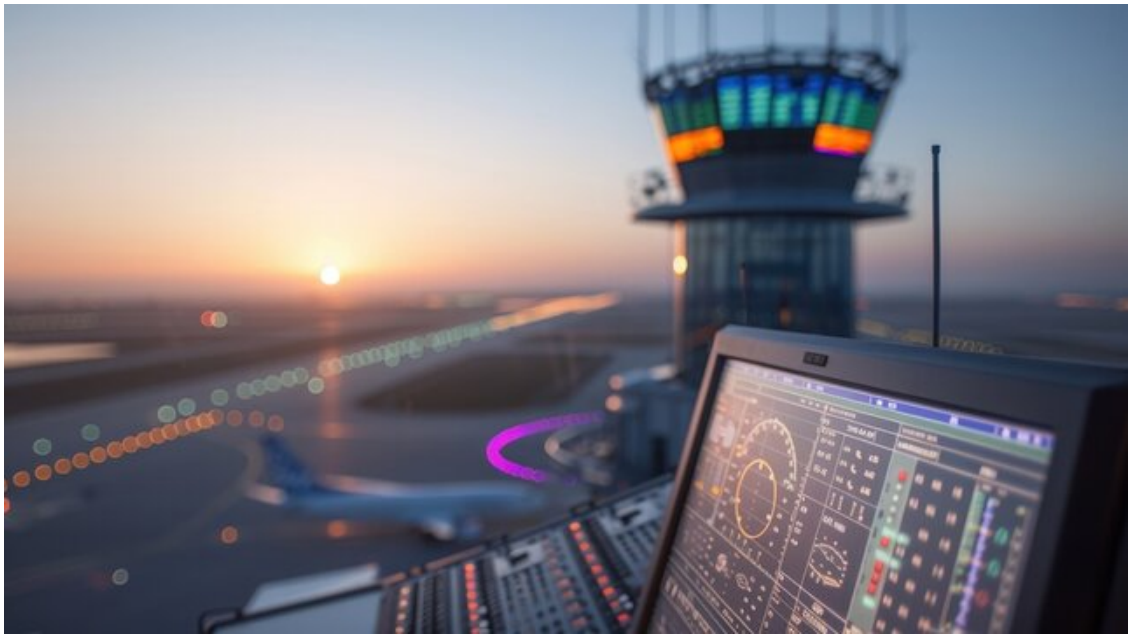
CHALMERS
UNIVERSITY OF TECHNOLOGY



Airport surface dataset for machine learning based taxi-out time prediction

Lisa Holländer
Moa Kallén

DEPARTMENT OF MECHANICS AND MARITIME SCIENCES

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2026
www.chalmers.se

Master's Thesis in Complex adaptive systems

Airport surface dataset for machine learning based taxi-out time prediction

Lisa Holländer
Moa Kallén

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Mechanics and Maritime Sciences
Division of Vehicle Engineering and Autonomous Systems
Chalmers University of Technology
Gothenburg, Sweden 2026

Airport surface dataset for machine learning based taxi-out time prediction
Lisa Holländer
Moa Kallén

Supervisor: Michael Forschlé, Saab Air Traffic Management
Examiner and supervisor: Ola Benderius, Department of Mechanics and Maritime Sciences

Master's Thesis 2026
Department of Mechanics and Maritime Sciences
Chalmers University of Technology
SE-412 96 Gothenburg
Sweden
Telephone +46 31 772 1000

Cover: Air traffic control tower at night with sunset view [1].

Typeset in LaTeX
Gothenburg, Sweden 2026

Airport surface dataset for machine learning based taxi-out time prediction
Lisa Holländer
Moa Kallén
Department of Mechanics and Maritime Sciences
Division of Vehicle Engineering and Autonomous Systems
Chalmers University of Technology

Abstract

Airport surface operation is an important part of air traffic management. It is a critical contributor of overall flight efficiency. Accurate prediction of taxi-out time for aircraft can help reduce delays and allow for more robust scheduling. This thesis uses airport surface movement data to predict taxi-out time using machine learning. By using radar and sensor data collected at one major airport, the goal is to create a dataset suitable for machine learning.

To achieve this, a comprehensive preprocessing pipeline was developed to harmonize airport data from A-SMGCS surveillance systems and ASTERIX-based systems. Key features influencing taxi times, such as traffic congestion, aircraft characteristics, airport infrastructure, temporal patterns, and weather conditions, were identified through a literature review. This thesis reconstructs the identified features using the available airport data. Additionally, feature selection and multicollinearity analysis were performed using correlation matrices, variance inflation factor (VIF), and Cramer's V statistics to ensure the robustness of the predictive model.

Finally, in order to evaluate the dataset, two supervised machine learning models were used. The models implemented were Multiple linear regression and Random forest. Random forest outperformed linear regression, showcasing its ability to capture nonlinear and complex patterns in the dataset. For a final robust evaluation of the dataset, k-fold cross validation was used on Random forest. The results were then interpreted using SHAP. SHAP identified features pertaining to the airport geometry and congestion-related features as the features with the most impact on the taxi-out time prediction.

