve the minimum MSE between the predictions and the abnormal AROT targets. This process will continue until a stopping rule is applied. Each of the terminal leaves represents one of the partitions of the input space. To provide a model that can generate accurate predictions and is not overcomplicated, we need to find the optimal tuning parameters for the tree. In this study we use two parameters. The first parameter is the minimum leaf size (lmin), for which we need enough data points in each terminal node to create a distribution. The parameter minimum leaf size can be used to stop the splitting process when the number of instances in a leaf is too small. In addition, if the tree contains too many variables, it is hard to interpret. The second tuning parameter for the tree is the maximum tree depth (dmax). Avery 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 variable or top 10 feature variables. The maximum tree depth can restrict the number of layers of a tree. In Figure 5.8, cross-validation is used to select the minimum leaf size, lmin and, in Figure 5.9, the MSE to select the maximum tree depth, dmax. So in our case the tree is fit for a range of values of the two parameters to three quarters data. Thereafter, the MSE of the predictions is computed on the remaining one quarter. This is done for each quarter of the data, and the four MSE values are averaged. The set of parameters that give the lowest MSE will be selected. As shown in Figure 5.9, we first train the trees with all 22 variables and different settings of dmax and lmin. We observed that the MSE drops as the tree depth increases from 1 to 7, regardless of the leaf size. After tree depth reaches the value 6, the MSE does not change significantly. On the other hand, a tree with a minimum leaf size of 16 performs slightly better than the trees with minimum leaf sizes of 20 and 24. We have also tried to set the minimum leaf size to be less than 16, but the model does not improve much. Moreover, if we further reduce the leaf size, we may not have enough instances in the leaves to fit a distribution. Thereafter we also train a model with the top 10 features shown in Figure 5.5, where we fit a tree to the entire data set with maximum tree depth and minimum leaf size set to 6 and 16, respectively. We then sort the predictors based on their feature importance and select the first 10 as the final predictors. Thereafter, we retrain the tree with these 10 variables. We change the values of dmax and lmin and then repeat the cross-validation process described above. The tree with lmin 16 still performs slightly better than the others, and the MSE does not change significantly after tree depth reaches the value 6. Thus, our final model has 10 predictors and is fitted with maximum tree depth and minimum leaf size set of 6 and 16, respectively. By learning the tree a mean and distribution is extracted per decision node. This is needed to observe risk precursors and understand what is likely to happen for abnormal AROT flights. Our model divides all the abnormal AROT flights into 17 segments. In other words, the regression tree shown in Figure 5.10 has 17 terminal nodes for which the outcomes are rounded off to 100, 105, 110, 115, or 120 s. We can interpret the most important predictors as the major factors that play key roles in influencing abnormal AROT.

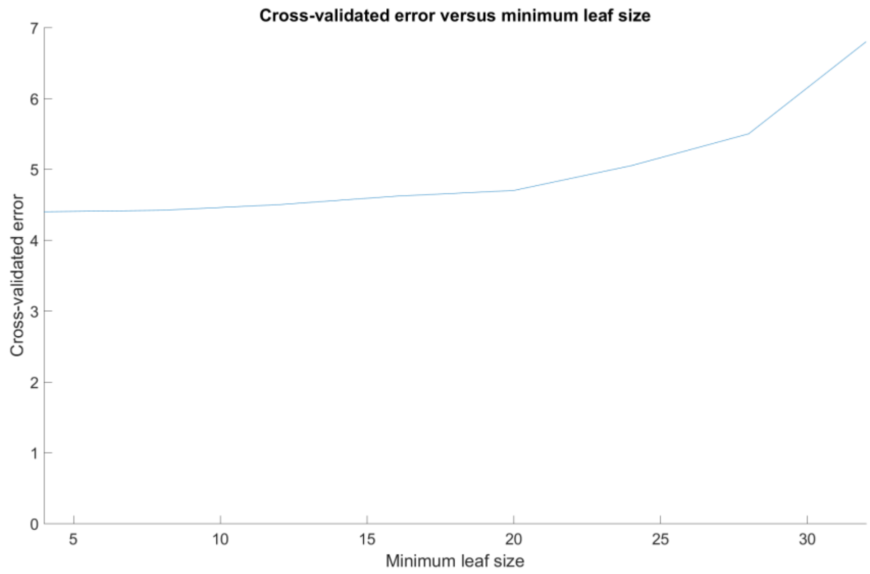


Figure 5.8: Cross-validated error versus minimum leaf size.

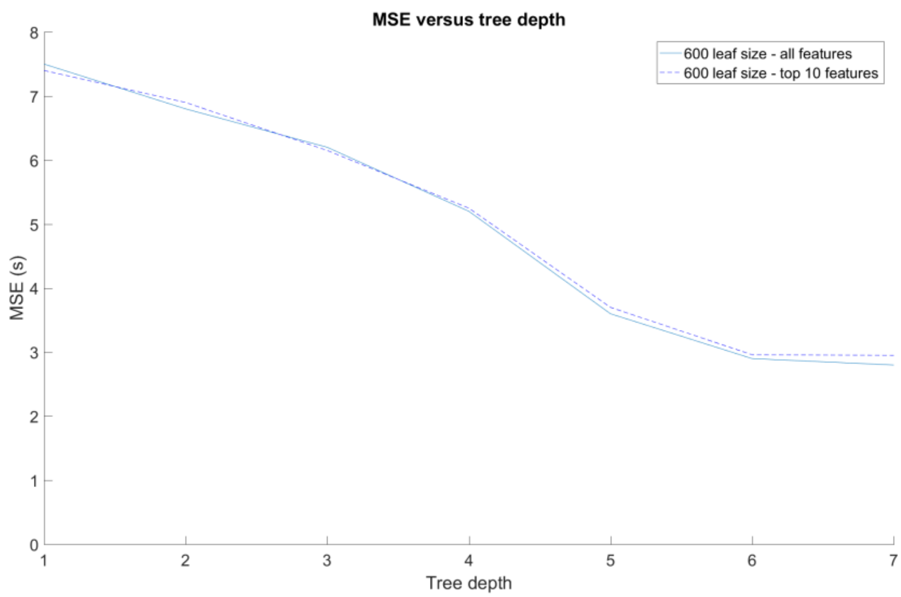


Figure 5.9: MSE versus tree depth for different leaf size and features.

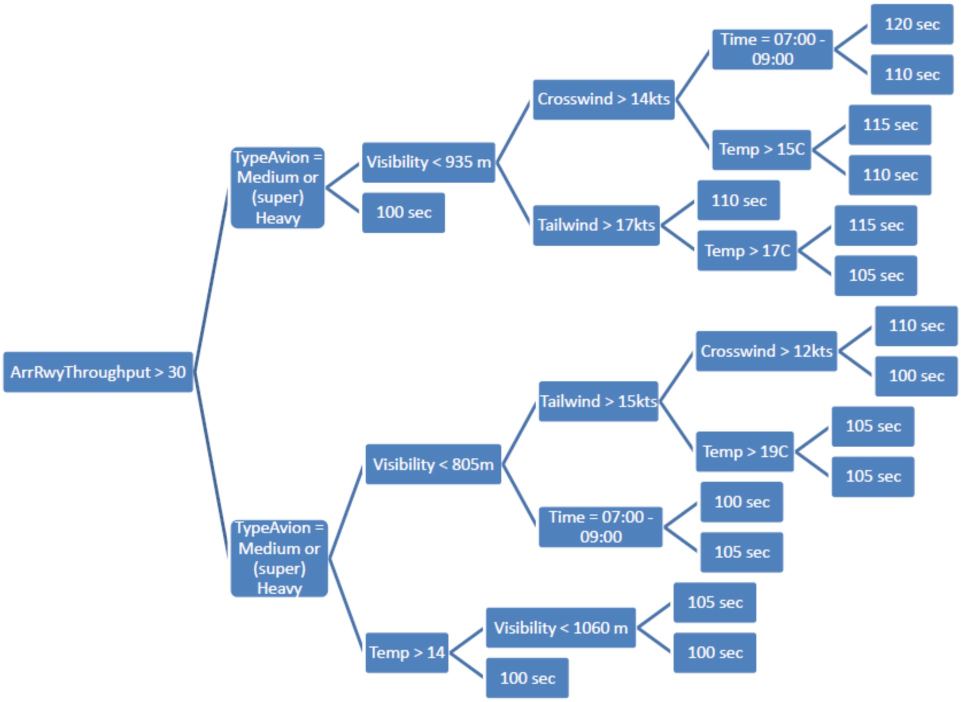


Figure 5.10: Regression tree only for abnormal AROT at runway 08R. The tree shows "what-if" statements. If the statement is right, we go to the upper node; if the statement is wrong, we go to the lower node.

After the tree is learned with a tree depth of 6, we observed for the 17 precursor categories their related precursors, which are listed in Table 5.2.

Table 5.2: Seventeen different abnormal AROT categories with their related precursors.

Precursor category	Amount of flights observed per precursor category	Median of the abnormal AROT and the Root MSE
1.ArrRwyThroughput > 30 TypeAvion = Medium or (super) Heavy Visibility < 935m Crosswind > 14kts Time = 07:00 -09:00	33	120 sec 3.5 sec
2.ArrRwyThroughput > 30 TypeAvion = Medium or (super) Heavy Visibility < 935m Crosswind > 14kts Time ≠ 07:00 -09:00	21	110 sec 2.8 sec

3.ArrRwyThroughput > 30 TypeAvion = Medium or (super) Heavy Visibility < 935m Crosswind ≤ 14kts Temp > 15C	18	115 sec 2.9 sec
4.ArrRwyThroughput > 30 TypeAvion = Medium or (super) Heavy Visibility<935m Crosswind ≤ 14kts Temp ≤ 15C	40	110 sec 2.1 sec
5.ArrRwyThroughput > 30 TypeAvion = Medium or (super) Heavy Visibility ≥ 935m Tailwind > 17kts	18	110 sec 3.4 sec
6.ArrRwyThroughput > 30 TypeAvion = Medium or (super) Heavy Visibility ≥ 935m Tailwind ≤ 17kts Temp > 17C	21	115 sec 3.2 sec
7.ArrRwyThroughput > 30 TypeAvion = Medium or (super) Heavy Visibility < 935m Tailwind ≤ 17kts Temp ≤ 17C	18	105 sec 3.9 sec
8.ArrRwyThroughput > 30 TypeAvion ≠ Medium or (super) Heavy	17	100 sec 4.0 sec
9.ArrRwyThroughput ≤ 30 TypeAvion = Medium or (super) Heavy Visibility < 805m Tailwind > 15kts Crosswind > 12kts	23	110 sec 2.2 sec
10.ArrRwyThroughput≤30 TypeAvion = Medium or (super) Heavy Visibility < 805m Tailwind > 15kts Crosswind ≤ 12kts	30	100 sec 2.5 sec

11.ArrRwyThroughput≤30 TypeAvion = Medium or (super) Heavy Visibility < 805m Tailwind ≤ 15kts Temp > 19C	26	105 sec 3.1 sec
12.ArrRwyThroughput≤30 TypeAvion = Medium or (super) Heavy Visibility < 805m Tailwind ≤ 15kts Temp ≤ 19C	19	105 sec 3.2 sec
13.ArrRwyThroughput≤30 TypeAvion = Medium or (super) Heavy Visibility ≥ 805m Time = 07:00 -09:00	20	100 sec 3.5 sec
14.ArrRwyThroughput≤30 TypeAvion = Medium or (super) Heavy Visibility ≥ 805m Time ≠ 07:00 -09:00	26	105 sec 4.2 sec
15.ArrRwyThroughput≤30 TypeAvion ≠ Medium or (super) Heavy Temp > 14C Visibility < 1060m	19	105 sec 2.4 sec
16.ArrRwyThroughput≤30 TypeAvion ≠ Medium or (super) Heavy Temp > 14C Visibility ≥ 1060m	24	100 sec 3.4 sec
17.ArrRwyThroughput≤30 TypeAvion ≠ Medium or (super) Heavy Temp ≤ 14C	20	100 sec 3.2 sec

Given the regression tree in Figure 5.10, we fit a parametric distribution to each terminal leaf. The probability distributions we considered include the Gumbel, Gamma, and F distributions. The following equation shows the Gumbel distribution because this one appears to fit best.

$$f(x) = \frac{1}{\beta} e^{-\left(\frac{x-\mu}{\beta} + e^{-\frac{x-\mu}{\beta}}\right)}$$

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

The shapes of the 17 terminal leaves distributions are quite different from each other. In general, the Gumbel distributions with lower medians are less spread out. This indicates that in these segments, the uncertainties of the AROT flights are low. If there are a lot of AROT flights in these segments at the airport, the managers should have more confidence in making adjustments to their plans.

5.4 Real-Time model

Based on the data availability we develop in this section a prototype model using the same combined ML method and CART technique to forecast both abnormal and normal AROT at CDG airport. In the previous section we showed the results only for abnormal AROT that were extracted by taking the mean and 2σ from all flights. This section will build a tree based on all 78,321 flights and associated radar data collected over 5 years. The tree will recognize by receiving real-time data if a certain flight will fall into an abnormal or normal leave node with a given mean and MSE. To generate real-time predictions from the model, we develop an application using MatLab. The output from running the application included the mean and quantiles of flights. The aim is to generate forecasts for each AROT flight and number of landing aircraft to a given runway per time window. Suppose that we are at time h, and try to make predictions for the next x min. Given real-time flight information before ALDT, our regression tree model will determine which segment the flight belongs to. For example, if a small aircraft type plans to land at runway 08R with an ArrRwyThroughput >25, then the median of the AROT is 76 s and the distribution of its connection time can be described by a Gumbel distribution with $\mu = 80$ and $\beta = 5.0$. Next, we produce the distribution of the number of AROTs during a certain time interval [h1, h2], where $h < h1 < h2 < h + x$. This distribution is obtained by aggregating all the distributions of the flights that landed in the last 2 h or will land in the next x min. For this prototype we assume to update the model and tree on 2 h of historical daily data from moment h1; by doing so the what-if statements from the tree are updated. For updating the top 10 precursors and for robustness and accuracy reasons, it is suggested to update the whole model and tree when a new prediction is made on a new day. The procedure of generating this distribution is summarized in two steps:

1) Suppose that there are n flights that landed in the last 2 h or will land in the next x min. We sample one AROT from each of the n flights' distributions, and calculate the time when the tail is vacating the runway. We then count how many flights landed between the time interval [h1, h2], and record this number as y1.

2) Repeat step I, m times, and construct an empirical distribution using y1, y2,…ym. Then the qth quantile of the number of flights landing between the time interval h1; h2 can be approximated by the qth quantile of y1, y2,…ym. In the live trial we produced the distributions of the AROT of the flights that have landed in the last 2 h or will land in the next x min.

Live Trial: Assumed Real-Time Data

We assume real-time data for a selected data set for which we exclude the actual data variables ALDT ACSpeedPoint. To conveniently generate predictions in real time, we develop a MatLab compiler that can work in most operation systems (Windows, Linux, Mac, etc.). Figure 5.11 shows the interface of the application. This application allows users to set the following parameters:

1) Forecasting window (x)

2) Runway at CDG (RWY)

3) Number of simulations (m)

4) Update frequency (min)

5) Forecast resolution (r)

6) Starting time of the first forecasting window (YY-MM-DD hh-mm-ss)

7) Machine learning (ML) technique

8) Ending time of the last forecasting window (YY-MM-DD hh-mm-ss)

The default settings of the first three parameters are 120 min, 1500 simulations, and 5 min, respectively. We update the predictions every 10 min and the default resolutions are 1, 5, 15, and 60 min. The starting time defaults to the current time if the user does not specify one. The ending time will be 24 h after the starting time. As shown in Figure 5.12, the predictions for this case study are generated on a rolling basis to runway 08R. Suppose that the trial started at 8:00 a.m. We first collected data of the flights that landed at CDG after 6:00 a.m. or will land before 10:00 a.m., and then generate forecasts for the next 2 h (8:00–10.00 a.m.). Thirty minutes later (8:30 a.m.), the second trial started. Similarly, we considered only the flights that landed at CDG after 6:30 a.m. or will arrive before 10:30 a.m., and generate forecasts for the time interval between 8:30 to 10:30 a.m. A 2-h time window prediction is chosen to feed the

ATCO need. The dots in Figure 5.12 show the difference between the predicted AROT and their real values (error). The difference is measured in seconds and shown for 12 flights, for which each flight falls into one of the ICAO categories “Small” (S), “Medium” (M), or “Heavy” (H). We did not show all the flights in this example but only three per 30 min to illustrate the differences per ICAO category.

```
Command Window
>> CDGRwyOccupancyTime
ML technique (Combine Lasso, Multi-Layer perception and Neural Networks (yes or no)):yes
Forecast window (min):120
Number of simulations:1500
Runway:08R
Forecast resolutions (separated with commas):1,5,15,60
Update frequency (min):5
Starting time to predict (YYYY-MM-DD HH:MM:SS):2015-07-20 08-00-00
fx Ending time to predict (YYYY-MM-DD HH:MM:SS):2015-07-20 14-00-00
```

Figure 5.11: Interface of application.

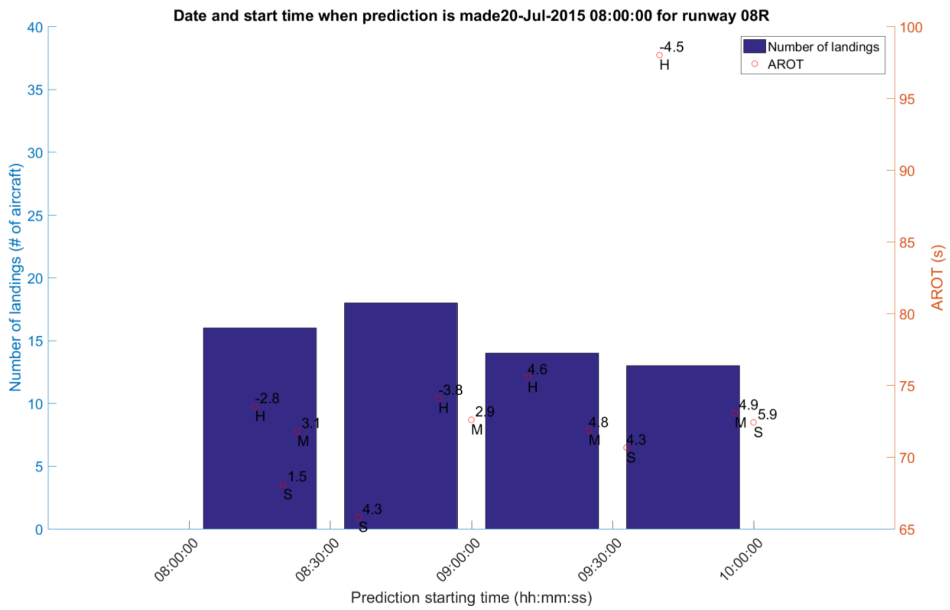


Figure 5.12: Output after running the application for the first trial.