This study demonstrates the feasibility and limitations of creating an airport surface dataset for machine learning for predicting taxi-out times. It highlights the importance of the data preprocessing steps, as well as the feature engineering, in airport surface-related predictive modeling.

Keywords: Airport operations, Air traffic management (ATM), Machine learning, Aircraft taxi-out time (AXOT), Taxi-out time prediction, Feature selection, Data harmonization, Explainable AI, SHAP

Preface

This report presents the outcome of our master's thesis project, at the Department of Mechanical Engineering at Chalmers University of Technology during the spring of 2026. The thesis investigates the oppertunity of predicting aircraft taxi-out times using historical airport surface movement data and machine learning algorithms.

Acknowledgements

We would like to express our gratitude and appreciation to all those who supported and contributed to the completion of this master thesis.

First we would like to thank Saab air traffic management (ATM) for giving us with the opportunity to work on this project. Special thanks to our supervisor Michael Forschlé, team leader of software development at Saab ATM, for empowering us, providing continuous feedback and offering valuable advice and contributions. We are also grateful to Robin Viktorsson for his interpretation of the data sources and his extensive knowledge of airport data. Additionally, we would like to thank all other colleagues at Saab ATM for creating a friendly workplace and offering meaningful perspectives in the air traffic management domain.

Furthermore, we would like to thank our examiner, Ola Benderius, for his guidance and support throughout the project. We also want to thank Edvin Ahlström, Johan Svanstedt, and Gabriel Forsberg for their feedback along the way.

Lisa Holländer, Moa Kallén, Gothenburg, June 2026

List of Acronyms

ADS-B	Automatic Dependent Surveillance–Broadcast
AI	Artificial intelligence
ALDT	Actual landing time
AOBT	Actual off-block time
ASTERIX	All purpose structured Eurocontrol surveillance information exchange
ASDE-X	Airport surface detection system - model X
A-SMGCS	Advanced surface movement guidance and control system
ATM	Air traffic management
ATOT	Actual takeoff time
AXOT	Actual taxi-out time
CAT062	Category 062
CSV	Comma separated values
CV	Cross validation
ETA	Estimated time of arrival
ETD	Estimated time of departure
EUROCONTROL	European Organisation for the Safety of Air Navigation
GNSS	Global Navigation Satellite System
ICAO	International civil aviation organization
IQR	Interquartile range
LR	Multiple linear regression
MAE	Mean absolute error
METAR	Meteorological Aerodrome Report
ML	Machine learning
MSE	Mean squared error
NOAA	National Oceanic and Atmospheric Administration
OEW	Operating empty weight
RF	Random Forest
RMSE	Root mean squared error
RSS	Residual sum of squares
MTOW	Maximun takeoff weight
SHAP	Shapley additive explanations
SQL	Structured query language
VIF	Variance inflation factor
WTC	Wake turbulence category
XAI	Explainable AI

Nomenclature

Below is the nomenclature of indices, sets, parameters, and variables that have been used throughout this thesis.

Indices

i,j	Aircraft/feature/observation indices
k	Predictor or fold index

Sets

A_i	Set of aircrafts relevant to aircraft i
X	Set of input features
F	Set of input features in SHAP
S	Feature subset in SHAP

Parameters

n	Number of observations
K	Number of cross-validation folds
k_{lower}	IQR lower whisker value
k_{upper}	IQR upper whisker value

Variables

t_s^i	Taxi start time of aircraft i
t_e^i	Taxi end time of aircraft i

N_i	Number of aircraft taxiing when aircraft i starts taxiing
Q_i	Number of aircraft expected to finish taxiing during taxi process of aircraft i
$\hat{y}$	Predicted target value
ϕ_i	SHAP value for feature i

Contents

List of Figures

List of Tables

1
Introduction

According to a joint report by Airports Council International World and the International Civil Aviation Organization (ICAO), global passenger traffic reached 9.5 billion passengers in 2024, and is expected to exceed 12 billion by 2030 [2]. Following a full recovery from the COVID-19 downturn, long-term air traffic forecasts indicate sustained growth in air travel. This continued growth places increasing pressure on *Air Traffic Management* (ATM). ATM is the set of airborne- and ground based operations that is designed to keep all aircraft movements safe and efficient [3].
In recent years, continuous improvements in ATM have focused on increasing the efficiency of en-route operations, i.e. long-distance and high-altitude flights. This has shifted the operational bottlenecks toward airports [4]. As traffic demand increases, many major airports experience congestion on runways and taxiways, leading to queues, delays and increased fuel consumption [5]. To address these challenges, major hub airports have considered infrastructure investments to enhance the performance. But since infrastructure expansion is often costly and time-consuming, improving the efficiency of existing airport surface operations has become a critical focus area [4, 6].

A key component of airport surface operations is the time the aircraft is taxiing between runway and gate. *Taxi-in* time is defined as the duration between runway exit and arrival at the assigned gate, while *taxi-out* time refers to the time between gate departure and runway entry. Together with turnaround time, which is the period during which an aircraft remains on the ground between arrival and the next departure, these time components directly influence airport efficiency and schedule reliability. Taxi time is affected by traffic density and congestion, runway configuration, and operational control decisions. Variability in taxi times contributes to uncertainty in arrival and departure scheduling, gate allocation, and overall airport performance. An accurate prediction of aircraft taxi-out times can support more robust scheduling and identify choke points between gate and runway, as well as contribute to the estimation of the optimal capacity of the overall airport. To be able to predict the taxi time, a set of features needs to be considered [7].

Simply having a large amount of data does not automatically lead airports towards intelligent and automated airport operations [7]. Recent studies have demonstrated the effectiveness of machine learning techniques for taxi-time prediction and have identified influential operational features. These approaches typically rely on structured datasets derived from surveillance systems or publicly available flight tracking data.

Historical data from airport surveillance systems such as radar and other sensors are often available in a raw data format. This consists of a wide variety of sources, different structures, inconsistent timestamps, incomplete records, noise and event definitions. Creating a dataset that can be used for predictive modelling requires substantial preprocessing, data cleaning, synchronization, and feature engineering [8]. Despite the increasing availability of airport data, comparatively less effort has been placed on how to go from this raw data to a useful dataset.

1.1 Objective

This thesis investigates the feasibility of using historic airport surface data to predicting taxi-out time with machine learning. This is carried out through a comprehensive analysis of recorded data from radars and sensors from a single busy airport.

1.2 Research questions

Through this thesis, based on the objective described, the following research questions are addressed:

- How can the airport surface data be harmonized for machine learning?
- Which features should be selected for taxi-out time prediction?
- Can machine learning be applied to the created dataset for predicting taxi-out time, and how is the performance?

1.3 Limitations

This thesis relies on radar and sensor data provided by an external commercial actor. The selected features are primary based on what information is available in the given dataset, and external sources may be supplemented as general information features such as weather data and ICAO categorizations. The approach in this project is developed for this specific data and may not be a general method for other types of radar and sensor datasets. The machine learning model for taxi-out time will be trained on data from a single airport, and therefore the final model may not perform as well on other airport configurations. Since the data is collected from surface operations, the altitude information is not available and will not be considered.

2
Background

This chapter reviews the types of airport surface data commonly used in taxi-time studies, as well as previous research on dataset construction, influencing factors, feature importance, and machine learning approaches for taxi-out time prediction. Feature contribution and importance evaluation is also included, for validation throughout the process.

2.1 Airport surface movement data

Airport surface movement data are monitored by systems such as *airport surface detection system, model X* (ASDE-X) in the U.S. and *advanced surface movement guidance and control system* (A-SMGCS) in other parts of the world. These systems offer real-time routing, guidance and surveillance for the control of aircraft and vehicles at the airport. Within these services, the position, identification and tracking of vehicles and aircraft on the airport surface are provided. The data is collected from a range of surveillance sensors, including radars, multilateration and global navigation satellite system (GNSS) based broadcasts such as automatic dependent surveillance-broadcast (ADS-B) [9, 10].

For a smooth data exchange from systems within the A-SMGCS framework it is common to use the 'all purpose structured Eurocontrol surveillance information exchange' (ASTERIX) standard, defined by Eurocontrol [11]. The set of ASTERIX standards consists of different categories of surveillance information, for a variety of purposes and functions. For example the Category 062 (CAT062), which carries combined track messages, i.e. the fused position, velocity and identification of aircraft on the airport surface. CAT062 comes in a binary format and gets transformed to a more human interpretable format, which is more suitable to work with when developing applications and analyzing.

While many studies explicitly state the use of ASDE-X or A-SMGCS data, few explicitly reference the use of raw ASTERIX data for taxi-time estimation. This may be because ASTERIX is regarded as a data exchange standard for A-SMGCS data rather than a research focus in its own right.

2.1.1 Data for taxi-out prediction

Actual taxi-out time (AXOT) is typically measured as the difference between the actual off-block time (AOBT), the time the aircraft pushes back and vacates its parking position, and the actual take-off time (ATOT) (Eq. 2.1) [12].

$$AXOT = AOBT - ATOT \tag{2.1}$$

One of the challenges in AXOT prediction lies in the construction of a reliable and structured dataset. Airport surface operations generate large volumes of data originating from multiple systems, often stored in different formats and levels of granularity. Consequently, data cleaning, synchronization and integration are essential steps before predictive modeling can be performed [8]. In many cases, additional processing is required to match, correlate, and merge data from different sources in order to obtain a representation of each flights surface movement [8].

The literature shows a variety of approaches to dataset construction, largely depending on the data structure and availability. Some studies rely on publicly available sources and combine multiple datasets to enrich the data. For example, one such study reconstructed surface trajectories using publicly available surveillance data, ADS-B, and integrated these trajectories with meteorological information in order to investigate the impact of weather on taxi-time prediction [7]. Other studies have utilized structured operational datasets provided by organizations such as Eurocontrol, which offer preprocessed surveillance and flight information.