It has been observed that for this case study and for each prediction trial, the first 30 min has a significantly lower error compared with the remaining 90 min prediction time. Table 5.3 shows these differences in percentages for four trials. Furthermore, it has been analysed that the results are statistically significant for four trials.

Table 5.3: Average error differences per trial and for the time prediction window 0–30 min and 30–90 min 0–30 min 30–90 min.

	0 - 30 minutes	30 - 90 minutes
Trial 1	7%	13%
Trial 2	5%	10%
Trial 3	4%	10%
Trial 4	3%	8%

5.5 Conclusion

This study demonstrates the use of combined ML techniques to forecast AROT per flight. The AROT aircraft behaviors, the data sources, key features, and ML outcome are first reviewed. Based on the availability of data and the importance of the problem, activities that will benefit from making greater use of data were then identified, where we focused on the AROT at runway 08R due to the largest number of flights designated at CDG.

A predictive model for AROT flights was built by combining the NN, MLP, lasso, and regression tree techniques. Then an approach was developed to generate distributions of each AROT flight and the number of landings to a specific runway (08R) within a time frame of 30 min. An application for CDG and VIE was also developed to produce these forecasts. Finally, a live trial at CDG is scheduled on the 11th of September 2017, which will allow us to assess the accuracy of the model and make improvements. For this a feasibility study will be set up, where we analyze how our predictability tool can be used by air traffic controllers in their decision making and planning to ensure resilience, safety, and efficiency of air traffic control operations locally in a sector, but also taking into account coordination with other sectors. Based on the available data, the feature selection techniques RrelliefF and Sequentialfs show that the following 10 features mostly influence the AROT behavior: QFU, typeavion, ArrRwyThroughput, Visibility, Crosswind, Tailwind, Temp, LongFlare, TimeReal, and CodeAeroportIATA. It has been also seen in a previous study [30] that the top three features are well known as highly correlated factors impacting AROT by using statistical analyses. From the regression tree, it is learnt that by knowing the top seven features in advance a good prediction can be made of the abnormal and normal AROT and for which each abnormal AROT flight will fall into one of the 17 precursor categories shown in Table 5.2. Furthermore, the regression technique performs best for finding associated precursors, for which the CART technique is used to fit a maximum tree depth and minimum leaf size of 6 and 16, respectively. There are some advantages associated to using our model. First, the ML technique used to build the model is fast, intuitive, and efficient. It can help the airport managers to understand the driving features of the AROT per runway. Second, our model has been built based on a large historical data set of 78,321 CDG and 500,000 VIE flights, for which 22 variables are available for selection as predictors. These variables also enable one to build new features using domain knowledge of the data. Third, our model can update the predictions in real time. The application developed for CDG and VIE can easily extract real-time data for both airports.

The forecasting procedure is effective and the predictions can be generated in a relatively short amount of time. Our model is the first to provide forecasts for each AROT flight to a specific runway. The forecasts of an AROT flight may help ATCOs to make better decisions, predict whether the flights will experience abnormal AROT, and to anticipate the aircraft sequencing on final approach by knowing when landing queuing starts. If an ATCO can retrieve this information far in advance, he/she may be able to generate more stable and accurate AROT to be used in A CDM (Airport Collaborative Decision making), which aims to improve the operational efficiency of all airport operators by reducing delays, increasing the predictability of events during the progress of a flight, and optimizing the use of resources. Although the model is developed for an AROT problem, the methodology proposed in this study can be easily applied to other runway processes, such as the prediction of unstable approaches. This will be a topic of our future research.

Acknowledgments

This work is supported by the EUROCONTROL Experimental Centre and the SESAR program funded by EU, EUROCONTROL, and the aviation industry. The authors would like to thank CDG and VIE airports for providing access to the aircraft final approach and runway radar data, and machine learning expert Massimiliano Zanin for his valuable insights. The authors also acknowledge the support and expertise of the technical teams that made this work possible.

References

[1] Logan, T. J., "Error Prevention as Developed in Airlines," International Journal of Radiation Oncology, Biology, Physics, Vol. 71, No. 1, Jan. 2008, pp. S178–S181. doi:10.1016/j.ijrobp.2007.09.040.

[2] Li, L., Das, S., Hansman, R. J., Palacios, R., and Srivastava, A. N., "Analysis of Flight Data Using Clustering Techniques," Journal of Aerospace Information Systems, Vol. 12, No. 9, 2015, pp. 587–598. doi:10.2514/1.I010329

[3] Herrema, F. F., Treve,V., Curran, R., andVisser, H. G., "Evaluation of Feasible Machine Learning Techniques for Predicting the Time to Fly and Aircraft Speed Profile on Final Approach," International Conference on Research in Air Transportation, Vol. 8, No. 4, 2016, pp. 4–8.

[4] Herrema, F., Curran, R., Visser, H., Sharpanskykh, A., Huet, D., Desart, B., and Lacote, R., "Taxi-Out Time Prediction Model Through Machine Learning Techniques at Charles de Gaulle Airport," SESAR Innovation Days Delft, Vol. 8, No. 8, 2016, p. 8.

[5] EUROCONTROL, "Arrival Runway Occupancy Time," 2012, http://www.eurocontrol.int/lexicon/lexicon/en/index.php/Arrival_runway_occupancy_time [last accessed 30 Sept. 2016].

[6] Amidan, B. G., and Ferryman, T. A., “Atypical Event and Typical Pattern Detection within Complex Systems,” 2005 IEEE Aerospace Conference, IEEE Publ., Piscataway, NJ, March 2005, pp. 3620–3631. doi:10.1109/AERO.2005.1559667

[7] Budalakoti, S., Srivastava, A., and Otey, M., “Anomaly Detection and Diagnosis Algorithms for Discrete Symbol Sequences with Applications to Airline Safety,” IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, Vol. 39, No. 1, 2009, pp. 101–113. doi:10.1109/TSMCC.2008.2007248

[8] Budalakoti, S., Srivastava, A. N., and Akella, R., “Discovering Atypical Flights in Sequences of Discrete Flight Parameters,” 2006 IEEE Aerospace Conference, IEEE Publ., Piscataway, NJ, March 2006, pp. 1–8. doi:10.1109/AERO.2006.1656109

[9] Srivastava, A. N., “Discovering System Health Anomalies Using Data Mining Techniques,” Proceedings of the 2005 Joint Army Navy NASA Air Force Conference on Propulsion, Charleston, SC, June 2005, pp. 1–11, https://c3.nasa.gov/dashlink/static/media/publication/Data_Mining_for_ISHM.pdf.

[10] Iverson, D. L., “Inductive System Health Monitoring,” Proceedings of the 2004 International Conference on Artificial Intelligence (IC-AI04), CSREA Press, June 2004.

[11] Schwabacher, M., Oza, N., and Mattews, B., “Unsupervised Anomaly Detection for Liquid-Fueled Rocket Propulsion Health Monitoring,” Journal of Aerospace Computing, Information, and Communication, Vol. 6, No. 7, 2009, pp. 464–482. doi:10.2514/1.42783

[12] Das, S., Matthews, B. L., Srivastava, A. N., and Oza, N. C., “MultipleKernel Learning for Heterogeneous Anomaly Detection: Algorithm and Aviation Safety Case Study,” Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, Washington, D.C., 2010, pp. 47–56.

[13] Matthews, B. L., Das, S., Bhaduri, K., Das, K., Martin, R., and Oza, N., “Discovering Anomalous Aviation Safety Events Using Scalable Data Mining Algorithms,” Journal of Aerospace Information Systems, Vol. 10, No. 10, 2013, pp. 467–475. doi:10.2514/1.I010080

[14] Kumar, V., Sherry, L., and Kicinger, R., “Runway Occupancy Time Extraction and Analysis Using Surface Track Data,” Vol. 10, No. 3, 2009, pp. 3–5.

[15] Lee, D. D., Smith, A., Cassell, R., and Abdul-Baki, B., “NASA Low Visibility Landing and Surface Operations (LVLASO) Runway Occupancy Time (ROT) Analysis,” Vol. 8, No. 2, 1999, pp. 25–30.

[16] Brardo, O., Casta, A., and Couturier, B., “Method and System for Assisting the Braking of an Aircraft,” 2017, http://www.freepatentsonline.com/20170129623.pdf [last accessed 28 July 2017].

[17] Lee, W., and Stolfo, S., “Data Mining Approaches for Intrusion Detection,” Proceedings of the 7th USENIX Security Symposium, USENIX Assoc., Berkeley, CA, Jan. 1998.

[18] Mukherjee, B., Heberlein, L. T., and Levitt, K. N., “Network Intrusion Detection,” IEEE Network, Vol. 8, No. 3, 1994, pp. 26–41. doi:10.1109/65.283931

[19] Bishop, C. M., "Novelty Detection and Neural Network Validation," IEE Proceedings of Vision, Image and Signal Processing, Vol. 141, No. 4, 1994, pp. 217–222. doi:10.1049/ip-vis:19941330

[20] Worden, K., Manson, G., and Fieller, N. R. J., "Damage Detection Using Outlier Analysis," Journal of Sound and Vibration, Vol. 229, No. 3, 2000, pp. 647–667. doi:10.1006/jsvi.1999.2514

[21] Brause, R., Langsdorf, T., and Hepp, M., "Neural Data Mining for Credit Card Fraud Detection," Proceedings of the 11th IEEE International Conference on Tools with Artificial Intelligence, IEEE Publ., Piscataway, NJ, 1999, pp. 103–106. doi:10.1109/TAI.1999.809773

[22] Bolton, R. J., and Hand, D. J., "Statistical Fraud Detection: A Review," Statistical Science, Vol. 17, No. 3, 2002, pp. 235–255. doi:10.1214/ss/1042727940

[23] Liao, T. W., "Clustering of Time Series Data – A Survey," Pattern Recognition, Vol. 38, No. 11, Nov. 2005, pp. 1857–1874.doi:10.1016/j.patcog.2005.01.025

[24] Gaber, M. M., Zaslavsky, A., and Krishnaswamy, S., "Mining Data Streams: A Review," ACM Sigmod Record, Vol. 34, No. 2, 2005, pp. 18–26. doi:10.1145/1083784

[25] Gariel, M., Srivastava, A. N., and Feron, E., "Trajectory Clustering and an Application to Airspace Monitoring," IEEE Transactions on Intelligent Transportation Systems, Vol. 12, No. 4, 2011, pp. 1511–1524. doi:10.1109/TITS.2011.2160628

[26] Chan, P. K., and Mahoney, M. V., "Modeling Multiple Time Series for Anomaly Detection," 5th IEEE International Conference on Data Mining (ICDM'05), IEEE Computer Soc., Houston, TX, 2005, pp. 90–97. doi:10.1109/ICDM.2005.101

[27] Keogh, E., Lonardi, S., and Ratanamahatana, C. A., "Towards Parameter-Free Data Mining," Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, New York, 2004, pp. 206–215. doi:10.1145/1014052.1014077

[28] Kumar, M., and Woo, J., "Clustering Seasonality Patterns in the Presence of Errors," Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, New York, 2002, pp. 557–563. doi:10.1145/775047.775129

[29] Möller-Levet, C., Klawonn, F., Cho, K.-H., and Wolkenhauer, O., "Fuzzy Clustering of Short Time-Series and Unevenly Distributed Sampling Points," Advances in Intelligent Data Analysis V, Vol. 2810, Springer, Berlin, 2003, pp. 330–340. doi:10.1007/978-3-540-45231-7-3-540-45231-7_31

[30] Herrema, F. F., Big Data Analysis and Machine Learning at Airports to Support Decision Making, Brussels, 2015, pp. 10–50.

[31] Chapman, P., Step-by-Step Data Mining Guide, SPSS, New York, 2000, pp. 50–76, https://www.the-modeling-agency.com/crisp-dm.pdf.

[32] Serial Metrics, "Machine Learning," 2013, http://serialmetrics.com/blog/what-is-machine-learning-feature-selection/ [last accessed 12 Feb. 2016].

[33] Robnik-Šikonja, M., and Kononenko, I., "Theoretical and Empirical Analysis of ReliefF and RReliefF," 2013, %%http://link.springer.com /article/10.1023%2FA%3A1025667309714 [last accessed 12 Feb. 2016].

[34] Robnik-Šikonja, M., and Kononenko, I., "An Adaptation of Relief for Attribute Estimation in Regression," No. 3, 2003, p. 9.

[35] Hyndman, R., "ARIMA Models," 2016, https://www.otexts.org/fpp/8 [last accessed 7 Nov. 2016].

[36] Arizona State University, Design and Analysis of Experiments, 5th ed., Thomson Digital, Tempe, Arizona, 2013, p. 449.

[37] Ning, H., "Overview Diagram of Azure Machine Learning Studio Capabilities," 2016, https://azure.microsoft.com/en-us/documentation/articles/machine-learning-studio-overview-diagram/

6.0 A machine learning model to predict runway exit at Vienna airport

T*This Chapter builds on the knowledge from Chapters 4 and 5. It focusses on the fourth ASP enabler or procedural and non-procedural Runway exit utilised (NREX). This Chapter presents a different combined ML technique. Furthermore, Chapter 5 did not focus on real-time operational visualisations for the ATCO. Therefore, the main purpose of this paper is twofold; the first is to extract the NREX as a function of operational parameters from historical data. Following this step, a prediction of the procedural or non-procedural runway exit taken with their associated risk precursors is performed. The identification and prediction is performed using the Classification Ensemble method. The key objective of this Chapter is to develop a real time visualization model that forecasts the procedural or non-procedural NREX for different classified aircraft using ICAO wake vortex categories (Heavy, Medium and Light) and weather conditions for VIE airport. This model should offer insight into the predictability of key precursors impacting runway exits taken and their impact on safety and the runway capacity provided.*

This chapter was previously published as:

Herrema, F.F., Curran, R., Hartjes, S., Ellejmi, M., Bancroft, S. and Schultz, M., 2019. A machine learning model to predict runway exit at Vienna airport. Transportation Research Part E: Logistics and Transportation Review, 131, pp.329-342.

Abstract
Runway utilisation is a function of actual yearly runway throughput and annual capacity. The aim of the analysis in this project is to find data driven prediction models based on the features and relevant scenarios that might impact runway utilisation. The Gradient Boosting machine learning method will be assessed on their forecast performance and computational time for predicting the procedural and non-procedural runway exit to be utilised after the landing rollout. The Gradient Boosting method obtained an accuracy of 79% and was used to observe key related precursors of unique data patterns. Tests were conducted using runway and final approach data consisting of 54,679 arrival flights at Vienna airport.

Keywords-component; Runway Utilisation, Runway Capacity, Runway Occupancy Time, Gradient Boosting

6.1 Introduction

Many of today's hub airports are at times unable to handle planned air traffic demand. Despite being saturated, some airports have political and environmental challenges associated with any further physical airport development. In view of expected further growth in air traffic demand, there is a clear need for safety and runway capacity improvements in an environmentally responsible manner. In order to enhance existing runway throughput, technology and procedures have enabled in certain circumstances reductions in legacy separation standards. Since demand for air traffic movements is continuously increasing, all stakeholders of the aviation system aim at a maximum utilisation for the given infrastructure, in particular the airport runways. A high number of runway movements entails the realisation of minimum separation standards between arrival and departing aircraft. With a view to avoiding accidents or risks of incidents, airline operators and Air Traffic Controllers (ATCOs) are moving toward proactive risk management which aims to identify and predict risk precursors and to mitigate the associated risks.

In this context, Arrival Runway Occupancy Time (AROT) is one major impact factor, which cannot be adapted or regulated in any manner. In fact, AROT is a combination of the resulting braking profile and the particular runway exit an aircraft utilises after the landing roll out. Landing or departing aircraft that immediately follow a landing aircraft can only use the runway once it has been vacated. If a landing aircraft misses a planned or foreseen exit or for whatever reason increases it’s AROT during the rollout phase, a tightly sequenced following aircraft will have to perform a missed approach and go around. These disturbances in the arrival/departure sequence will result in delayed operations for the scheduled movements. Due to the uncertainty surrounding AROT times, spacing buffers are routinely applied by ATCO so that separation standards are never infringed. Furthermore, the variation in the application of these buffers is down to the weather of the day, notably winds and precipitation but moreover the experience level of the controller and even if the aircraft is considered to be a locally based aircraft or airline. As a rule of thumb, minimum separation standards of 2.5 NM can only be achieved if landing aircraft vacate the runway in less than 50 seconds. The majority of Hub airports not only have active AROT management campaigns but also define runway exit management as a procedure within the Aeronautical Information Publication (AIP). These procedures will define the most suited exit for the aircraft type additionally, if unable to comply as published, they are to advise tower as soon as possible. These published procedures define the exits to be utilised however; a landing clearance implies the full-dedicated use of the runway. As such, an aircraft may land and exit the runway at an exit other than the exit defined by AIP procedure with or without prior ATCO coordination. Thus, it is of major importance to identify precursors for the probability of missed runway exits and landings with increased AROT. With our contribution, we provide a Machine Learning (ML) approach to predict the runway exit utilised based on actual movements at airports. Currently, there is no supplementary operational system that assists the Arrival Manager (AMAN) and Departure Manager (DMAN) on predicted runway exits and AROT. AMAN systems provide an automated sequencing support for approach and runway ATCOs, whilst continuously optimising arrival traffic sequences and runway slot times for landing aircraft. This is accomplished by a more efficient and predictable arrival management process that can assist in reducing low-level holdings and tactical intervention by the ATCO. AMAN takes into account the locally defined maximum landing rate (capacity), the required separation for aircraft in the touchdown zone (safety) and additional operational criteria. DMAN is an advanced controller tool for optimising runway throughput and taking maximum benefit from the available runway capacity. To achieve optimal use of runway capacity and airspace capacity in the Terminal Management Area (TMA), a DMAN assists the ATCO in managing departure traffic by providing optimised take-off sequences in considering departure trajectories. AMAN and DMANs are essential controller tools that provide guidance and as such ensure the best use of the available runway capacity (i.e maximum through put). An additional support tool providing real time AROT alerts would be an advantage if not a necessity in a future environment of High Intensity Runway Operations (HIRO) where the associated risk of a loss of separation between aircraft in time and/or distance has a direct impact on incident and accident avoidance. A live trial should give insight into the risk mitigation.