In contrast, several studies have relied directly on airport-provided surface surveillance systems, such as ASDE-X or A-SMGCS, as the basis for their prediction models [13]. These datasets typically contain high-resolution movement information but require substantial preprocessing to derive structured variables such as taxi time, departure time or route distance. In other studies, researchers have been able to attain datasets directly from airport authorities [14].

Despite the central importance of dataset construction, the preprocessing procedures are often only briefly described in the literature. Although airport surveillance systems rely on standardized data exchange formats such as ASTERIX, the transformation from raw surveillance messages to structured machine learning datasets involves multiple non-trivial steps, including trajectory reconstruction, event detection, filtering, and feature extraction. These steps are rarely documented in detail, creating a gap between raw operational data and the final predictive models presented in published studies.

The way in which raw data is processed and structured directly influences which variables can be extracted and how taxi-out time is ultimately modeled. Therefore, understanding the relationship between data sources, preprocessing strategies, and feature construction is essential.

2.2 Features for taxi-out prediction

Taxi-out time is influenced by a multitude of factors and events. Numerous studies have focused on predicting taxi-out time using various features from datasets. These features can be categorized as follows.

2.2.1 Traffic and congestion factors

Ground congestion is consistently used as a feature during the prediction of taxi-out time. Several studies report that taxi-out duration increases with higher arrival and departure demand, longer runway queues, and a greater number of aircraft simultaneously taxiing on the airport surface [15, 16, 17, 18]. Common congestion-related variables therefore include the number of taxiing aircraft, runway queue characteristics, traffic demand, and recent historical taxi-out times, all of which have been shown to significantly affect predictive performance.

A common way to represent surface congestion is through the number of aircraft already taxiing when a given aircraft begins its movement. This is typically expressed as

$$N_i = \sum_{j \in A_i} \left[t_s^i \in \left(t_s^j, t_e^j \right) \right] \tag{2.2}$$

where N_i denotes the number of aircraft taxiing at the time aircraft A_i starts taxiing. The parameters t_s^j and t_e^j denote the start and end of the taxiing process of the aircraft j, respectively. An aircraft j is considered to be taxiing at the start time of aircraft i if t_s^i lies within this interval.

Several extensions refine this representation by distinguishing between arrivals and departures, thereby capturing directional interactions between different traffic flows. In such formulations, separate variables are defined based on the movement type of the considered aircraft and the surrounding interacting traffic [17, 19]. Separating arrivals and departures makes it possible to differentiate between the types of aircraft being modeled and the types of traffic being counted. These interaction-specific variables, which represent the amount of traffic, allow models to measure the influence of other operations on taxi time.

Another important group of variables describes the state of the runway queue. Runway queue position, defined as the number of aircraft that will depart before the aircraft under consideration, is a strong predictor of taxi-out time [20, 21]. Closely related measures include queue length, such as the number of departures ahead during taxi-out, or arrival queue length for taxi-in operations. Some studies further incorporate nonlinear terms, such as quadratic queue length, to capture the increasing marginal impact of congestion as queues grow longer [22]. Other studies also obtain queue-related variables using ASDE-X data to superimpose key attributes that capture the traffic situation on a digitized airport map, which then provides information on aircraft positions and movement status [20].

Traffic demand is also commonly represented in datasets for prediction and can be obtained through aggregated measures of arrival and departure activity over fixed time. These include counts of departures and arrivals, as well as rates computed over rolling time windows (e.g., 15, 30, or 60 minutes). Such variables capture short-term fluctuations in airport activity and serve as indicators of congestion buildup. In addition, moving averages of recent taxi-out times are frequently used as proxies for downstream congestion and operational constraints [16, 20, 18].

In addition, several studies highlight the effect of interactions between different traffic flows [23, 20, 7]. In particular, arrival flows can interfere with departure operations through runway crossings or shared runway use, leading to additional delays. The impact of these interactions depends on the operational context, including runway configuration and traffic mix, and is therefore often modeled through combinations of arrival and departure demand variables. Some studies also account for more complex interactions, such as runway crossings and parallel runway operations, by incorporating arrival and departure rates from interacting runways [20]. Additionally, the influence of surrounding traffic may differ depending on whether aircraft is arriving or departing [7].

Although the above variables are often defined in terms of time intervals that depend on the full trajectory of an aircraft, their implementation in predictive settings requires reconstructing the state of the system at the time of prediction. When using historical datasets, this can be achieved by determining whether other aircraft are active at a given time based on their timestamps. For example, an aircraft can be considered to be taxiing at time t if its off-block time occurs before t and its takeoff time occurs after t. This approach allows congestion-related variables to be computed using only information available at the prediction time, thus avoiding information leakage.

2.2.2 Aircraft and operational factors

Aircraft-specific and operational characteristics have been shown to influence taxi time. Differences between departures and arrivals are commonly observed, with departures often exhibiting lower average taxi speeds due to runway queuing effects. This has been implemented in studies using a variable that tells whether the flight is a departure or arrival [17, 14].

Furthermore, aircraft features are often used in studies related to taxi-out time prediction. Common features include aircraft type, flight number and airline [7, 24, 6, 25, 26, 27, 13, 28, 29]. Specifically, whether the airline was a low-cost airline or not has been shown to affect the taxi-out time, with low-cost airlines resulting in higher taxi-out times. The distinction between international and domestic flights has also been studied for its impact on taxi time [21].

Geometrical and operational aircraft features further influence taxi out time. Factors such as the number of engines, wake turbulence category, aircraft weight category,

and wing span require varying operational handling and separation, thereby affecting taxi time [7, 24, 6, 25, 26, 27, 13, 28, 29, 21]. Moreover, the sequence of previous flights on the runway, particularly the mix of different aircraft weight types, can also impact taxi time.

2.2.3 Spatial and infrastructure factors

The airport layout and infrastructure characteristics play a significant role in determining taxi time. The total taxi distance between stand and runway is consistently identified as a key predictor, as longer taxi routes naturally require more time [17]. Ideally, the exact distance of taxiing route for each flight would be obtained to account for its impact on taxiing time. However, such detailed information about the taxi route is not always available. As a result, approximations are often used to capture spatial variability. In some studies, this is approximated using simplified measures such as the straight line distance from the aircraft to the runway threshold [20].

Another approach is to use route-specific variables that describe combinations of origin and destination points [19]. So-called 'area names' can be introduced, which serve as binary indicators representing specific stand-runway combinations. For arrivals, these variables can be constructed based on the combination of landing runway and the parking stand, while for departures they are defined by the departing stand and takeoff runway. These variables can be known prior to starting operations and act as proxies for multiple route-dependent factors influencing taxi time, including taxi distance, angle taken by aircraft, and runway configuration.

Another common approach is to use runway configuration as a proxy for taxi distance and routing complexity, since different configurations imply different paths between terminals and runways [22, 20]. This approximation is more accurate for airports with a simple layout, for example those which have only one runway for arrivals and another for departures.

Runway configuration and direction of operation further impact taxi time by affecting both taxi distance and interactions with other traffic [7]. This is especially true for airports with multiple runways, where which runway is active can be affected by weather conditions, traffic demand and other factors to optimize operational efficiency [22]. To capture these effects, some studies use historical information on runway usage and introduce variables representing frequently used arrival and departure configurations [20].

Furthermore, geometric properties of the taxi route have also been shown to affect taxi duration. In particular, the cumulative turning angle and the number of sharp turns influence taxi speed, as aircraft must reduce speed when maneuvering [17, 7]. Similarly, long straight taxiway segments may allow higher speeds, while push back related movements such as 180-degree turns can result in additional delays [7].

2.2.4 Temporal factors

Temporal variables capture recurring patterns in airport operations and have been shown to significantly influence taxi-out time. Time of day is frequently identified as an important state variable, suggesting the existence of daily traffic surges [15, 16].

In addition to capturing long-term patterns, temporal features can also represent short-term operational dynamics. For example, historical performance indicators, such as the average taxi-out time during recent time intervals, have also been used to capture short-term operational trends [15]. These variables enable predictive models to account for systematic variations across different periods of the day.

Commonly used temporal features include the hour of the day, minute of the hour, and day of the week, which together captures both daily and weekly patterns in traffic demand and operational conditions [29, 24]. There has also been use of seasonal features, such as summer or winter [30].

2.2.5 Weather factors

Weather conditions are known to affect both aircraft performance and airport surface operations, and are therefore commonly included in taxi-time prediction models. When weather conditions are suboptimal, the impact on airports and the surrounding airspace could be a reduced visibility for pilots and ATM controllers [22]. This leads to a decline in capability for traffic management and decreased airport capacity.

To capture these effects in prediction models, data-driven approaches frequently integrate meteorological data together with traffic and flight data to enhance predictive performance. Such data is often obtained from sources such as the National Oceanic and Atmospheric Administration, METAR Weather reports, Open-Meteo, and weather-related traffic management initiatives such as the Severe weather avoidance plan [31, 7, 20, 32].

A wide range of weather related variables have been examined, including air pressure, temperature, wind speed, visibility, rain, snow, drizzle, fog, mist, haze, lightning, and cloud ceiling [7, 22, 28, 13, 26, 25]. These factors influence taxi speed, runway configuration, and ATM controller decisions, thereby indirectly affecting taxi-out duration.

2.3 Machine learning methods for taxi-out time prediction

Recent studies have employed various *machine learning* (ML) techniques to support collaborative decision-making in airports [33]. Taxi-out time prediction is typically formulated as a data-driven regression problem, where historical operational data

are used to model relationships between engineered features and observed taxi duration [4]. Both traditional regression models and more flexible ML approaches have been applied in the literature. Airport surface movement data is often high-volume and high-dimensional, with attributes such as time stamps, coordinates, and other operational variables [8]. Depending on the modeling approach, these methods can capture different types of relationships between features and taxi-out time.