6.1.1 Related work and Interview with Vienna ATCO

We divide the literature in two groups. We start with the first group which focuses on the work to be performed on runway capacity enhancements and methods to predict exit usage. In the context of efficient runway operations, the AROT is an important driver.

AROT along with the runway exit utilised is key in quantifying actual throughput and thus generating predictions with respect to a runway utilisation indicator. For certain predictions Machine Learning techniques can be used. Previous studies have explored and applied ML techniques using radar and A-SMGCS data, but the identification and prediction of the runway exit used along with the observation of related precursors are not well developed. A statistical analysis of the final approach and AROT is done by [1] using data from the Detroit multilateration surveillance system. A study on surveillance data highlighting benefits and including different sources of information to improve capacity and safety is conducted [2].

Several operational factors and their impact were analysed in [3]. In [4], a model to predict the landing performance of airplanes is developed with a focus on locating high-speed exits. This model was based on empirical heuristics, which were derived from field observations, as a different mix of aircraft and different environmental conditions at airports will result in specific approaches for runway exit designs. In [5, 6], a model for optimally tailored runway and exit layouts is proposed whereas [7] provides the airport taxiway structure and links it to the runway exit choice process. The runway utilisation depends on the runway used as well as several additional factors (e.g. number of arrivals and departures or runway configurations, efficiency of taxi operations) [8]. Furthermore, efficient airside operations will depend on a balanced consideration of capacity/demand management [9], aircraft/runway scheduling [10], taxiway planning/ground movements [11] and gate assignment [12], which clearly emphasise the demand for efficient runway exit selection for landing aircraft. In this context, [13] provides an operable calculation method to manage the runway exit availability considering uncertain exit usage and exit times. In [14], an analysis method for medium-speed manoeuvres and more specifically, runway exit manoeuvres is presented. A Monte Carlo simulation algorithm and empirical heuristics derived from field observations were used in [15] to estimate landing-roll trajectories and to predict aircraft landing performance on runways in order to locate high-speed exits. In [16] an application was designed that relates to the optimisation of runway exits based on assessment of runway conditions and aircraft-based braking capability, with the aim of selecting the best runway exit to optimise runway throughput.

The second group focuses on interviews with Vienna (VIE) ATCOs and how they subjectively identify lacking predictability on managing runway operations. They would welcome a support tool that warns controllers on the following event predictions:

- Missed Procedural Runway Exit (MPRE)[5]
- When the leader is at threshold and the follower has a time separation less than 40 seconds[6].
- When the leader has an AROT of more than 50 seconds[2] and the follower is at the required ICAO minimum separation[7].
- When the time window available to accommodate departure traffic during mixed-mode HIRO.

An indication will be provided when an event is detected and when the prediction accuracy of that event is higher than 80% during HIRO[3]. This first paper will only focus on the prediction of the target variable procedural or non-procedural runway exit to be used (NREX). A procedural exit for VIE Runway 34 (RWY34) is defined as a Heavy aircraft type that takes exit B5/B4, a Medium type that take exit B7/B5 or a Light types that utilises at B9/B7 (Table 1 and Figure 1). We refer to non-procedural when a flight vacates the runway at an exit further along than the Aeronautical Information Publication (AIP) intended one. For example an alarm

[5] MPRE is defined as an aircraft on the runway that missed his intended runway exit from AIP. The intended runway exit of the ATCO is the optimal for maintaining the desired throughput.

[6] Based on the interview with Vienna ATCO.

[7] International Civil Aviation Organization (ICAO) 'Application of separation minima' (NAT Doc 008) and based on the interview with Vienna ATCO. Each flight falls into one of the ICAO categories ‘Light’, ‘Medium’ or ‘Heavy’.

is raised when the tool predicts with an accuracy of 80% or greater that a Light[3] or Medium[3] aircraft type will vacate Runway 34 (RWY34) via exit B1, B2 or B4, or when a Heavy[3] aircraft vacates at B2 or B1 (see Table 6.1 and Figure 6.1) provided the model has a proven accuracy of 80% or greater based on the last tree update. Thereafter, the ATCO decides based on experience if a go-around is required for the following landing aircraft. In this context and based on the interviews with VIE ATCOs, this paper will focus on two different time window predictions during real time landing operations. The following predictions could lead to an improvement in runway safety and throughput.

- The first prediction will be produced at 2NM upstream from the runway threshold, enabling tactical operations tools to be fed and ATCOs to be warned about the four events identified above, together with their associated precursors.
- The second prediction will be related to the period between time of separations and the next 1 or 2 hours, and can be used either as a strategic operational tool for the ATCO supervisor to make a decision on the final approach sequence or can be used as input for the AMAN and DMAN.

Table 6.1: Procedural AIP exit[4] at RWY34 for VIE airport.

Aircraft ICAO category	**Procedural exit**	**Non-procedural exit**
Heavy	B4, B5	B1, B2
Medium	B5, B7	B1, B2, B4
Light	B7, B9	B1, B2, B4, B5

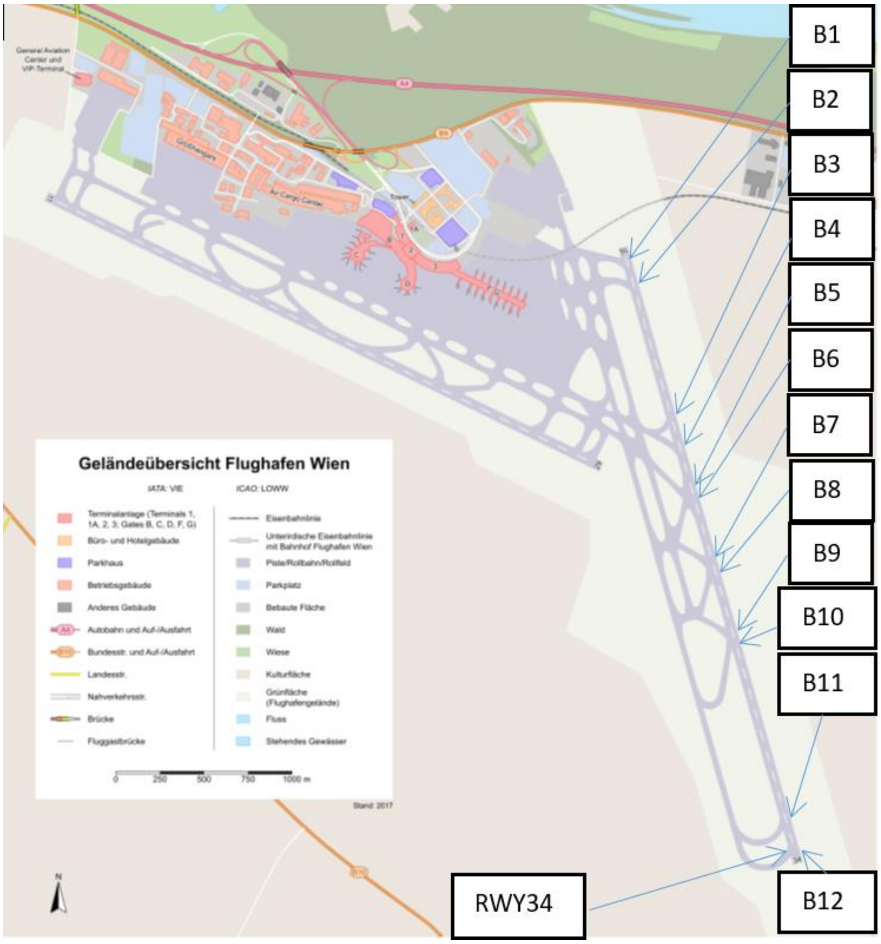


Figure 6.1: VIE Runway Design. The left green arrows in the lower left picture show the possible runway exit utilisations for RWY34.

Based on ATCO requirements and experiences NREX predictions will be made during HIRO. Vienna's RWY34 has a maximum capacity of 30 landings per hour during mixed-mode operations. As a consequence, HIRO are defined as 25 landings or more on RWY34 during the last 60 minutes from prediction at time (*t*). The most common configuration of HIRO (RWY34, mixed-mode operations, configuration D) was selected for the use case. Based on ATCO practical experience with RWY34 operations, 70% of the flights take the NREX, of which Heavy aircraft types take exit B5/B4, Medium types B7/B5 and Light B9/B7 (Table 6.1 and Figure 6.1). The actual exit used on RWY34 may be different from the AIP published procedures typically, this will occur when the runway is contaminated and/or during an increase in tailwind.

Based on a literature review and interviews with VIE ATCOs, the following shortcomings in NREX prediction methodology were identified; first, extracting principal components using many aircraft operational runway variables has as yet, not been considered.

Second, as yet, no algorithm has been developed that is capable of real-time computation and to provide the classifications associated with the individual NREX predictions to synthesise the what-if statements. In this paper we propose a state of the art ML method – Gradient Boosting (GB) which is capable of synthesizing the what-if statements and is expected to overcome the noted lacking aspects from literature. In [17], the characteristics of these techniques are reviewed and their added value is explained for real-time and what-if statements computations.

6.1.2 Aim

The main purpose of this paper is twofold; the first is to extract the NREX as a function of operational parameters from historical data. Following this step, a prediction of the procedural or non-procedural runway exit taken (NREX) with their associated risk precursors is performed. The identification and prediction is performed using the GB method. The key objective of this study is to develop a real time model that forecasts the procedural or non-procedural NREX for different classified aircraft using ICAO wake vortex categories (Heavy, Medium and Light) and weather conditions for VIE airport. This model should offer insight into the predictability of key precursors impacting runway exits taken and their impact to the runway capacity provided.

6.1.3 Structure

The structure of this paper is as follows. First, the case study is presented in section 6.2. Second, the methodology is outlined in section 6.3 whilst introducing the NREX behaviour and data preparation. Furthermore, the results for the best performed ML technique (Boosted trees) are shown. In section 6.4 the prototype model is outlined and finally conclusions and recommendations are drawn in section 6.5 and 6.6.

6.2 Case Study

The Runway Utilisation (RU) use case is a function of actual yearly runway throughput and annual capacity. We will build an algorithm that would support the tower ATCO during HIRO by making predictions based on historical observations of runway traffic. For these predictions we use the GB method to identify patterns and to observe related risk precursors in historic data [18, 19]. These patterns and precursors can be transferred into 'what-if' statements by analysing relations between the target and the prediction variables.

The RU use case is part of work being conducted by SafeClouds[8] for which four building blocks have been defined. Each building block represents a part of the final algorithm. The key aim of each building block is to: (1) predict the runway exit to be used; (2) predict the time from overflying the runway threshold until tail clear of the runway; (3) predict the time

[8] www.safeclouds.eu

separation of the follower when the leader is at the runway threshold and finally (4) enhance the DMAN sequencing by predicting the time window available to release a departure during mixed-mode[9] HIRO. This paper will only focus on the NREX prediction. After integration of the four individual building blocks, the final algorithm could be used as input for the Vienna AMAN and DMAN system.

The ML techniques will be addressed by modelling the predicted NREX during mixed-mode operations at Vienna airport (RWY34, mixed-mode operations, configuration D). When configuration D is operational, no traffic is allowed for the second runway (RWY11/29). In total we have 54,679 arrival flights and 13,745 departure flights. For this case study when only take the arrival flights into account. The NREX are categorical variables that depend on the prediction variables described in Table 6.2. Our four step approach is based on learning a model for RWY34 and six different NREX classifications during different weather conditions. First, the NREX is extracted for each arrival flight. Second, all ICAO[3] flight categories (Heavy, Medium and Light) are classified into taking their procedural or non-procedural AIP intended exit. Third, during HIRO conditions we examine one specific ICAO category. The first category chosen is the Medium type since this is the most common ICAO type operating at Vienna airport. Finally, we identify and predict NREX procedural or non-procedural per flight and observe key precursors impacting this variable.

6.3 Methodology

For this study, we propose a methodology composed of six steps. The method is based on our previous studies [18, 19] and implements the Statistical Package for the Social Science (SPSS) ML method [20]. This methodology describes the steps to come up with a usable predictability model that identifies and predicts NREX per aircraft pair and observes key precursors impacting the runway exit taken. Details are given for each step in the following paragraphs.

6.3.1 Final approach and Runway data sources

The aircraft performance on the ground is a key driver for runway throughput, especially when reduced airborne separation minima are applied. Several factors, such as aircraft type, weather conditions (wind and visibility), traffic demand and air traffic controller workload influence NREX. In order to predict NREX and extract risks precursors, final approach and runway radar data are used.

[9] Mixed-mode operations, where take-offs and landings can take place at the same runway.

Radar[10], A-SMGCS[11], Wind profiler[12], SNOWTAM[13], SODAR[14] and METAR[15] data are extracted from runway schedulers and have been provided by VIE airport. The data set covers 2 years of final approach data from 2014 to 2016. In total, the data comprises 54,679 arrival flights for RWY34. For these flights we have all the data listed in Table 6.2. The prediction variables are selected by taking into account the selected features for predicting the runway exit utilised at Orly airport. For this study we have performed a feature selection study to understand which features are needed and are most important for making the prediction at 2NM. The most important features were selected using the GB feature selection property (*OOBPermutedVarDeltaError*).

Table 6.2: Prediction variable 1 till 14 and target variable 15.

Variables	Description	Mean	Max	Min
1. ACType	Aircraft Type (binary variable)	-	-	-
2. ICAOcat	Aircraft ICAO Category (binary variable)	Medium	-	-
3. Height2NM	Aircraft height above Vienna terrain at 2NM	2.01NM	2.30NM	1.80NM
4. Height5NM	Aircraft height above Vienna terrain at 5NM	5.01NM	5.32NM	4.78NM
5. GroundSpeed2NM	Aircraft Ground Speed at 2NM	130kts	155kts	110kts
6. GroundSpeed5NM	Aircraft Ground Speed at 5NM	168kts	215kts	140kts

[10] Radar is an object-detection system that uses radio waves to determine the speed, altitude and direction of an aircraft.

[11] A-SMGCS is a system providing guidance, routing and surveillance for the control of aircraft.

[12] Wind profiler is a type of weather observing equipment that uses radar and sound waves to detect the wind speed and direction at different altitudes.

[13] SNOWTAM notifying the presence, or removal, of hazardous conditions due to snow, ice, slush or standing water.

[14] SODAR measure the scattering of sound waves by atmospheric turbulence.

[15] METAR is a format for reporting weather information.

Variables	Description	Mean	Max	Min
7. Visibility	Horizontal visibility in meters	5km	Infinity	100m
8. Cloud1	The height of the first cloud layer in meters	2km	Infinity	100m
9. Cloud1Okta	How many eights of the sky are covered in cloud, ranging from 0 oktas (completely clear sky) through to 8 oktas (completely overcast). (binary variable)	5	8	0
10. SODARVelocity	SODAR Velocity at 200 meter from the ground (2NM before the threshold)	7kts	25kts	2kts
11. SODARheadwind	SODAR headwind at 200 meter from the ground	340°	340°	0°
12. WMAheadwind34	Headwind for RWY34 at 15 meter (50ft) from the ground	340°	340°	0°
13. WMAWindSpeed34	Wind speed for RWY34 at 15 meter (50ft) from the ground	5kts	26kts	0kts
14. Throughput	The amount of landings performed on the runway using the Estimated Time of Arrival (ETA) during the last 60 minutes.	28	32	25
15. NREX	Procedural (1) or non-procedural (0) runway exit taken	80% procedural flights and 20% non-procedural	-	-

6.3.2 Data preparation

The data preparation phase covers all activities required to set up the final dataset from the initial primary aircraft operational runway and final approach radar data. Before the model will be trained, first the data set was cleaned by Austrocontrol (Austrian air navigation service provider) on missing values. As a next step the AROT is extracted by calculating the elapsed time between the aircraft crossing the threshold and its tail vacating the runway [16], which is received from the variable *AircraftOnTheRunway*. The variable *AircraftOnTheRunway* shows for each second a "1" if the aircraft is on the runway and a "0" if it doesn’t. The Runway Exit (RET) utilised is extracted by subtracting the utilised Latitude/Longitude RET coordinate with each possible RET Latitude/Longitude coordinate for RWY34. The minimum absolute value defines the RET used. Figure 6.2 shows an example of the AROT versus the number of flights per ICAO category for different RET utilised for RWY34. The RET is needed to extract the NREX. The red vertical dashed lines show the average AROT per ICAO category (Heavy, Medium and Light).