In the context of ATM, the interpretability of artificial intelligence (AI) systems is an important consideration because of the critical nature of the field. Previous research has highlighted that explainability is still only partially addressed in many AI applications in ATM, despite its importance for building trust in automated decision-support systems [34]. Consequently, understanding the underlying factors influencing predictions, such as congestion, taxi routes, and operational delays, is essential for improving both predictive performance and operational decision-making.

2.3.1 Taxi-out prediction as a supervised regression problem

Supervised learning refers to a class of machine learning methods in which a model learns a mapping between a set of input variables X and an output variable y and applies this mapping to predict outputs for unseen data [35]. This relationship can be expressed as

$$f : X \to y \tag{2.3}$$

where X represents the set of input features and y represents the target variable. Depending on the type of target variable, supervised learning can be used for classification, where the target variable is categorical, or regression, where the target variable is continuous. In the case of taxi-out time prediction, the target variable represents a continuous duration, and the problem is therefore formulated as a supervised regression task.

2.3.2 Statistical diagnostics and data validation

This section focuses on statistical diagnostics and data validation techniques applied to flight delay prediction datasets. These techniques improve the quality and relevance of the data, leading to more accurate and reliable prediction models.

2.3.2.1 Outlier detection

Outlier detection is a critical aspect of data cleaning for flight prediction datasets. Outliers are often the result of extreme conditions, and may not be relevant to predict regular operations [36]. Removing or handling outliers helps ensure that the dataset reflects typical flight conditions, which will contribute to more accurate predictions.

One statistical technique used to detect outliers is the interquartile range (IQR). IQR is obtained from the difference between the 75th percentile (Q3), and the 25th

percentile (Q1), along with defining lower and upper bounds (Eq. 2.4-2.6).

$$IQR = Q3 - Q1 \tag{2.4}$$

$$\text{Lower Bound} = Q1 - 1.5 \cdot IQR \tag{2.5}$$

$$\text{Upper Bound} = Q3 + 1.5 \cdot IQR \tag{2.6}$$

The constant 1.5 is the standard value, even though other values such as three can be applied when defining a set of extreme outliers. The data points that fall outside the bounds are classified as outliers and can be further visualized using a boxplot. Since the IQR method uses quartiles, which is related to the median, this makes it more robust to extreme outliers compared to other methods which use the mean and standard deviation based of the normal distribution. This makes the IQR method more suitable for a slightly skewed dataset [37, 38].

2.3.2.2 Collinearity

Collinearity occurs when the independent variables in a model are correlated with each other [39]. This overlap makes it difficult to isolate the unique predictive value of individual variables, as they essentially act as proxies for each other. Consequently, while the overall model might indicate that it explains variation in the dependent variable, the individual predictors often appear non-significant because the model cannot distinguish which specific variable is driving the effect. As such, when severely collinear independent variables are used in a model, it can be virtually impossible to decide which predictors are in fact related to the dependent variable. Additionally, including correlated or irrelevant features can lead to model overfitting or decreased prediction performance [36].

There are multiple methods of assessing the amount of collinearity in a set of independent variables [39]. One simple way to detect collinearity is to plot the *correlation matrix* of the predictors [40]. A correlation matrix is a table that displays the *correlation coefficients* between variables. These coefficients are often obtained using Pearson's correlation, which is calculated as the covariance of two variables divided by the product of their standard deviations. An element of the correlation matrix with a large absolute value indicates a pair of highly correlated variables, suggesting a collinearity problem in the data.

However, not all collinearity problems can be detected by inspecting the correlation matrix. It is possible for collinearity to exist between three or more variables even if no pair of variables has a particularly high correlation. This is called multicollinearity, which can be assessed by computing the variance inflation factor (VIF). The VIF measures how much the variance, which is the square of the standard error, of a coefficient is increased because of collinearity [39]. This can be expressed as

$$VIF(\hat{\beta}_j) = \frac{1}{1 - R^2_{X_j|X_{-j}}} \tag{2.7}$$

where $R^2_{X_j|X_{-j}}$ is the R^2 from a regression of X_j onto all other predictors. If the VIF is 1, there is no multicollinearity at all. A VIF value that exceeds 5 indicates

a problematic amount of collinearity. If it is very large, such as 10 or more, multicollinearity is a serious problem. When identified, it can be fixed by removing one of each highly correlated feature [41].

However, both the correlation matrix and VIF can only be used for continuous data. For categorical data, or mixed data, these methods cannot be used since they do not follow a normal distribution or have equal distances between the intervals [42]. As an alternative, Cramer's V can be used to measure the correlation between categorical features. This statistic ranges from 0 to 1, where a higher value indicates a stronger association. It is particularly useful since it provides a standardized measure of association strength, making it easy to interpret, although it cannot determine the direction of the relationship. It is calculated as

$$\phi_C = \sqrt{\left(\frac{\chi^2}{N(L-1)}\right)} \tag{2.8}$$

where χ^2 is the chi-squared statistic, N is the sample size, and L is the smaller value of either the number of columns or the number of rows.