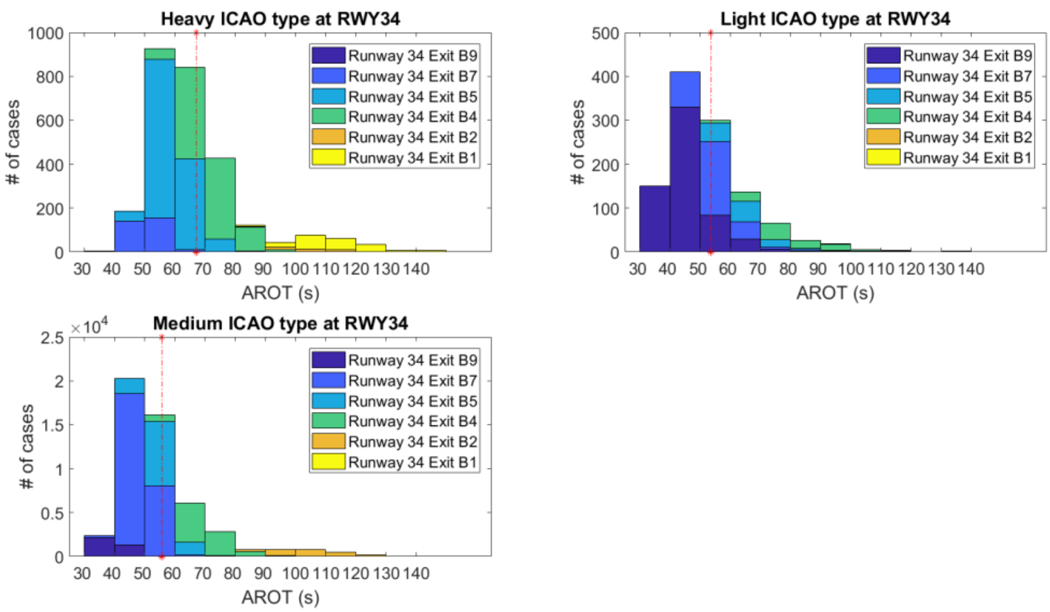


Figure 6.2: Number of cases versus the AROT for RWY34 and runway exits for ICAO categories Heavy, Medium and Light during HIRO.

The attentive reader will notice that the distance to arrival is absent from the data, therefore making the identification of the point of prediction (2NM) wanted in our forecasting problem difficult. It is thus necessary to format and complete the original data. Therefore, the following modifications have been done to the dataset:

Distance from arrival: The vector of distances of the flight (at each time stamp) D_i^* from the reference point T_{thres}, which indicates the threshold of RWY34, is added. We define the threshold T_{thres} coordinates (X_{thres}, Y_{thres}, Z_{thres}) as (48.092222°, 16.596667°, 597ft). For each trajectory point, D_i^* is calculated as the great-circle distance from T, using the Haversine formula:

$$A = sin^2\left(\frac{X_i - X_{thres}}{2}\right) + cos(X_i) \cdot cos(X_{thres}) \cdot sin^2\left(\frac{Y_i - Y_{thres}}{2}\right)$$
$$D = 2 \cdot H \cdot atan^2\left(\sqrt{a}, \sqrt{1-a}\right)$$

With coordinates in radian and $H = R + E_{VIE}$, R being the mean radius of the earth and E_{VIE} = 597ft being the elevation of RWY34.

Flight Level to Height: Merging the data has also allowed to link all flights with a specific atmospheric pressure measured and reported in the METAR as the QNH (Query: Nautical Height). It is possible to approximate the height from the Flight Level (FL) using the following formula:

$$Z_i(NM) = Z_i(FL) \cdot 100 + \frac{288.15}{0.0065 \cdot 0.3048}\left(\frac{QNH^{\frac{0.0065 \cdot 287}{9.81}}}{1013.25} - 1\right)$$
$$Height = Z_i(NM) \cdot \frac{E_{VIE}}{6076}$$

Note that 1NM equals 6076ft.

Angle to runway: The angle with respect with direction of the runway has been computed. It shows the variation in trajectory with respect to the direction of the runway - and we suspect it to be indirectly correlated with the adherence to the Instrument Landing System (ILS). The angle, in radians, is calculated with respect to the threshold coordinates as:

$$\emptyset_i = arctan\left(\frac{X_i - X_{thres}}{Y_i - Y_{thres}}\right)$$

Unwanted noise: It was recommended by Austrocontrol to apply additionally data smoothing techniques for eliminating unwanted noise in the data. Therefore, we use additional the Median Absolute Deviation (MAD) outlier detection technique which identifies data points that are significantly different from the rest of the data [21]. For each feature with vector *A* and *N* scalar observations, the MAD is defined as:

$$MAD = median(|A_i - median(A)|)$$

For $i = 1,2, \ldots., N$

The scaled MAD is defined as:

$$c \cdot median(abs(A - median(A)))$$

Where $c = \frac{-1}{(sqrt(2) \cdot erfcinv(\frac{3}{2}))}$

By having the MAD, outliers are extracted as data points that are more than three scaled MAD away from the median.

Unwanted flight scenarios: Additionally, we observed two flight scenarios in our data set that shouldn't be included in the model. These identified flights are filtered out from the entire data set and are defined as follows:

1) When for a single flight two individual time series of ones are detected for the variable *AircraftOnTheRunway;*

2) When for each 1 NM segment - from 5 until 0NM before threshold - the flight level has not decreased with more than 200 feet (go-arounds).

Holdout validation method: The data has been divided into 45,679 flights and 15 different aircraft and weather indicators which are shown in Table 6.2. Next, we construct the datasets and find the stability of two different data parts, known as the *holdout validation* method. The method is recommended for large data sets [22] and the method starts with randomly assigning data points to two different sets. We specify the default ratios for splitting the data into training and validation datasets as, respectively, 75% and 25% of the available data. A matrix *X* is then formed, where each row represents a flight *Y*, column 1 until 14 a prediction variable and column 15, the NREX target variable (see Table 6.2). Based on different data sources and the variables listed in Table 6.2, standardize the feature matrix (*X*) for all the flights (*Y*). Split the matrices *X* and *Y* in two subsets X_{train}; Y_{train} used to train the model and X_{val}; Y_{val} used to evaluate the model accuracy.

6.3.3 Feature selection

The most important features will be selected using the *OOBPermutedVarDeltaError* feature selection property for identifying which features are important for best describing the variance in the Vienna data set [23]. For all 14 prediction variables, the measure is the increase in prediction error if the values of that variable are permuted across the out-of-bag observations. This measure is computed for every tree, then averaged over the entire ensemble and divided by the standard deviation over the entire ensemble. In previous papers, we have introduced this technique [18, 19]. The objective of feature selection is three-fold: improving the prediction performance of the predictors, providing faster computational performance and more effective predictors [24] as well as providing a better understanding of the underlying process that generated the data [25]. As shown in Figure 6.3, the final OOBPermutedVarDeltaError model has the following top 12 predictors;

Throughput,SODARVelocity, SODARDirection, WMAWindSpeed34, GroundSpeed5NM, Height5NM, Cloud1, Visibility, Height2NM, GroundSpeed2NM, ICAOcat and ACType. The variables WMAWindDirection34 and Cloud1Okta have been excluded from the model since they contribute significant lower to the accuracy of the model. Thereafter we project the validation data onto the reduced dimensions of the training data. We are projecting any new validation data onto the training data subspace to compare and classify (*holdout validation*).

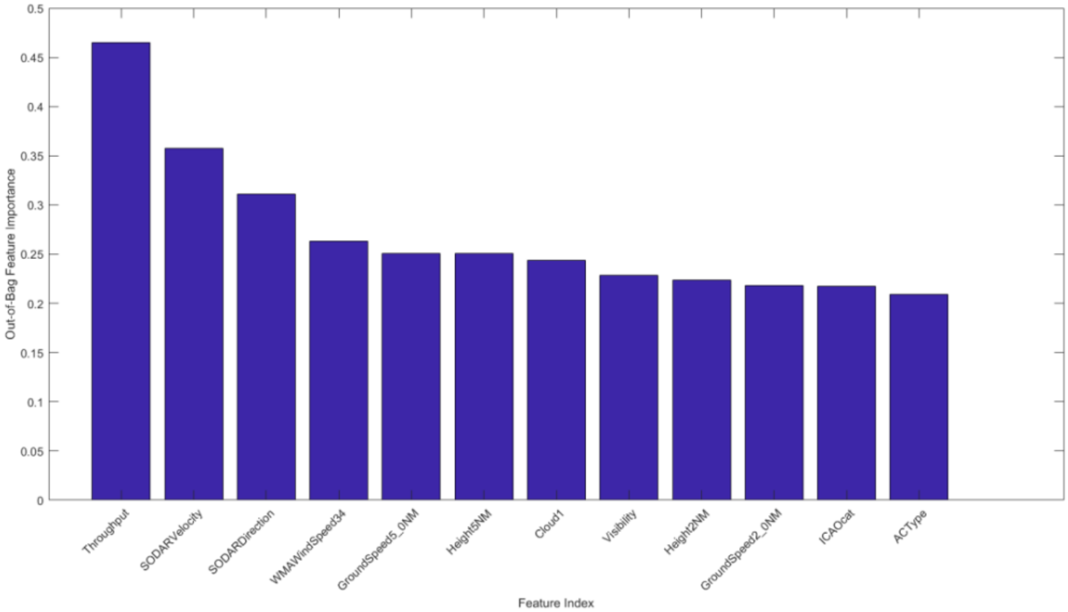


Figure 6.3: Normalized feature selection using *OOBPermutedVarDeltaError* algorithm (see Table 6.2 for feature/variable definitions).

6.3.4 Suitable ML classification techniques

To choose suitable ML techniques for the NREX prediction we have to take into account the size, quality and nature of the data. ML techniques can be classified into different categories following three main strands; unsupervised learning, supervised learning and reinforcement learning. Supervised learning can be divided into two different subcategories which are classification and regression. In this study, there will be a primary focus on classification learning since these methods are used for predicting binary numbers (NREX) whereas regression is often used for the prediction of real numbers such as the time to fly on final approach. For supervised classification learning, we use the GB method [18] approach that can be considered as a baseline for this study. In general, combining multiple classification models increases predictive performance and robustness of the model [17]. Individual decision trees tend to over fit. In GB one tries to combine the models produced by several *trees* into an *ensemble* that performs better than the original trees. GB decision trees combine the results of many decision trees, which reduce the effects of overfitting and improves generalization [26].

For the GB method we take into account the characteristics from the obtained data sets such as dealing with binary and real numbers within the data [17]. Furthermore in previous work [27], the GB technique was introduced to predict the time to fly on final approach. The technique was assessed on their capabilities to produce fast and accurate predictions and to test a number of 'what-if' statements.

A possibility to achieve this is by using the outcome of the GB method. The GB was built to synthesize the what-if statements for the predicted NREX taken and to give a clear overview for the ATCO for what is likely to happen under certain situations. The method is based on expert studies [28, 29] and will be explained below in three steps:

ML method

- First we learn the NREX for 54,679 flights with 12 features and the GB method. Three models will be learned using the GB algorithm.
- Second we assess the NREX results of all three models on their accuracy, F1 score, Area Under the Curve (AUC) and computational time.
- Finally we show the results for the best performed model and top 12 features from the previous step.

6.3.5 Assessing GB method

Three GB models were developed, one for each ICAO category. As mentioned in section 6.2, we applied feature selection on the training data to reduce the number of dimensions and predictors. Thereafter we assess the GB on their forecast performance and computational time with the top 12 predictors included. The forecast performance includes the metrics: accuracy, F1 score and the Area Under the Curve (AUC).

The classification accuracy depends on the number of samples correctly classified for each model. The accuracy is calculated as follows:

$$accuracy = \left(\frac{True\ Postives\ (TP) + True\ Negatives\ (TN)}{True\ Postives\ (TP) + False\ Positives\ (FP) + FalseNegatives\ (FN) + True\ Negatives\ (TN)}\right)$$

The F1 score is calculated as follows:

$$F1 = \left(\frac{2 * precision * recall}{precision + recall}\right)$$

Where precision is

$$Precision = \left(\frac{TP}{TP + FP}\right)$$

And recall or sensitivity

$$Recall = \left(\frac{TP}{TP + FN}\right)$$

The AUC score is calculated as follows:

$$AUC = sensitivity\ vs\ specifity$$

Where specificity

$$Specificity = \left(\frac{TN}{TN + FP}\right)$$

Table 6.3 and Table 6.4 shows the results for all three models.

Table 6.3: Forecast and computational time performance for the GB technique.

	Forecast performance (accuracy in %)	**F1 score**	**AUC**	**Computational time (s)**
Heavy ICAO category model Procedural Non-Procedural	98 75	Precision = 97.8 Recall = 98.3 F1=**98**	0.81	5.5 4.6
Medium ICAO category model Procedural Non-Procedural	95 **79**	Precision = 95.8 Recall = 95.6 F1=95	**0.82**	5.5 5.3
Light ICAO category model Procedural Non-Procedural	98 73	Precision = 94.3 Recall = 98.4 F1=96	0.78	3.5 2.9

Table 6.4: True classes versus the predicted classes for RWY34 and six different classifications during HIRO.

True exit class	Heavy Non Procedural (0)	TN 984 flights 6.0%	FP 328 flights 2.0%				
	Heavy Procedural (1)	FN 246 flights 1.5%	TP 14844 flights 90.5%				
	Light Non Procedural (0)			TN 733 flights 13.4%	FP 268 flights 4.9%		
	Light Procedural (1)			FN 71 flights 1.3%	TP 4396 flights 80.4%		
	Medium Non Procedural (0)					TN 4232 flights 13.0%	FP 1148 flights 3.4%
	Medium Procedural (1)					FN 1213 flights 3.7%	TP 26212 flights 79.9%
		Heavy Non Pro-cedural (0)	Heavy Pro-cedural (1)	Light Non Pro-cedural (0)	Light Pro-cedural (1)	Medium Non Pro-cedural (0)	Medium Pro-cedural (1)
		Predicted class					

13.0%+3.4% are all Medium Non Procedural flights. Therefore, the accuracy of predicting at 2NM if a Medium ICAO flight is non Procedural = 13.0/(13.0+3.4)*100% = **79%**

Table 6.4 shows the true class for the six classifications in the rows, whilst the columns show the predicted class. The diagonal cells show where the true class and predicted class match. If these cells are green, the classifier has performed well and classified observations of this true class correctly.

Each blue box in Table 6.4 counts 4 cells. The sum of the four cells is the total amount of flights per ICAO class; Heavy_(Non-Procedural), Medium_(Non-Procedural) or Light_(Non-Procedural). The second row from above shows all Heavy_Procedural flights with true exit class B4, B5 and B7. In this row, 90.5% of the flights are correctly classified, so

90.5% is the true positive rate for correctly classified points in this class, shown in the green cell in the True Positive Rate (TPR) column. The other flights in this row are misclassified: 1.5% of the flights are incorrectly classified as Heavy_Non-Procedural, 1.5% is the False Negative Rate (FNR) for incorrectly classified points in this class, shown in the red cell. Out of all Light flights, 93.8% is correctly predicted and 6.2% is wrongly predicted, for Medium flights this is respectively, 92.9% and 7.1%.

For this paper, we are only interested in the flights that didn't utilise their AIP proposed runway exit. After analysing the forecast performance and the computational time we conclude that the Medium ICAO category model performs best.

Based on these results we can conclude that when an indication will be provided at 2NM for Medium aircraft types, the ATCO should have confidence (with a *Non-Procedural* accuracy of 79%, see explanation in Table 6.4 that this flight will take runway exit B1, B2 or B4. Our model gives an accuracy of 81% for all validation flights (accuracy formula at the beginning of this section). It has also been tested that a lower accuracy is obtained without applying feature selection. However, by applying feature selection the model is trained 5% slower.

6.3.6 Observe risk precursors with Gradient Boosting

The purpose of building a Boosted tree is to identify a set of if-then-else (what-if statements) split conditions in order to identify the main risk precursors that most influence NREX. The observed flights for RWY34 are analysed and used to build three Boosted trees which should give a good understanding of which features influence the NREX choice. The hyperparameters for GB include the number of decision trees in the forest and the number of features considered by each tree when splitting a node. The model performed best using 95 decision trees and 12 features. Table 6.5 shows[16] an example for a medium procedural and non-procedural flight. By building this tree we start at the root node and ask a sequence of questions about the predictors. For each iteration, the tree chooses the variable and the split point to achieve the minimum *node error* between the predictions and the NREX targets. Each of the terminal leaves represents one of the partitions of the input space. To provide a model that can generate accurate predictions and is not overly-complicated, we need to find the optimal tuning parameters for the tree. In this study we use two parameters. The first parameter is the *minimum leaf size* (l_{min}), for which it is required enough data points in each terminal node to create a distribution. The parameter minimum leaf size can be used to stop the splitting process when the number of instances in a leaf is too small. In addition, if the tree contains too many variables, it is hard to interpret. The second tuning parameter for the tree is the *maximum tree depth* (d_{max}). A very large tree with many leaves might over-fit the data, while a small tree might not be able to capture the important structures of all the variables. The maximum tree depth can restrict the number of layers of a tree. While cross-validation is used to select the minimum leaf size, l_{min}, the minimum *node error* is used to select the maximum tree

[16] Note that we didn't show the tree due to its complexity. Therefore, we only show precursors for a Medium_Non-Procedural flight in Table 5, extracted from the tree.

depth, d_{max}. So in our case the tree is fit for a range of values of the two parameters to three quarters data. Thereafter, the error of the predictions is computed on the remaining one quarter. This is done for each quarter of the data, and the four error values are averaged. The set of parameters that gives the lowest error will be selected.

As shown in Figure 6.4, we first train the tree with different settings of l_{min}. We observed that there is an optimal point for the cross-validated error (error). The tree with a minimum leaf size between 130 and 140 performs best compared to other l_{min} numbers. For d_{max} we use the cost-complexity pruning technique which is based on [32]. Pruning reduces the complexity of the final classifier and hence improves predictive accuracy by reducing overfitting. The technique generates a series of trees where T_0 is the initial tree and T_N is the root in isolation. At step *j*, the tree is created by removing a subtree from tree *j*-1 and replacing it with a leaf node with value chosen as in the Boosted tree building algorithm. The subtree that is removed is chosen as follows. Define the error rate T over data set S as error(T,S). The subtree that minimizes T, is chosen for removal. Once the series of trees has been created, the best tree is chosen based on generalized accuracy as measured by cross-validation. We change the values of d_{max} and l_{min} and repeat the cross-validation process described above. The tree with l_{min} = 135 still performs slightly better than the others, and the misclassification does not change significantly after tree depth reaches the value 11. Thus, our final ultimate model, has 12 predictors (reduced by OOBPermutedVarDeltaError feature selection) and is fitted with a maximum tree depth and minimum leaf size set to 11 and 135, respectively.

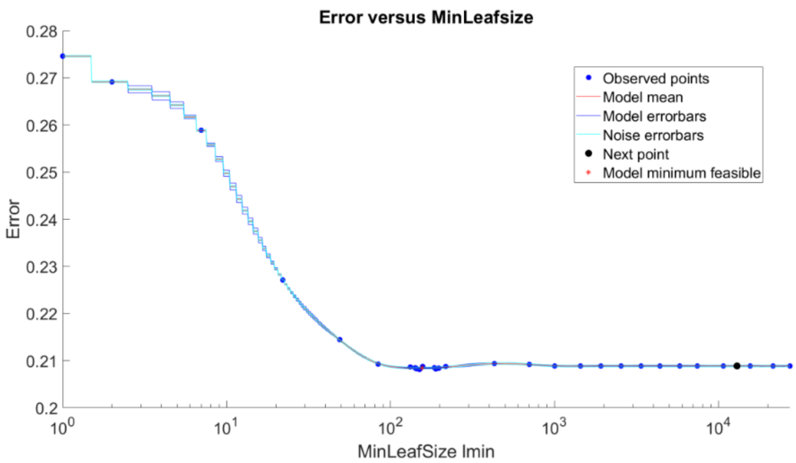


Figure 6.4: Error versus minimum leaf size (*lmin*) for Medium aircraft types.

By learning the tree, a mean and distribution are extracted per decision node. This is needed to observe risk precursors and understand what is likely to happen for the runway exit taken. Our model divides all the Medium NREX possible outcomes into 28 segments. In other words, the tree has 28 terminal nodes for which the outcomes are classified to Medium_Procedural and Medium_Non-Procedural. We can interpret the most important predictors as the major factors that play key roles in influencing NREX selection. After the tree is learned with a tree depth of 11, we observed for the 28 NREX categories their related precursors and risk impact. Table 5 shows only the top 5 precursors that end up with a non-procedural exit.

Table 6.5: Example of precursors for a Medium_Non-Procedural flight.

Non-procedural exit	Precursors	Number of times observed in the tree	Normalized impact value
1	Throughput<28	5	0.46
2	Cloud1<8750	4	0.25
3	GroundSpeed2.0NM>=147kts	3	0.23
4	WMAWindSpeed< 29kts	2	0.28
5	GroundSpeed5_0NM>= 155kts	2	0.25

The entire tree for each model (Heavy, Medium and Light) can be received on request via the main authors email address. For example the tree for the Medium ICAO category includes for each terminal node (28 in total):

1. Procedural or Non-Procedural;
2. precursors;
3. node error;
4. node probability and
5. node risk.

6.4 Prototype model

Based on what we have learned in the previous steps and the data availability, a prototype model has been developed using the GB to forecast NREX selection at 2NM before the RWY34 threshold at Vienna airport. The aim is to generate forecasts for each arrival flight on RWY34 per time window. Suppose we are at time *t*, and try to make predictions for the next *x* minutes. Given real-time flight information at 2NM from the RWY threshold, our GB model will determine which segment the flight belongs to. The GB tree will recognise, by receiving similar structured (X_{train}; Y_{train}) real time data in which leaf node a certain flight will fall with a given node; error, probability and risk.

To generate real-time NREX predictions from the model, we are developing an application using Python. The predictions are automatically presented in Comprehensive Airport Simulation Technology (*CAST*) to visualize real-time predictions. The CAST dashboard has been deployed in Austrocontrol operations and has currently one operational user. Figure 6.5 shows an example of the CAST dashboard.

Figure 6.5: Predictive NREX visualisations in CAST for RWY34 at Vienna. This example shows that an arrival flight is likely to utilise at exit B2.

In a live trial, the operational datasets can be accessed in real-time for which we produce the classification accuracies of the NREX for each arrival flight. However, in the predictive visualisation tool (Figure 6.5) we only show an expected Non-Procedural runway exit taken. The example during mixed mode operations shows arrivals (red line) and departures (green line). The ICAO light arrival aircraft is expected to utilise RWY34 at exit B5. Therefore an alert is indicated for the controller. It should be noted that before executing the live trial, the GB tree was already learned with 2 years of historical data. By making real-time time predictions on a particular day the tree has to be learned with a minimum of two hours of data from the day of prediction.

Suppose the trial started at 0800 hrs. We first collected data of the flights that arrive at VIE RWY34 after 0600 hrs, and then we generated forecasts for the next 2 h (0800–1000 hrs). Thirty minutes later (0830 hrs), the second trial started. Similarly, we only considered the flights that arrive at VIE after 0600 hrs or arrived before 0830 hrs, and we generated forecasts for the time interval between 0830 and 1030 hrs. It has been observed that, for this case study and for the second prediction trial, an increase in prediction accuracy is obtained of 5%, which was due to the increased number of daily arrival flights.

Discussion on NREX Results

To the best of our knowledge, there is no existing NREX prediction function to compare against. However, we could compare different metrics with each other. In Table 6.3, we built a lookup table of both models to compare our NREX predictions against. The results of our case study include a final model that has 12 predictors and is fitted with a maximum tree depth and minimum leaf size set to 11 and 135, respectively (Figure 6.4). The models take into account (1) the layout of the arrival runway 34, (2) the time period and amount of data needed to learn, (3) the computational time needed for a prediction, (4) the instances at which the prediction is performed, and (5) the prediction variables.

From the accuracy metric (see table 6.4) we conclude that for non-procedural exit the Medium aircraft performs best and for procedural the Heavy and Light category performs best. We first focus on the F1 score, which is a measure of a test's accuracy. It considers both the precision p and the recall r of the test to compute the score; p is the number of correct positive results divided by the number of all positive results returned by the classifier, and r is the number of correct positive results divided by the number of all relevant samples (i.e. all samples that should have been identified as positive). In other words, precision is a measure that tells us what proportion of arrival flights that we predicted as procedural, actually took a procedural exit, whereas recall is a measure that tells us what proportion of flights that actually took a procedural exit was diagnosed by the algorithm as procedural. It is clear that recall gives us information about a classifier's performance with respect to false negatives (how many did we miss), while precision gives us information about its performance with respect to false positives (how many did we catch). From Table 6.3 we can conclude that the F1 score performs best for Heavy aircraft, which leads to the conclusion that the procedural heavy model performs very well.

On the other hand, the AUC score measures the positive rates against the false positive rates. The AUC performs best for the medium aircraft. This can be explained by the fact that this model has very few false positives which is an indication of the specificity.

The computational time to learn a model depends on the number of historical flights used to learn a model. Hover the numbers are close to each other meaning that we cannot conclude a best performed model.

6.5 Conclusion

This study demonstrates the use of ML techniques to forecast NREX per flight. We first reviewed the AROT and RET aircraft behaviours, the data sources, precursors and ML outcomes. A model was derived to predict NREX for each flight using the GB method. We then developed an approach to generate classification associated to each NREX flight and the number of aircraft to RWY34 within a time frame of 30 minutes. We also developed an application for VIE to produce these forecasts. In order to facilitate a live trial at VIE, the accuracy of the model has been assessed and improvements were made. To this end, we will conduct a feasibility study where we analyse how our prediction tool can be used by ATCOs in their decision making and planning to ensure resilience, safety and efficiency of airport operations. This is foreseen both within a local sector and within the scope of coordination with other sectors.

There are some advantages associated to using our model. First, the ML technique that we use to build the model is fast, intuitive and interpretable. It can help the airport managers to understand the driving features of the NREX selection for RWY34. Learning our model with *OOBPermutedVarDeltaError* results in a model which is more accurate than when *OOBPermutedVarDeltaError* is not used for new similarly structured data [31]. Second, our model has been built based on 12 predictors, enabling to build new features using domain knowledge of the data. We observed that the GB technique performs best, when a maximum tree depth and minimum leaf size is adopted of 11 and 135, respectively. Third, our model can update the predictions in real-time. The application we developed for VIE airport allows easy extraction of real-time data. The forecasting procedure is effective and the predictions can be generated in a short amount of time. Finally, our model is the first to provide NREX forecasts for RWY34. When an indication is raised at 2NM from the RWY threshold for Medium aircraft types, the ATCO should have confidence that this flight will take runway exit B1, B2 or B4. The NREX forecasts for a flight movement may help ATCO's to make better decisions, to predict whether the flights will experience additional AROT and to anticipate in advance when arriving queuing starts.

6.6 Recommendations

A Real-Time Simulation (RTS) will be executed in a EUROCONTROL Real Time simulator with a primary objective to assess the operational feasibility and acceptability of using the predicted procedural or non-procedural runway exit taken (NREX) in an approach environment. With the RTS, you can conduct preliminary investigations on the impact that new ATM concepts will have on the air traffic controller working environment. The simulation is scheduled for the 23rd and 24th of April 2019.

Airport operational rules, pilot practices, regulations, and standards can vary significantly over a time period of 3 years. Therefore, in future development of the conceived NREX model, the utility of a faster, trend-tracking model will be explored by reviewing the prediction accuracy of the model on a day-to-day basis compared to a static model based on the full 3 years of data. Additionally, pilot practices, ATCO observations, weather events, maintenance down-time and runway/taxiway repair will also be included in our NREX model. An example of a pilot practice is that efficiency is the main reason why flights depart from a specific point, or may exit at a specific point. This includes efficiency in; economic environmental, best practice (1) better departure sequence and (2) waiting time for departure is lower. An example of an ATCO observation is that Lufthansa Medium types have lower AROT compared to other airlines for the same runway and associated landing conditions.

While the model has been developed for a NREX prediction problem, we believe the methodology proposed in this study can be readily applied to other runway processes as well, such as the prediction of Arrival Runway Occupancy Times. This topic will be explored in future research.

Acknowledgement

This work is supported by EUROCONTROL Airport Research; the H2020 project SafeClouds and the aviation industry. The authors would like to thank Austrocontrol for providing access to the aircraft final approach radar and runway data. VIE ATCO Philipp Wächter, runway throughput expert Vincent Treve and head of airport research at EUROCONTROL Bob Graham for their valuable insights. The author also wishes to acknowledge the support and expertise of the technical teams within SafeClouds that made this work possible.

References

[1] B. Ghalebsaz-Jeddi, G.L. Donohue, J.F. Shortle. 2009. A Statistical Analysis of the Aircraft Landing Process. Journal of Industrial and Systems Engineering 3(3): 152-169

[2] J. Roudet, P.- E. Thurat, N. Turcot. 2016. Airport ground-traffic surveillance systems data feed innovative comprehensive analysis. Transportation Research Procedia 14:3741 – 3750

[3] S.H. Goldthorpe. 1997. Sensitivity of runway occupancy Time (ROT) to Various Rollout and Turnoff (ROTO) Factors. NASA Contractor Report 201712, NASA, Hampton, VA, United States

[4] B. J. Kim, A. A. Trani, X. Gu, C. Zhong. 1996. Computer Simulation Model for Airplane Landing-Performance Prediction. Transportation Research Record Journal of the Transportation Research Board 1562(1):53-62

[5] Hobeika, A. G., Dona, E. L., & Nam, A. S. 1988. Optimal Location of High-Speed Runway Exits Using Automated Landing, Rollout, and Turnoff. Transportation Research Record 1147:34-39

[6] Trani, A. A., Hobeika, A. G., Sherali, H., Kim, B. J., Sadam, C. K. 1990. Runway exit designs for capacity improvement demonstrations – Phase 1: Algorithm development. Technical Report NASA-CR-187955, NASA, FAA, Hampton, VA, United States

[7] Cheng, P., Liu, W., & Zou, X. 2014. Airport Surface Trajectory Optimization Considering Runway Exit Selection. 17th International IEEE Conference on Intelligent Transportation Systems (ITSC14)

[8] Zhu, X., Li, N., Sun, Y., Zhang, H., Wang, K., Tsai, S.-B. 2018. A Study on the Strategy for Departure Aircraft Pushback Control from the Perspective of Reducing Carbon Emissions. Energies 2018, 11, 2473

[9] M. Janic. 2017. Analysing and modelling some effects of solutions for matching the airport runway system capacity to demand. Journal of Air Transport Management 65:166-180

[10] R. Prakash, R. Piplani, J. Desai. 2018. An optimal data-splitting algorithm for aircraft scheduling on a single runway to maximize throughput. Transportation Research Part C 95:570-581

[11] J. Guépet, O. Briant, J.-P. Gayon, R. Acuna-Agos. 2017. Integration of aircraft ground movements and runway operations. Transportation Research Part E 104:131–149

[12] H. Balakrishnan, Y. Jung. 2007. A framework for coordinated surface operations planning at Dallas-Forth Worth International airport. In AIAA Guidance, Navigation and Control Conference and Exhibit, Hilton Head, South Carolina, United States

[13] X. Zou, P. Cheng, W.D. Liu, N. Cheng, J.P. Zhang. 2018. A two-stage taxi scheduling strategy at airports with multiple independent runways. Transportation Research Part C 95:165-184

[14] Coetzee, E., Krauskopf and M., Lowenberg, M. *Analysis of Medium-Speed Runway Exit Maneuvers*. Journal of aircraft. 48 (5), p1553-1564.

[15] Kim B.J., Trani A.A., Gu X. and Zhong C., Computer simulation model for airplane landing-performance prediction, Transportation Research Record, 1996, (1562), p53–62.

[16] Barbaresco F., Juge P., Klein M., Canal D., Ricci Y., Schneider J.Y., , Mutuel L., 2013, Wake Vortex Detection, Prediction and Decision Support Tools In SESAR Program, 32th DASC Conference Syracuse.

[17] Blockeel, H. (2016). *Machine learning and inductive inference*. Leuven: ACCO UITGEVERIJ. p100-140.

[18] Herrema, F.F., Curran, R., Visser, H.G., Treve, V., and Desart, B., 2018. Predicting abnormal runway occupancy times and observing related precursors. *Journal of Aerospace Information Systems*, *15*(1), pp. 10–21.

[19] Herrema, F.F. Curran, R., Visser, H.G., Huet, D. and Lacote, R., 2018. Taxi-out time prediction model at Charles de Gaulle airport. *Journal of Aerospace Information Systems, 15*(3), pp.120–130.

[20] P. Chapman (2000). *Step-by-step data mining guide*. https://www.the-modeling-agency.com/crisp-dm.pdf New York: SPSS. P12.

[21] Leys, C., Ley, C., Klein, O., Bernard, P. and Licata, L. (2013). *Detecting Outliers: Do not use standard deviation around the mean, use absolute deviation around the median*. Journal of Experimental Social Psychology. 49 (4), p764-766.

[22] Ji-Hyun, K. (2009). *Estimating classification error rate: Repeated cross-validation, repeated hold-out and bootstrap*. Journal of Computational Statistics & Data Analysis. 53 (11), p3735-3745.

[23] Lagomarsino, D. (2017). *A Tool for Classification and Regression Using Random Forest Methodology: Applications to Landslide Susceptibility Mapping and Soil Thickness Modeling*. Journal of Environmental Modelling & Assessment. 22 (3), p201-214.

[24] Navakovic, J. and Rankov, S. (2011). *Classification Performance Using Principal Component Analysis and Different Value of the Ratio R*. Journal of computers, communications and control. 2 (1), p317-327.

[25] Xue, B., Zhang, M. and Browne, WN. (2013). *Particle Swarm Optimization for Feature Selection in Classification: A Multi-Objective Approach*. IEEE Transactions on Cybernetics. 43 (6), p1656-1671.

[26] Freidman, J.H. *Stochastic Gradient Boosting*. Computational statistics & data analysis 38 (4), pp. 367-378, 2002.

[27] Herrema, F.F., Treve, V., Curran, R. and Visser, H.G., 2016, June. *Evaluation of feasible machine learning techniques for predicting the time to fly and aircraft speed profile on final approach*. In International Conference on Research in Air Transportation (Vol. 8, No. 4, pp. 4-8).

[28] Arizona state university (2013). *Design and analysis of experiments*. 5th ed. Arizona: Thomson Digital. 449.

[29] Hai Ning. (2016). *Overview diagram of Azure Machine Learning Studio capabilities.* Available: https://azure.microsoft.com/en-us/documentation/articles/machine-learning-studio-overview-diagram/. Last accessed 7th Nov 2016.

[30] Breiman, L., J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Boca Raton, FL: CRC Press, 1984.

[31] Bermell-Garcia, P., Verhagen, W.J.C., Astwood, S., Krishnamurthy, K., Johnson, J.L., Ruiz, D., Scott, G., Curran, R. *A framework for management of Knowledge –Based Engineering applications as software services*, *Enabling personalization and codification*. (2012) Advanced Engineering Informatics, 26 (2), pp. 219-230. Cited 22 times

7.0 Validation Runway Utilisation tool

Finally, together with EUROCONTROL a proof of concept and validation exercise was performed, based on the use of feasible ML and BD technologies. The two best performed models; abnormal runway occupancy times from Chapter 5 and missed procedural runway exit (Chapter 6) are validated into the EUROCONTROL ATC RTS tool. Both ASP enablers are validated on their RU requirements, operational needs and operational feasibility. The outcome for the RU requirements is summarized in Section 8.1.2. This ensured the best use possible of the existing safety data in order to enhance decision making at European level.

This chapter was previously published as:

C.C. Morgan, F.F. Herrema, M. Ellejmi & R. Curran. Validation of the Runway Utilisation concept at Vienna airport. 9th SESAR innovation days (SID 2019), Athens.

Abstract
The final RU tool includes a machine learning and big data infrastructure (Section 2.5) that provides the computational power required to ingest and analyse the data managed for Vienna airport. The tactical and strategical predictive RU tool updates ASP patterns and is able to show alerting issues and support decisions. The real-time simulation reported in this document is part of the first RU validation process to investigate how the knowledge gained from machine learning and big data tools can be applied in the operational environment to support the tower runway controllers in their work. The AROT and NREX were selected for the RU support tool in the previous chapters as they allow for *intuitive*, *interpretable*, *quick* and *accurate* decisions through suitable visualisations. Thus, we validated how the real-time RU support tool (AROT and NREX) could be used by ATCOs in their decision-making and planned to ensure safety and efficiency (*fast*, *intuitive* and *interpretable*) of airport operations through suitable *visualisation*.