2.3.3 Linear models

Linear models in machine learning are supervised algorithms that predict outcomes by fitting a straight line (or hyperplane) to data, assuming a linear relationship between input features and the target variable. The relationship between input variables and predicted taxi-out time can be directly interpreted through model coefficients, allowing insight into how individual operational factors influence the prediction [43]. Simple regression approaches include linear models such as simple linear regression and multiple linear regression. In this report *Linear regression* (LR) refers to multiple linear regression. The LR model can be expressed as

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_k + \epsilon \tag{2.9}$$

where k is the number of distinct predictors, X_j represents the jth predictor and coefficient β_j quantifies the association between that variable and the response [40]. The random error or disturbance term is represented by ϵ. β_j is interpreted as the average effect on Y of a one unit increase in X_j, holding all predictions fixed [44, 39, 40].

The regression coefficients $\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_k$ are unknown, and must be fitted. Given fitted $\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_k$ predictions can be made using the formula in Eq. 2.10.

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \cdots + \hat{\beta}_k x_k \tag{2.10}$$

The parameters are estimated using the least squares approach, as shown in Eq. 2.11.

$$RSS = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_{i1} - \hat{\beta}_2 x_{i2} - \cdots - \hat{\beta}_k x_{ik})^2 \tag{2.11}$$

$\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_k$ are the multiple least squares regression coefficient estimates, which are chosen to minimize the sum of squared residuals (RSS) [40].

Due to their simplicity and transparency, linear models are often used as baseline models in machine learning studies. Since the structure is more interpretable, this allows researchers to gain insight into which features that influence the prediction before applying more complex models. This way the linear models provide a useful benchmark for evaluating whether more advanced methods offer meaningful improvements in predictive performance or not.

2.3.4 Tree-based ensemble methods

Tree-based ensemble methods such as *Random forest* (RF) and *Gradient boosting* are often used in taxi-time prediction since they have a strong predictive performance and an ability to highlight feature importance [4, 7]. These models can also capture the potential nonlinear relationships between features and taxi-out time.

RF constructs an ensemble of decision trees, and then averages their predictions to produce the final output. With this ensemble approach, the model is more robust to outliers and reduces the risk of overfitting compared to just a single decision tree. Several studies have shown that RF models generalize well across different datasets and can achieve a strong predictive performance in taxi-time prediction tasks [4]. It also often produces strong results even with limited hyperparameter tuning [7].

Gradient boosting, in contrast, builds trees sequentially and allows each new tree to correct errors made by previous ones. This makes it very effective in capturing complex patterns and handling uncertainty in operational conditions. For example, [18] found that while linear models, ordinary least squared and ridge regression performed well under stable and predictable runway configurations, gradient boosting was better at capturing variability and effects associated with a more complex operational environment.

2.3.5 Evaluation of machine learning models

In order to evaluate the prediction performance of the models, commonly used performance metrics for a continuous target variable include *root mean squared error* (RMSE), *mean squared error* (MSE), and *mean absolute error* (MAE) , (Eq. 2.12–2.14). These metrics quantify the deviation between predicted values, $\hat{y}$, and actual values, y. Lower values of RMSE, MSE and MAE indicate that the prediction accuracy is higher [4]. These metrics are expressed as

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{2.12}$$

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{2.13}$$

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|\,. \tag{2.14}$$

Performance metrics also include *coefficients of determination* (R^2) score, which describes the proportion of variance in the dependent variable that is explained by

the independent variables (Eq. 2.15).

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{2.15}$$

$$\tag{2.16}$$

In addition to performance metrics, *learning curves* illustrates how performance changes with the training set size. Learning curves can be useful for identifying if models are likely to be overfitting or underfitting, while RMSE and *k-fold cross-validation* (CV) can provide a more reliable estimate of predictive performance [18].

In K-fold CV, the data is split into K 'folds'. In each run, one fold is held as the testing or validation set, and the remaining $K-1$ folds are used to train the model. The K-fold CV calculates the average performance across all folds, resulting in a more reliable estimate (Eq. 2.17). The K-fold CV can be expressed as

$$\text{CV} = \frac{1}{K}\sum_{k=1}^{K} L^{(k)}, \tag{2.17}$$

where L is the loss on the fold k. However, while the performance metrics such as RMSE and R^2 quantify the accuracy of a prediction, they do not offer any insight into the underlying operational drivers for the prediction. To get insight into what drives the prediction, previous studies have looked at feature importance [7].

2.3.6 Feature contribution and importance evaluation

Evaluating feature contributions is crucial for understanding the underlying factors that influence taxi-out time predictions. Understanding the rules underlying the model's decisions not only increases trust in its predictions but also provides insights into how the modeled process works and offers intuition for improving the results [24]. While different machine learning models offer various built-in methods for assessing a features contribution to the taxi-out prediction, these methods can be categorized within a broader theoretical framework.

2.3.6.1 Taxonomy of XAI

Explainable AI (XAI) methods are commonly divided into *intrinsic* explainability and *post-hoc* explainability [45]. Intrinsic methods refer to models that are transparent by design, such as linear regression, where the internal logic is mathematically accessible through the model's structure. In contrast, post-hoc means that the explainability is achieved at a later time after the model creation. Post-hoc is typically applied to models that are not always interpretable, such as RF or neural networks.