7.1 RU pre-validation steps

The validation study was conducted in an ATCO Real Time Simulation (RTS) tool inside the EUROCONTROL Experimental Centre (EEC). The EEC in Paris validates any ATCO concept on the runway, during the final approach or in the airspace. They do this using realistic (real-time) simulations to obtain hard data and human ATCO performance conclusions. Many simulations have been executed in the EEC RTS simulator using different airport environments. This led to different deployed runway throughput solutions like: RECAT-EU, TBS and airspace design solutions [1]. Over the last year we have been developing and validating ML support decision solutions using Machine Learning (ML)/ Big Data (BD).
Under a H2020 project named SafeClouds[17] (that enabled the thesis funding), we agreed to validate the predicted AROT and NREX using ML and BD in real-time. This RTS is a V1[18] validation activity to investigate the way in which the knowledge gained from the Runway Utilisation (RU) activities can be applied in the operational environment and to support the tower runway controllers' work. While validating new solutions in the air traffic management domain, three different maturity levels (V1, V2 and V3) [2] need to be cleared before industrialisation and deployment. Maturity assessment analyses come from project activities i.e. concept validation activities to assess the progress of R&D work for a solution through different trial scales. The maturity assessment involves checking whether the set of maturity criteria for a trial maturity phase have been successfully achieved by a SESAR solution being analysed and whether it is supported by a maturity assessment tool (MAT) provided by the SESAR Joint Undertaking[19].

[17] www.safeclouds.eu
[18] V1 identifies the operational and technical solutions for meeting the target performance identified in Section 7.2.1
[19] www.sesar.eu

The EEC RTS [3] enables connecting RU ML python algorithm to the simulation platform that makes real-time predictions. The predictions were made on a local computer based on historical raw data from AUSTROCONTROL. The raw data sets were uploaded and secured on the Amazon S3 bucket [4] (Section 2.5) and has so far covered 100000 flights. The flights include the NREX and AROT along with their corresponding prediction values. Subsequently, the Extraction Transformation and Loading (ETL) of the data was executed [5]. The extraction and transformation (merging of the data) step was done in Spark using key value pairs [6] to merge data sources with similar time stamps. Then, the data was uploaded on Python – the cleaning, feature selection and prediction was executed it. After the ML was applied, the predictions were directly sent to the RTS (I-drive) for validation.

The simulator must read all the arrival flights for each exercise at once, such that all predictions are made in advance. Each exercise counts 43 arrival flights, and there are thus 43 predictions. We tested if the predictions are *fast* enough by measuring the time between the moment the prediction was made and the moment the prediction was visible on the local I-drive from the simulator. After conducting two training sessions on Tuesday (Table 7.5), we concluded that all predictions were made in less than three seconds. As a next step, we plan to measure the time between the prediction and the action from the ATCO. However, we expect that the time from reading the data from the I drive and visualizing the prediction on the HMI to taking an action (by the controller) will not be more than one second.

After performing the pre-computational steps and time measurements we concluded that the validation approach captures 3 of the 5 RU requirements. The platform:

- Enables making predictions with low computational time, triggering actions by the ATCO in less than five seconds.
- Enables showing the prediction value in green (=>80% accuracy) or in red (<80% accuracy).
- Cannot show the precursors that impact an enabler exceeding their threshold (*intuitive*/ *interpretable*). This is possible during the maturity level 2 by the reception of precursors from the tree.
- Enables showing the NREX and AROT prediction value on the HMI when the following criteria are met: 1) *fast* and *accurate* ASP prediction and 2) when an ASP enabler exceeds a threshold (resulting in ASP risk).

7.2 Validation methodology

7.2.1 RTS objectives

This RTS is a V1 validation activity to investigate how the knowledge gained from BD tools and ML activities can be applied in the operational environment and support the tower runway controllers in their work. There were three main objectives of the RTS:

- **Operational Needs**

  To gain feedback from controllers in terms of whether such a controller support tool based on ML and enhanced prediction of RU meets controllers' operational needs.

- **Operational feasibility & acceptability**

  To assess the operational feasibility and acceptability of a controller support tool based on ML and enhanced predictions of RU.

- **Controller Runway Utilisation requirements**

  To assess the requirements of the controllers with regards to controller support tool based on ML RU predictions (e.g. AROT and NREX), for example, in terms of information requirements, timeliness of information, accuracy of predicted information.

7.2.2 RTS scope

Based on the findings from ML activities an initial prototype controller support tool was developed to inform controllers in advance of the predicted AROT and/ or NREX for each aircraft, i.e. the ML RU controller support tool.

The prototype was used to provide controllers with a possible example of how the enhanced predictions of runway utilisation gained from ML could be applied in the operational environment to support their work.

The simulation was conducted using the EUROCONTROL RTS eDEP platform with iTWP and a 3D external view. The RU ML support tool prototype for predicting AROT and NREX was integrated into the EUROCONTROL eDEP integrated Tower Working Position (iTWP). [3]. The simulation was based on the Vienna approach / tower environment using Runway 34 in segregated arrival mode only. Two controllers from the Vienna Tower participated in the simulation.

7.2.3 Solution description

Within this study we are developing an algorithm that would support the tower ATCO during HIRO by making predictions relating to AROT and / or NREX based on historical observations of runway traffic.

The predictions are produced at 2NM upstream from the runway threshold. This enables tactical operational tools to be supplied with real-time data and provide **additional information or warning to the tower controllers** if appropriate, based on these predictions.

The RU ML support tool provides an indication on the tower runway controller CWP HMI of the likelihood that that the AROT of the landing aircraft will adhere to HIRO rules. A red indication will be provided when the prediction accuracy of NREX is lower than 80% and/ or an AROT with a Mean Squared Error (MSE) higher than 6 seconds. A green indication will be provided when the prediction accuracy is higher or equal to 80% or lower than 6 seconds for the AROT. The prediction will be produced at 2NM upstream from the runway threshold, enabling tactical operations tools to be fed and ATCOs to be warned about the two predictions identified above.

7.3 RTS conduct

7.3.1 Environment

The operational environment used for the RTS was based on the Vienna environment. Vienna airport has two runways (see Figure 7.1). This validation exercise only concerned a single approach HIRO environment for RWY34 (the most common pattern of operations), departures will not be simulated.

The RWY 34 exits are B9/B7 for the light aircraft, B5/B7 for the medium aircraft and B5/B4 for the heavy and super heavy aircraft. Within this exercise, it was also possible that a Light or Medium aircraft type vacate RWY34 via exit B1, B2, B4 or B5 (only for Light) or when a Heavy aircraft vacates at B2 or B1.

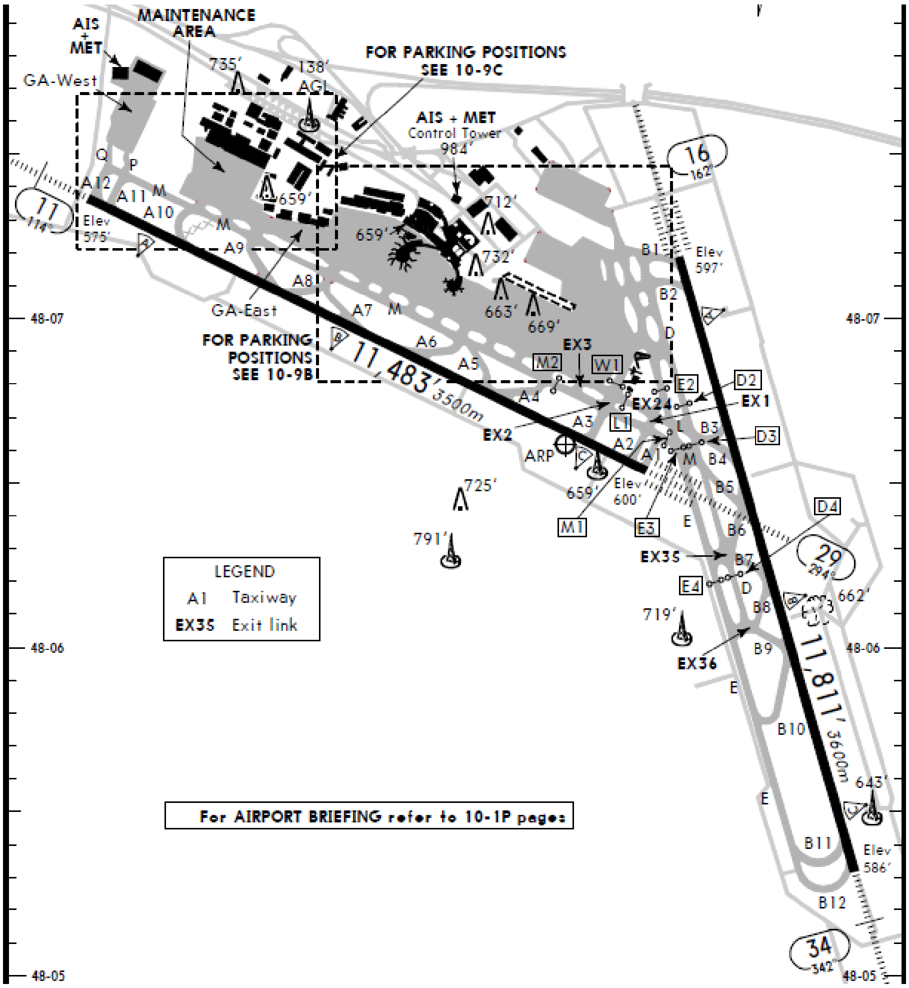


Figure 7.1: Vienna Runway layout

7.3.2 Traffic

Two traffic samples were used in the SafeClouds RTS. The traffic samples were taken from the RTS performed within SESAR 2020 PJ02-O3 (namely traffic samples W2A1 and W2A2) and consist of arriving aircraft only.

The traffic samples were based on real flight data taken from the morning traffic in Vienna (August 2015) which have been adapted so that they have a mix that corresponds to an extrapolation of what the traffic is currently predicted to be at Vienna Airport in 2020.

The traffic sample has included the aircraft types, call sign and traffic mix comparable to Vienna airport traffic. Table 7.1 presents the distribution of aircraft type categories within the sample.

Table 7.1: Traffic distribution

ICAO WTC	RECAT LOWW WTC	Arr
A380-800	**A380-800**	1
Heavy	**Heavy** (except B76X/B75X/A310)	3
	B76X/B75X/A310	1
Medium	**A320/B737NG**	18
	Medium (except A320/B737NG)	16
Light	**Light**	2

7.3.3 Wind Profile Modelling

The following low wind profile was used:

Table 7.2: Wind characteristics applied in the Safe Clouds RTS.

LEVEL Feet MSL	WIND HEADING	WIND SPEED Knots	WIND SPEED m/s	Crosswind Component Knots	Headwind Component Knots
000-4000	320	0	0	0	0
000-4000	320	20	10,28	6,8	18,8

The same wind was applied in all runs. The wind remained constant throughout each exercise, so there was no wind variation during an exercise.

7.3.4 Speed Profile Modelling

True air speed (TAS) profiles on approach have been analysed to create modelled profiles, which were split by aircraft type and wind band. The simulation platform used speed profiles (from Chapter 3.0), which were split by aircraft type, wind band to simulate variability.

The model used is outlined in the figure below and is described using four parameters:

- The glide speed V_{GLIDE} maintained down to the deceleration fix;
- The deceleration fix, defined as a certain distance from the threshold;
- The stabilisation fix, defined as a certain distance from the threshold;

- The final approach speed V_{APP} reached and maintained by the aircraft from the stabilisation fix to touchdown.

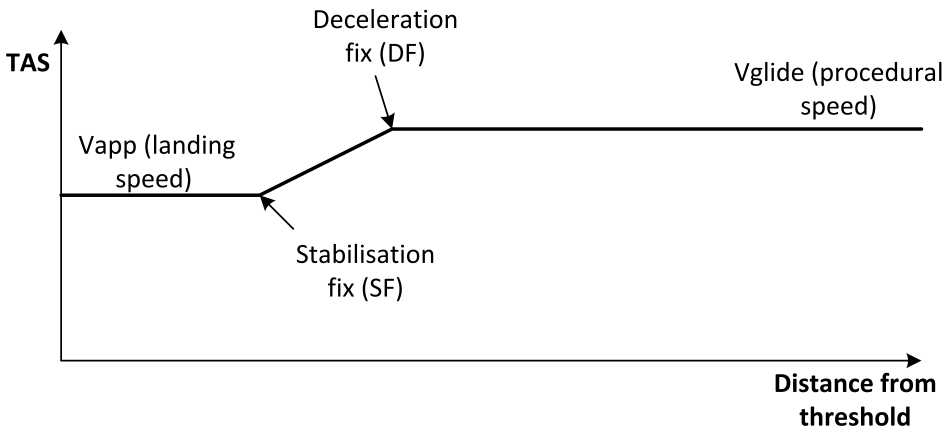


Figure 7.2: Aircraft speed profile model for arrivals

7.3.5 Separation Scheme

The wake turbulence separation scheme was the current wake turbulence separation scheme used in the Vienna approach and tower environment, i.e. Distance Based ICAO wake turbulence separation scheme without any support tool under VMC. In VMC in Vienna visual separations are often applied therefore, MRS pairs may be delivered under 2.5NM under visual separation rules.

7.3.6 Arrival Runway Occupancy Time

The average predicted AROT is highlighted in Table 7.3

Table 7.3: Average AROT in the SafeClouds RTS.

	Traffic sample 1	Traffic sample 2
Super Heavy	76 seconds	79 seconds
Heavy	65 seconds	68 seconds
Medium	55 seconds	56 seconds
Light	49 seconds	50 seconds

In each exercise, a number of non-procedural exits were simulated. A non-procedural exit refers to when a flight vacates the runway at an exit further along than the Aeronautical Information Publication (AIP) intended one. The number of non-procedural exits that occurred in each measured exercise run was about 15% of the total landing aircraft. will be implemented in the RTS.

Table 7.4 shows the procedural exit and non-procedural for Heavy, Medium and Light aircraft as will be implemented in the RTS.

Table 7.4: Procedural AIP exit at RWY34 for VIE airport with associated non-procedural exit.

Aircraft ICAO category	Procedural exit	Non-procedural exit
Heavy	B4, B5	B1, B2
Medium	B5, B7	B1, B2, B4
Light	B7, B9	B1, B2, B4, B5

7.3.7 Tower simulation platform

The EUROCONTROL eDEP Integrated Tower Controller Working Position including the 3D external view was used to simulate the tower runway position for RWY34 at Vienna in the Safe Clouds V1 RTS. The tower runway controller worked only arrivals in segregated mode runway operations. Controllers were required to input all aircraft clearances /instructions and sequence changes directly into the system via the ITWP HMI- The tower runway position is also manned by one pseudo-pilot. The ground position is fully automated.

7.3.8 Example of the RU support tool for predicting NREX

The RU ML support tool for predicting AROT / NREX was integrated into the EUROCONTROL eDEP integrated Tower Working Position (iTWP).

The RU ML support tool for predicting NREX indicate to the tower controller on the tower CWP HMI, whether not, each landing aircraft has a less than 80% prediction of taking the assigned runway exit.

When the mouse is moved over the runway exit information either in the track label or in the EFS then a vertical arrow will be displayed next to the runway exit.

If an aircraft has, a less than 80% prediction of taking the assigned/procedural runway exit as defined in the AIP the arrow displayed will be red. If the prediction of taking the assigned/procedural runway exit as defined in the AIP is more than 80% then a green arrow will be displayed (as will the RWY and assigned RWY exit presented in the aircraft label).

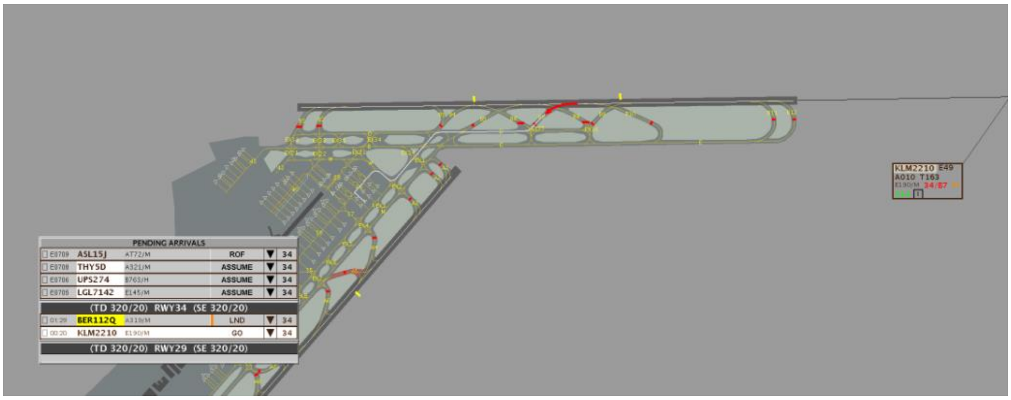


Figure 7.3: iTWP interface showing red arrow at RWY exit (B7) indicating the prediction that the approaching aircraft will take the procedural RWY exit for that aircraft (B7) is less 80%.

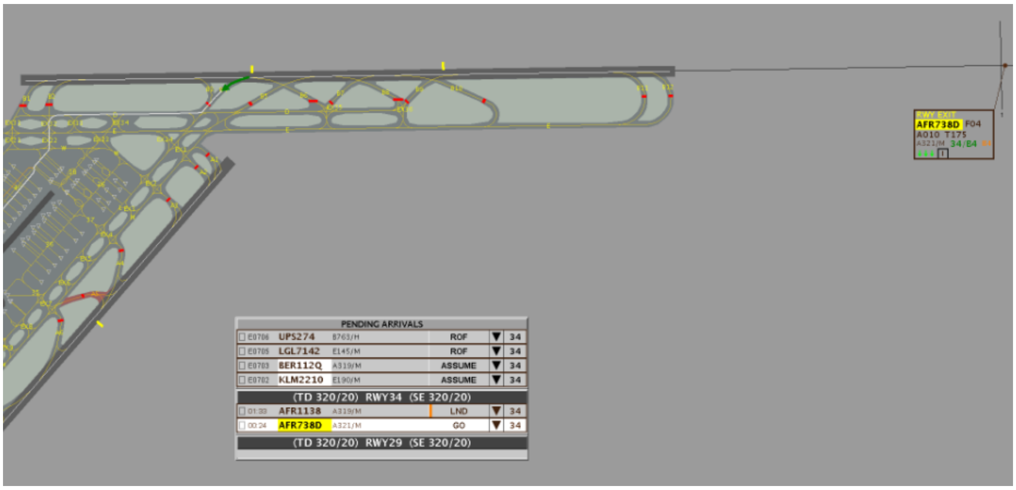


Figure 7.4: iTWP interface showing green arrow at RWY exit indicating the prediction that the approaching aircraft will take the procedural RWY exit for that aircraft (B4) is above 80%.

7.3.9 Conduct of RTS

The RTS took place over a 4-day period from 23rd to 26th May 2019 (Table 7.5).

Two Vienna tower controllers participated in the RTS. Both controllers were already familiar with the eDEP iTWP simulation platform, as they had participated in several previous RTS conducted in SESAR1 WP6.8.1 and SESAR2020 PJ02, using the eDEP iTWP.

The controllers were initially briefed on the objectives of the RTS, the ML concept for RU and the RU support tool.

Once fully briefed the controllers were each given a training exercise to re-familiarise themselves with the simulation environment, the iTWP HMI and also familiarise them with the RU ML controller support tool.

Following the training exercises, each controller was asked to work the tower runway positions, as they would do in real life operations under VMC with the initial prototype RU ML support tool.

After each exercise, debriefs were conducted with the controllers to gain their feedback regarding the RU ML support tool prototype.

Based on the controllers' feedback from the initial exercises, two additional versions of the RU ML support tool were developed and assessed by the controllers during the simulation.

Table 7.5: RTS schedule

Time	Tuesday	Wednesday	Thursday	Friday
09:15-09:30		Briefing	Briefing	Briefing
09:30-10:30		Exercise 1 - Reference ATCO 1 Traffic TW2A1	Exercise 5 – Solution RWY threshold ATCO2 Traffic TW2A1	Exercise 7 – Solution 2NM with Alert ATCO2 Traffic TW2A1
10:30-11:00		Debrief	Debrief	Debrief
11:15-12:00		Exercise 2 – Reference ATCO2 Traffic TW2A2	Exercise 6 – Solution RWY threshold ATCO1 Traffic TW2A2	Exercise 8 – Solution 2NM with Alert ATCO1 Traffic TW2A
12:00-12:30		Debrief	Debrief	Debrief
12:30-13:30	Briefing on concept, HMI & RTS objectives	Lunch	Lunch	Lunch
13:30-14:30	Training – Solution ATCO1 Traffic TW2A2	Exercise 3 – Solution 2NM ATCO1 Traffic TW2A1	Safe clouds concept Use Case discussion	Project meeting

14:30-15:00	Debrief	Debrief		
15:00-16:00	Training – Solution ATCO2 Traffic TW2A1	Exercise 4 – Solution 2NM ATCO2 Traffic TW2A2		End of day debrief
16:00-16:30	End of day debrief	Debrief	End of day debrief	End of RTS

Therefore, over the 4 days of the RTS, each of the controllers worked with three slightly different versions of the RU support tool and provided their feedback. The three different versions of the RU support tool consisted of:

1. The initial prototype **RU support tool** with the NREX and AROT prediction for an aircraft updated when the aircraft was at **2NM from the runway threshold**.
2. The **RU support tool** with the NREX and AROT prediction for an aircraft updated when the aircraft was **at runway threshold** (therefore the information was updated later but had a higher percentage accuracy/reliability than the information updated when the aircraft was at 2NM from the runway threshold (approx. 6% more accurate).
3. The **RU support tool** with the NREX prediction for an aircraft updated when the aircraft was **at 2NM** from the runway threshold **plus an automatic pop-up information alert** displayed on the iTWP HMI when an aircraft was predicted to take a non-procedural runway exit (i.e. with a less than 80% prediction of taking the assigned/procedural runway exit as defined in the AIP) that would increase the AROT to above what was usual for that aircraft type.

The feedback obtained from all the controllers following each exercise was noted and is summarised per exercise in the results section below according to the three objectives defined in Section 7.2.1

7.4 Results from the V1 real time simulation

The results of this V1 validation activity are based solely on controllers feedback based on the three versions of the ML RU controller support tool tested in the RTS.

7.4.1 Operational needs

- The controllers felt that information based on ML regarding runway utilisation could be used to support operations and controllers work by enhancing controllers' situation awareness and hence provide potential safety benefits.

- Whereas, predictive information relating directly to runway exit (NREX) was not considered to be needed by controllers to support their work, predictive information relating to a change in AROT was seen to be very useful and would support the operational needs of the controllers. (However, one of the controllers stated that the ML predicted NREX would be ‘’a nice to have option although it was a bit of a gimmick’)

- The ML predicted information relating to AROT or ideally information regarding the consequences of a change in AROT, especially if there is a potential negative impact on controllers work, was seen to needed from an operational perspective and would support controllers in their work. The tower runway controllers stated that they only need to be alerted about unusual behaviour or if there is a potential situation where controller may have to do something, for example if the AROT of the preceding aircraft is greater than the time to touch down of the follower and there could be a potential loss of separation, then an information alert should be provided to controllers.

- Controller’s reported that they would not act directly on the predictions (e.g. give a go-around to a follower aircraft) if reliability was not 100%, but they would monitor the situation more closely and wait to see how the situation unfolded.

- Although, controllers did report that they would use such AROT predictions to try and prevent any predicted AROT increases from occurring, for example by stating the procedural runway exit the aircraft is required to take when communicating with the pilot or requesting pilots to expedite the runway.

- Pilots may also like to have the information given to them by the controllers for example, one controller stated that pilots would like to know in advance if a preceding aircraft is staying longer on the runway and exiting further down the runway. (This feedback from the controllers is based on a real life incident in operations as a pilot was complaining that the controllers gave such information too late when the follower was on the final approach and 1 or 2NM from the runway threshold).

7.4.2 Operational feasibility and acceptability of the ML RU controller support tool

- The ML RU support tool was considered operationally feasible and acceptable to the Vienna controllers that took part in the V1 simulation.

- The predicted information regarding AROT if automatically presented to the controllers as an information alert was seen as being ‘’very valuable’’ as it would ‘’draw the controllers attention to a potential situation’’ that may impact operations. This would help to enhance the tower runway controllers’ situation awareness relating to potential runway incursions, and therefore have potential safety benefits.

- However, the controllers stated that as the predicted information was not 100% reliable they would use the information presented to check and monitor a situation more closely. Controllers reported that they may use the information to issue instructions such as reminding the pilot to take the procedural runway exit or expedite the runway, or wait a little longer before giving a landing clearance in order to help mitigate any potential increase in AROT. However, the controllers said they would not act on the predictive information in terms of issuing a go-around (for an arrival aircraft following the concerned arrival) or giving a line-up clearance (to a departing aircraft following the concerned arrival).

- The controllers reported that the additional information based on ML predictions would not have any impact on their workload.

- The controllers did not feel the RU information based on ML predictions could be used as a means to increase runway throughput capacity.

- Although ML predictions could potentially optimise runway operations under certain circumstances. For example, in mixed mode runway operations if the leading arrival aircraft was taking an earlier exit the runway would be free earlier, therefore AROT would be reduced and perhaps a departure would be possible.

- When questioned, if the first aircraft was predicted to exit the runway early and the second aircraft was at 4.5NM would the controllers tell the second aircraft to maintain speed and reduce as late as possible (''keep speed as long as possible'') to allow for additional space for a departure in between the second and third aircraft on final. The controllers responded that they would not do this based on predictive information. As stated previously, the controllers would use the alert as information only and would wait and observe the aircraft to see if it vacates earlier or not as there could be potential safety impact of over-relying and acting on the predictive information based on ML in such a situation.

- Both controllers proposed that the predicted AROT determined by ML could be used to further optimise runway throughput operations by integrating the predicted AROT into the ORD tool (AO-0328) developed within SESAR 2020 PJ02-01. In such an 'advanced' solution the AROT determined by the ML could be fed into the ORD tool to update the FTD chevron when lead aircraft is at 1.5NM - 2NM from the runway threshold. In this way, if AROT was the constraining factor between two arriving aircraft on the final approach any changes to the AROT based on ML prediction could be directly displayed to the controller via the FTD and the spacing between the aircraft pair optimised for the AROT constraint.

- The level of reliability/accuracy of the predicted information by ML that is acceptable to controllers needs to be determined to ensure that controllers can build adequate trust in the alert/ controller support tool and there are not too many false alerts.

7.4.3 Controller information requirements for a ML RU support tool

- The computational time of the AROT and NREX was less than five seconds when predicting during real-time operations at 2NM, making, it according to the controllers, suitable for tactical real-time operational use.
- Based on the interview with the ATCOs (Section 1.1), they would prefer an indication only when the prediction accuracy of a non-procedural runway exit utilised is higher than 80% or when the abnormal AROT has an RMSE of less than five seconds.
- The important information for the tower controllers is not the predicted RWY exit (NREX) but the AROT. Therefore, the controllers do not need to know the predicted RWY exit but the predicted AROT or ideally the consequences of a change in AROT, especially if that consequence could have a negative impact on operations.
- Controllers require an automatic pop-up information alert showing that there may be an issue. Controllers do not want to ‘’seek’’ for the information as implanted in the initial prototype (i.e. place mouse on the aircraft label or EFS to find the information); this is cumbersome and may lead to controllers missing the predicted information updates that could impact operations: One of the controllers stated that ‘’constantly checking the NREX by hovering the mouse on the aircraft label and then looking at the arrows displayed on the runway in the ASMGCS display took their attention awareness from checking what was happening the air on the final approach’’.
- Controllers do not need to have an information alert presented if there is no potential negative impact on operations, for example, if the AROT is predicted to be less than expected as an aircraft takes an earlier non-procedural exit, an alert would not be needed.
- Therefore, an information alerts should be displayed only if there is a potentially negative situation predicted: For example with consecutive arrivals, if the time to threshold of the follower aircraft is smaller than the predicted AROT with a buffer of the lead aircraft (exact buffer required to be decided in a safety assessment TBD) or, in mixed mode operations if there is not enough room for the planned consectuve departure.
- If an alert is displayed the concerned aircraft (lead and follower) will need to be highlighted to ensure the controllers react to the correct aircraft.
- Updated predictions regarding runway exit (NREX) are considered to be a ‘’nice to have’’ but not essential. The format the arrow indicating the runway exit on the HMI as assessed in the RTS are OK and easy to interpret. Therefore NREX predictions could be a selectable option for controllers.

- Controllers do not need to know the level of reliability/accuracy of the predicted information presented on the CWP HMI. (Therefore, the colour of the arrow indicating the accuracy of the predicted information, as implemented in the RTS, is not needed).

- The information should only be displayed when the reliability of the prediction is above a defined value (e.g. 80% as defined in the RTS was seen as sufficient but the exact value of reliability that is acceptable to controllers needs to be determined to ensure that controllers can build adequate trust in the alert/ controller support tool and there are not too many false alerts).

- If implemented the alert should take into account whether there is a follower or not (arrival or departure), as alert would not needed if there is no follower close to the lead that will have a prolonged AROT.

- Controllers reported that update to information based on ML predictions is required at latest when the follower aircraft is at 4NM, (i.e. lead aircraft at 1.5Nm to 2NM from the runway threshold); this gives the controller time to react if necessary on the follower aircraft under both segregated and mixed mode runway operations. At 1-2NM you can also instruct the lead aircraft to expedite the runway and remind them to take the procedural exit, whereas, if you get this updated prediction on the lead aircraft when it is at the runway threshold it is too late. In the RTS when the NREX was updated with predicted information at the runway threshold, this was considered to be too late for the controllers to react – both on the follower and lead aircraft. However, the NREX updated prediction when the lead aircraft was at at 2NM was considered OK even if the reliability of the prediction was slightly less (approx. 6% less in terms of reliability).

- In mixed mode runway operations the tower runway controllers need a tool that provides the sequence. It was suggested that the updated AROT prediction should be incorporated into the sequence tool and help controller determine whether or not there is a potential problem for the follower aircraft (arrival or departure) or a potential benefit (in the case of a departures following an arrival). This tool would be displayed in addition to the information alert.

- The controllers also suggested that the information alert should disappear on acknowledgement by the controller. If the potential separation infringement continues and runway incursion is likely, the RIMCAS can then be displayed.

- Both controllers proposed that an ‘advanced’ solution could be developed where the predicted AROT is integrated into the ORD tool (AO-0328) developed within SESAR 2020 PJ02-01. As mentioned previously in such an ‘advanced’ solution, the AROT determined by the ML could be fed into the ORD tool to update the FTD chevron when lead aircraft is at 1.5NM - 2NM from the runway threshold.

References

[1] Cappellazzo, Valerio, et al. "Design Principles for a Separation Support Tool Allowing Optimized Runway Delivery." *2018 Aviation Technology, Integration, and Operations Conference*. 2018.

[2] Gomez, Alfredo, et al. "Concurrent system engineering in air traffic management: steering the SESAR program." *Proc Posters Work CSD&M* 2013 (2013): 25-32.

[3] EUROCONTROL. (2007). *Integrated Tower Working Position*. Available at: https://www.eurocontrol.int/publication/integrated-tower-working-position-evaluation-report-session-1-april-2007. Last accessed 28th Sep 2019.

[4] Palankar, Mayur R., et al. "Amazon S3 for science grids: a viable solution?." *Proceedings of the 2008 international workshop on Data-aware distributed computing*. 2008.

[5] Tiwari, Prayag. "Improvement of ETL through integration of query cache and scripting method." *2016 International Conference on Data Science and Engineering (ICDSE)*. IEEE, 2016.

[6] Verma, Ankush, Ashik Hussain Mansuri, and Neelesh Jain. "Big data management processing with Hadoop MapReduce and spark technology: A comparison." *2016 Symposium on Colossal Data Analysis and Networking (CDAN)*. IEEE, 2016.

8.0 Conclusion & Recommendations

This chapter includes three sections. The first reviews the research objectives presented in Chapter 1. Then, the main contributions of the research are summarized. Limitations of the work are stated. Lastly, suggestions for future work and extensions to this work are discussed.

8.1 Conclusion

In this thesis, we have addressed the design of an ATCO Runway Utilisation support tool. The tool captures ASP patterns and fits into a single approach operations environment. Its performance was assessed in terms of the three objectives listed in Section 1.3, and the identified limitations and merits were summarised. The basis for the tool design and the imaging ML techniques was developed in Chapters 1 and 2.

As shown in Chapters 3–6, ASP enablers depend on the impact of the prevailing meteorological and airport conditions on the transport and decay of the wake turbulence. To ensure accurate predictions and decision support, all four ASP enablers, Time to Fly (T2F), abnormal runway occupancy time (AROT), taxi-out time (TXOT) and runway exit utilised (NREX) were assessed with respect to feasible ML techniques, their different features and historical data requirements.

8.1.1 Objectives 1 and 2

For objective 1, we assessed different ML techniques, different features and amount of data required, and for objective 2, we tested if, for each enabler identified, ML can be used to effectively identify and formulate ASP patterns, risks and precursors for tactical (real-time) and strategic (1 hour in advance) decision-making.

We developed various decision trees to extract patterns and precursors. Additionally, interpretable results for tactical and strategic decision-making were obtained when tested on both measured and synthetic data.

The following conclusions can be drawn relating to the four ASP enablers along with their subsequent impact on runway throughput operations and objective statements 1 and 2 as stated in Section 1.3:

(1) Using predicted T2F and TAS allows us to make *accurate* decisions leading to a better characterisation of large spacing or infringements (T2F and TAS – Chapter 3):

Objective 1: Based on our T2F and TAS experimental results, we can conclude that learning multitask regularised regression with the feature selection technique “RreliefF” is promising especially when combined with Principal Component Analyses. The ML techniques Lasso and Multilayer Perceptron (MLP) were revealed be the most feasible and most accurate techniques for predicting the TAS and T2F from 8.5NM till 0.5NM and from 4.5 till 0.5NM. Combining these techniques results in a more robust and accurate ML model which is simply the average of the predictions of the Lasso and MLP. Stable Root Mean Squared Error (RMSE) values were obtained when learning minimum 60 flights per aircraft type, wind band and distance from threshold, resulting in a computational time of 10 seconds, thereby making it difficult to provide the ATCO with operational usable real-time predictions. Ground speed at 10NM along with aircraft type, visibility, head-wind and crosswind information seem to capture substantial variation of the T2F and TAS in the last 10NM. When receiving these driving features at 10NM, the ATCO receives accurate and understandable T2F and TAS predictions which they could use for similar situations (intuitive decisions). The ML techniques are more accurate and more robust to changes and overall improve the accuracy of the data mining approach. We have seen that the standard error decreases with larger sample sizes, as the estimate of the population mean improves. Table 3.5 and Table 3.6 suggest that by learning an ML model with head-wind information, the RMSE is significantly lower than without head-wind information for both RMSE from 8.5–0.5NM and from 4.5–0.5NM. Furthermore, the tactical predictions for the 4.5–0.5NM segment has a lower RMSE (2.9 seconds) compared to RMSE from 8.5–0.5NM (4.8 seconds). The more accurate prediction (2.9 seconds) can be explained by the fact that the 4.5–0.5NM segment includes less abnormal historical data. In other words, it includes more patterns, making it easier for the algorithm to learn from. For the strategic predictions (1 hour in advance), we devised an RMSE of 4.5 seconds for the segment 4.5–0.5NM and 6.9 seconds for the 8.5–0.5NM segment.

Objective 2: Considering the most important prediction variable – ground speed at 10NM – might lead to certain operational issues. To be able to predict the T2F in real-time, an ATCO has to wait till the aircraft is at 10NM. The T2F for an ATCO is beneficial to calculate the compression on final approach using, for example, the TBS concept. The dynamic TBS for the follower aircraft needs to be known before 10NM. Therefore, it is suggested to predict the GS at 10NM of the previous aircraft (based on historical flight information of that time period). Moreover, the computational time (10 seconds) might be too large for real-time operations. We conclude that our hypothesis is not true for the ASP enabler T2F and TAS. Tactical predictions should be produced faster and tactical and strategical decisions in this case should be made before 10NM.

(2) Using abnormal taxi-out times (TXOT – Chapter 4) allows us to make *accurate* decisions:

Objective 1: The RMSE metric was chosen as the most important performance indicator, which gives, for the applied regression tree method, on any given day, an average error of 1.6 min. For predictions 1 hour in advance (strategical decisions) a RMSE of 2.5 min was obtained. Stable RMSE values are obtained when learning is performed using a minimum of 70,000 taxi-out flights, resulting in a computational time of 80 seconds. It was observed that the Regression Tree ML technique performed best; for this regression technique, a maximum tree depth and minimum leaf size was adopted of 6 and 4000, respectively. Regression trees are very interpretable, as long as they are short. The regression tree in Figure 4.7 has 61 terminal nodes making it easy to extract certain patterns or events and explain them using 'what if'-scenarios (as defined by the tree). A tree with a depth of 6 requires a maximum of 6 features and split points to create the explanation for the prediction of an individual pattern. There were some advantages associated to using the current model. First, the ML technique that was used to build the model was intuitive and interpretable. It could help the airport managers to understand the driving features of the TXOTs per runway–stand. The precursors help the managers to have a grasp on accurate TXOT without the need for reasoning (intuitive decisions). The most important features are: "unimpeded TXOT," congestion level, "Saturation level, number of departures in the last 20 min," "de-icing stand," "month," "time," "departure stand," "QFU," and "AOBT." Second, the model has been built based on a large historical dataset (500.000 flight records). More than 40 variables were available for selection as predictors. These variables also made it possible to build new features using domain knowledge of the data. Third, the model was able to update the predictions in real time. The application developed for Charles de Gaulle (CDG) Airport allowed easy extraction of real-time data. The real-time forecasting procedure was effective, and the model was the first to provide TXOT forecasts for each flight to a specific runway.

Objective 2: From our prototype (Section 4.5) we can conclude that ML is feasible for extracting precursors and patterns that support the controller on tactical and strategical decision-making. Risks were mainly observed when the unimpeded time is larger than 22 minutes and the congestion level is larger than 32 movements per hour. The downside of our model is that we don't have access to a RTS that could validate this ASP enabler in a CDG environment. Furthermore, the computational time (80 seconds) is too large for testing the model in a real-time operational environment.

(3) Using abnormal Arrival Runway Occupancy Times (AROT – Chapter 5) allows us to make *intuitive*, *interpretable, visual*, *fast* and *accurate* decisions:

Objective 1: A predictive model for AROT flights was developed by combining the Neural Network, MLP, lasso, and regression tree techniques. An RMSE of 4.3 seconds was obtained for tactical predictions and 6.9 seconds for strategic predictions. Subsequently, an approach was developed to generate distributions of each AROT flight and the number of landings for a specific runway (08R) within a time frame of 30 minutes. An application for CDG and Vienna (VIE) was also developed to produce these forecasts. The regression tree indicated that by knowing the top 10 features (Figure 5.5) in advance, a good prediction can be made of the abnormal and normal AROT for which each abnormal AROT flight will fall into one of the 17 precursor categories shown in Table 5.2. Using seventeen terminal nodes makes it easy to extract interpretable abnormal AROT patterns and explain them using 'what if'-scenarios (Figure 5.10). Furthermore, the regression technique performs best for finding associated precursors, for which the Classification and Regression Tree (CART) technique was used to fit a maximum tree depth and minimum leaf size of 6 and 16, respectively. Therefore, our model has several advantages. First, the ML technique used to build the model is quick (computational time of 5 seconds), intuitive and interpretable. It can help airport managers understand the driving features of the abnormal AROT per runway during similar situations without the need for reasoning (intuitive). Second, our model has been built based on a sizeable historical data set of 78,321 CDG and 500,000 VIE flights for which 22 variables are available for selection as predictors. These variables also enable one to build new features using domain knowledge of the data. Third, our model can update the predictions in real time.

Objective 2: Thus, it can be concluded that we can use CART to extract abnormal AROT patterns, risks and precursors for tactical and strategic decision-making. Therefore, the AROT conclusions for the RU requirements, operational needs and operational feasibility are addressed in Section 8.1.2.

(4) Using the non-procedural runway exit used (NREX – Chapter 6) allows us to make *intuitive*, *interpretable*, *visual*, *quick* and *accurate* decisions:

Objective 1: A model was derived to predict NREX for each flight using the Gradient Boosting as the most feasible method. An accuracy level of 79% was obtained (tactical decisions) from the confusion metric using 45,679 arrival flights. For strategic predictions, we obtained an accuracy level of 71%. We subsequently developed an approach to generate classification associated with each NREX flight and the number of aircraft for RWY34 within a time frame of 30 minutes. Gradient Boosting trees are easily interpretable as long as they are short. The NREX tree for medium ICAO category had 28 terminal nodes for which the outcomes were classified to Medium_Procedural and Medium_Non-Procedural, making it interpretable for the ATCO. We can interpret the most important predictors as the major factors that play key roles in influencing NREX selection. These precursors are throughput, SODAR velocity and direction, WMA wind speed, ground speed at 5NM, height at 2NM and 5NM, cloud, visibility and ground speed at 2NM. We also developed an application for VIE to produce and visualize these forecasts in real-time with a computational time of 5 seconds. The results obtained indicated that our VIE model performed 5% better by comparing the NREX prediction results with the Orly airport.

Objective 2: Thus, it can be concluded that we can use Gradient Boosting to extract NREX patterns, risks and precursors for tactical and strategic decision-making. Risks and the most important precursors were identified for cases where the throughput is lower than 28 landings per hour, the Cloud layer is lesser than 8750m, the Groundspeed at 2NM is higher than 147kts, WMAWindSpeed is lower than 29kts and the Groundspeed at 5NM is higher than 155kts. These precursors could be used during similar situations allowing the ATCO to anticipate a non-procedural exit (intuitive). NREX operational needs and operational feasibility conclusions are presented in Section 8.1.2.

8.1.2 Objective 3

AROT and NREX were selected as they allow us to make *intuitive*, *interpretable*, *fast* and *accurate* decisions through suitable *visualisation*. Therefore, we conducted an operational needs and operational feasibility study (objective 3) in which we analysed how our real-time RU support tool (AROT and NREX) can be used by ATCOs in their decision-making and planning to ensure safety and efficiency (fast, intuitive and interpretable) of airport operations through suitable visualisation (Chapter 7). The feasibility study and evaluating the RU requirements (Section 1.2.1) was conducted in an ATC RTS tool. The conclusions and recommendations from the RTS (Chapter 7) are summarised below:

Fast and *accurate* predictions:

- The computational time of the AROT and NREX was less than five seconds when predicting during real-time operations at 2NM, making, it according to the controllers, suitable for tactical real-time operational use.

- Based on the interview with the ATCOs (Section 1.1), they would prefer an indication only when the prediction accuracy of a non-procedural runway exit utilised is higher than 80% or when the abnormal AROT has an MSE of less than six seconds.

Intuitive and *interpretable* predictions:

- The ML RU controller support tool for AROT was considered operationally feasible and acceptable by the Vienna controllers that participated in the simulation. The predicted NREX information based on ML was also considered operationally feasible and acceptable by one of the Vienna controllers but was seen as something that would be a 'nice to have' option. The operational needs and high-level requirements for such a tool in operations are further detailed in Section 7.3.

- The controllers concluded that certain predicted information based on ML, such as AROT, could be used to support operations and controllers' work by enhancing controllers' situation awareness and, thereby, providing potential safety or runway throughput benefits. The RU tool enabled a runway throughput increases for exercise 5 and 6 (Table 7.5) with one landing per hour.

- Although the NREX and AROT predictions are accurate (above 80% accuracy and MSE < 6 seconds), the controllers reported that they would not act directly on the predictions (e.g. give a go-around to a follower aircraft), but would monitor the situation more closely and would wait to see how the situation unfolded.

- At this stage, the RTS was unable to provide the real-time tree structure for both AROT and NREX. Therefore, from the validation effort we cannot derive any conclusions regarding interpretable decisions. However, the controllers appreciate seeing the tree in a subsequent simulation and they already provided feedback regarding the trees presented in Figure 5.10 and Table 6.5.

- Controllers believe that the tree is ideal for capturing interactions between features in the data. The interpretation is arguably quite simple. The data ends up in distinct groups that are often easier to understand than points on a multi-dimensional hyperplane, as in linear regression.

Visualisation:

- The Vienna controllers that participated in the simulation concluded that the NREX information was clearly visualised at 2NM. The visualisations were performed in terms of a red or green indicator depicting less or more than 80% accuracy in terms of prediction. As a subsequent step, controllers indicate that they would only like to see precursors when a non-procedural exit is expected (with a green indicator) for a Light, Medium or Heavy aircraft type. This was accepted by the controllers during the RTS.

- The controllers concluded that the AROT information in the RTS should be visualised better at 2NM. The visualisations were performed in terms of the MSE of the predicted AROT. As a subsequent step, controllers only wish to see abnormal AROT when the MSE is below 6 seconds. This was accepted by the controllers during the RTS.

- The participating controllers during the validation exercise do not feel the need to have the driving features displayed on the RTS screen for normal AROT and NREX predictions during a subsequent validation exercise. Even though the reasoning is not visualised in this simulation, they do not expect that it would influence their decision-making. However, they would like to see the driving features displayed for abnormal AROT and non-procedural exit utilised predictions.

Therefore, based on the findings from the validation activity, we can conclude that the validation has been completed as the RU support tool was reported to meet controllers' operational needs and provide certain safety benefits. The impact of an ML RU controller support tool on controllers' work and runway operations needs to be further investigated in follow-on validation activities. Potential benefits and impacts relating to the ML RU controller support tool that need to be investigated in more detail in the upcoming validation activities are outlined in Section 8.3.

Finally, the ATCOs concluded that the RTS was successful in predicting both the AROT and NREX. They observed improved operations in certain weather conditions, such as increased runway throughput and a potential for a greater level of safety. We conclude that the result of the present research is a new RU support tool that enables to provide unique interpretable and intuitive information out of AROT and NREX patterns on final approach and the runway. The Gradient Boosting technique proves ideal for the detection of patterns, risk and precursors. When predicting the NREX, 95 decision trees and 12 features were used for this technique. Consequently, tactical and/or strategic decisions can be supported using this approach.

A common conclusion for the AROT and NREX enablers is that the most important features were extracted between 2 and 5NM using decision trees such as Gradient Boosting. Generally, these precursors capture patterns that strongly influence the predictability of the ASP enablers. The 10 and 5NM range includes the abnormal historical data which is hard to learn for the decision trees.

8.2 Limitations

The RTS is foreseen to be applied only within a given local sector and not within the scope of coordination between sectors. This topic will be explored in future research and a next validation exercise. Moreover, the RTS RU tool covers only the ASP enablers, AROT and NREX. The ASP enablers T2F and TXOT will be addressed in a different RTS exercise.

Additionally, pilot practices, ATCO observations, weather events, maintenance down-time and runway/taxiway repair should be included in our RU model. An example of a pilot practice is that efficiency is the primary reason why flights depart from a specific point, or might exit at a specific point. This includes efficiency in economic environmental, best practice (1) better departure sequence and (2) lower waiting time for departure. An example of an ATCO observation is that Lufthansa Medium types have lower AROT compared to other airlines.

Airport operational rules, pilot practices, regulations and standards can vary significantly over three year time span. Therefore, in future development of the conceived RU model, the utility of a quicker, trend-tracking model will need to be explored by reviewing the prediction accuracy of the model on a day-to-day basis compared to a static model based on the entire 3 years of data.

8.3 Recommendations for further development

- Three potential solutions were identified for a RU support tool for AROT and NREX predictions and proposed for further investigation:
  - A **simple solution** for segregated runway modes only: This solution would comprise an automatic pop-up information alert when there might be a potential issue e.g. the AROT of the preceding aircraft is greater than the average due to, for instance, a non-procedural runway exit further up down the runway.
  - An **intermediate solution** for mixed mode runway operations. This solution would comprise an automatic pop-up information alert as defined in the simple solution above **plus** the predicted AROT. Mixed mode runway operations would also require a sequence list of the arrivals and departures. The aircraft sequence of arrivals and departures with the ML predicted AROT could be presented in the EFS or AMAN-DMAN tool. Ideally, the sequence list with the ML predicted AROT would be implemented as a decision aid (as done in SESAR 2020 PJ02-01 AO-) which could inform the controllers whether there is enough space between two arriving aircrafts to allow for a departure.
  - An **advanced solution**. The advanced solution would comprise the predicted AROT determined by the ML being integrated into the ORD tool developed within SESAR 2020 PJ02-01 (AO-028). In an advanced solution

with an ORD tool as developed in SESAR 2020 PJ02-01 input, the AROT determined by the ML into the FTD chevron when lead a/c is at 1.5NM–2NM from the runway threshold.

- More detailed information requirements for a RU support tool are identified in Section 7.4 and should be taken on-board in the development of future ML RU prototypes. These requirements will be further validated and refined in the upcoming validation activities.

While the model has been developed for an ASP prediction problem, we believe the methodology proposed in this study can be readily applied to other runway processes as well, such as the prediction of Departure Runway Occupancy Times. This topic will be explored in future research.

About the author

Floris Herrema is a data scientist at the airport unit in EUROCONTROL. His work is focused on developing and exploiting machine learning models to support Total Airport Management concepts and Airport Operations decision support. Floris joined the EUROCONTROL Wake Team in 2014 and undertook analytical tasks related to re-categorising wake vortex separation minima. He received the SESAR young scientist award for his MSc thesis in 2015. Over the last 4 years he has worked on exploiting machine learning to predict airport runway utilisation, as a PhD candidate.

Motivation in Machine Learning and Big Data

After successfully graduating with an MSc from TU Delft in association with EUROCONTROL, I concluded that my motivation and expertise lies in analysing quantitative data sets by accounting for intelligence and judgement. EUROCONTROL offered a unique opportunity to work with huge quantitative data sets and find a new way of thinking and decision-making. I decided to work on this project since data are becoming the new raw material for businesses. The rapid evolution and adoption of big data analyses by industries have improved the efficiency of big data analysis methods for finding efficient operations. However, converting data to information is still a challenge.
My goal during this study was to find and create algorithms that glean the most important information from these big data sets and make it understandable which would allowing more efficient operations. According to me, information is the most valuable commodity. Therefore, I used the following steps ('big five') while analysing big data sets:

- Finding and analysing relevant big data sources
- Finding and utilising algorithms and intelligence
- Judgement
- Decision making which makes operations efficient and lowers costs
- Implementation

Getting information out of big data is like finding an oyster with a pearl. This study supported my journey of becoming a big data analyses and machine learning expert.

Associated publications

The following fourteen papers, authored by myself and my supervisors, have been published as result of the work in this thesis.

- F.F. Herrema, R. Curran and V. Treve. Typical additional spacing-buffer to apply at 4DME for delivering separation minima. 34th Digital Avionics Systems Conference, September 2015.
- G. van Baren, F.F. Herrema and V. Treve. Predicting time to fly on final approach for optimized delivery of separation. Integrated Communications Navigation and Surveillance Conference (ICNS 2016), Washington.
- F.F. Herrema, R. Curran, H.G. Visser and V. Treve. Evaluation of feasible machine learning techniques for predicting the time to fly and aircraft speed profile on final approach. International Conference for Research in Air Transportation (Vol. 8, No. 4, pp. 4-8). June 2016.
- F.F. Herrema, R. Curran, H.G. 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 (2018), pp. 120-130.
- F.F. Herrema, R. Curran, H.G. Visser, V. Treve and B. Desart. Predicting Abnormal Runway Occupancy Times and Observing Related Precursors, Journal of Aerospace Information Systems, Vol. 15, No. 1 (2018), pp. 10-21.
- F.F. Herrema, R. Curran, S. Hartjes, M. Ellejmi, S. Bancroft and M. Schultz, 2019. A machine learning model to predict runway exit at Vienna airport. Transportation Research Part E: Logistics and Transportation Review, 131, pp.329-342.
- G. Nikolovski, F.F. Herrema and V. Treve. Local TBS delay reduction effect on global network operations. 8th International Conference on Research and in Air Transportation (ICRAT 2018), Barcelona.
- S. Belkoura, F.F. Herrema and P. Wachter. A Boosted Tree framework for Runway Occupancy and exit prediction. 8th SESAR innovation days (SID 2018), Salzburg.
- F.F. Herrema and V. Treve. A Real time Runway exit prediction model using LIDAR and ADS-B data, Accepted for the ELSEVIER Special Issue of Transportation Research Part E: Logistics and Transportation. 01-06-2020.
- E. de Mol, M.A. Mitici, B. Desart and F.F. Herrema. Estimating the flight departure duration at a single airport. 10th EUROSIM2019 congress, 02-07-2019, La Rioja.
- C.C. Chalon, F.F. Herrema, M. Ellejmi & R. Curran. Validation of the Runway Utilisation concept at Vienna airport. 9th SESAR innovation days (SID 2019), Athens.
- R. Dalmau Codina & F.F. Herrema. Encoder-Decoder Approach to Predict Airport Operational Runway Configuration. 9th SESAR innovation days (SID 2019), Athens.
- F.F. Herrema & R. Dalmau Codina. Data analytics driving a bright future in ATM, 2020, www.internationalairportreview.com.
- F.F. Herrema, A. Marsden & B. Desart. A Gradient Boosting model to predict the stand occupancy time at CDG airport, Accepted for the ELSEVIER Special Issue of Transportation Research Part E: Logistics and Transportation. 01-09-2020.