Furthermore, post-hoc explainability may further be divided into model-specific or model-agnostic techniques. Model-specific methods are tailored to the internal mechanisms of a particular class of algorithms, while model-agnostic techniques

treat the model as a black box and works without accessing the model's internal parameters.

Another important distinction relates to the scope of the explanation. Regardless of whether a method is intrinsic or post-hoc, it can be further categorized into global and local explanations. Global explanations allow a user to understand how the model works across the entire range of data, identifying general trends and long-term drivers. Local explanations provide specific explanations for individual predictions.

2.3.6.2 Shapley Additive Explainations (SHAP)

Shapley additive explanations (SHAP) is a local post-hoc explainability technique that has been used to quantify feature contributions in machine learning models [33, 45]. Multiple studies have used SHAP to assess feature importance results from taxi-out prediction [24].

SHAP assigns each feature a contribution value for a specific prediction based on cooperative game theory. A positive SHAP value indicates that a feature increases the model prediction relative to the baseline prediction. Conversely, a negative SHAP value indicates that the feature decreases the prediction relative to the baseline prediction. The magnitude of the SHAP value reflects the strength of the feature's contribution. The SHAP value for feature i is defined as

$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} \left[f_{S \cup \{i\}}(x_{S \cup \{i\}}) - f_S(x_S) \right], \tag{2.18}$$

where F denotes the set of all features and S represents a subset of features excluding feature i. The terms $f_{S \cup \{i\}}(x_{S \cup \{i\}})$ and $f_S(x_S)$ represent the model prediction based on the feature subset $S \cup \{i\}$ and the subset S, respectively. The difference measures how the prediction changes when feature i is added to the subset S [46].

As such, SHAP operates primarily as a local explanation framework, tailored to individual predictions. These results are often visualized using Waterfall plots. To reach a global understanding of model behavior, local SHAP attributions are aggregated across the entire dataset. The mean absolute SHAP values are calculated for each feature, establishing a global feature importance metric that reflects overall impact on the models target space. This is often visualized using beeswarm and bar plots.

Furthermore, SHAP has model-specific approximation methods. For linear models, LinearSHAP assumes input feature independence, allowing SHAP values to be approximated directly from the model weights coefficients. Conversely, for tree-based ensembles such as RF and gradient-boosting models, the TreeExplainer optimizes the calculation process while still capturing more complex and non-linear feature interactions [47].

3
Methods

This chapter outlines how the dataset for supervised machine learning to predict AXOT was created, including data harmonization, feature construction and validation process. The dataset consists of one observation per row, where each observation corresponds to a single flight and contains features available prior to departure, along with the ground truth label AXOT.

Instead of relying on exploratory feature engineering, this thesis follows a literature-review approach, where features previously identified as relevant for AXOT prediction are assembled from the initial set of CSV files. Tab. 3.1 provides a summary of these features, which guides the data harmonization and feature construction process.

Table 3.1: Features derived from the literature review.

Category	Features mentioned in Literature Review
Traffic and congestion	Taxiing departures, taxiing arrivals, arrival demand, departure demand, runway queue position, runway queue length, arrival queue length, aircraft movement status, counts of departures and arrivals (15, 30, 60 min), rates over rolling windows, moving averages of recent taxi-out times, runway crossings, and shared runway use.
Aircraft and operational	Departure vs. arrival indicator, airline type (low-cost vs. non-low-cost), aircraft weight category, aircraft type, number of engines, wake turbulence category, wing span, flight type (domestic vs. international) .
Spatial and infrastructure	Total taxi distance, straight-line distance to runway threshold, stand-runway combinations (area names), runway configuration, direction of operation, cumulative turning angle, and number of sharp turns.
Temporal	Time of day, hour of the day, minute of the hour, day of the week, season, and average taxi-out time during recent time intervals.
Weather	Air pressure, temperature, wind speed, wind direction, visibility, rain, snow, drizzle, fog, mist, haze, hail, thunderstorm, lightning strike, and cloud ceiling.

3.1 Data harmonization

The data used in this project originates from A-SMGCS surveillance recordings structured in ASTERIX format. It has passed through multiple transformation layers for easier data interpretation by the external commercial partner of this thesis and has been exported as comma-separated values (CSV) files. These CSV files form the basis of all analysis and modeling in this thesis. Initially, the dataset consisted of 10 234 files, with a total file size of approximately 90 GB. The dataset covers several days across the 2019–2025 period.

To be able to use machine learning for prediction of AXOT, there was a need for one single dataset. From having a dataset consisting of a large amount of CSV files, the first step of the process was data harmonization. Unlike basic data integration, which focuses primarily on moving or combining data from one system to another, data harmonization addresses how that data is interpreted. It resolves differences in terminology, definitions, measurement units, hierarchies, and logic. In a harmonized environment, key metrics mean the same thing everywhere they appear. The ap-

proach described below was carried out as a pipeline that could be fed similar data, keeping the important information for the next step in order to build the machine learning dataset (Fig. 3.1).

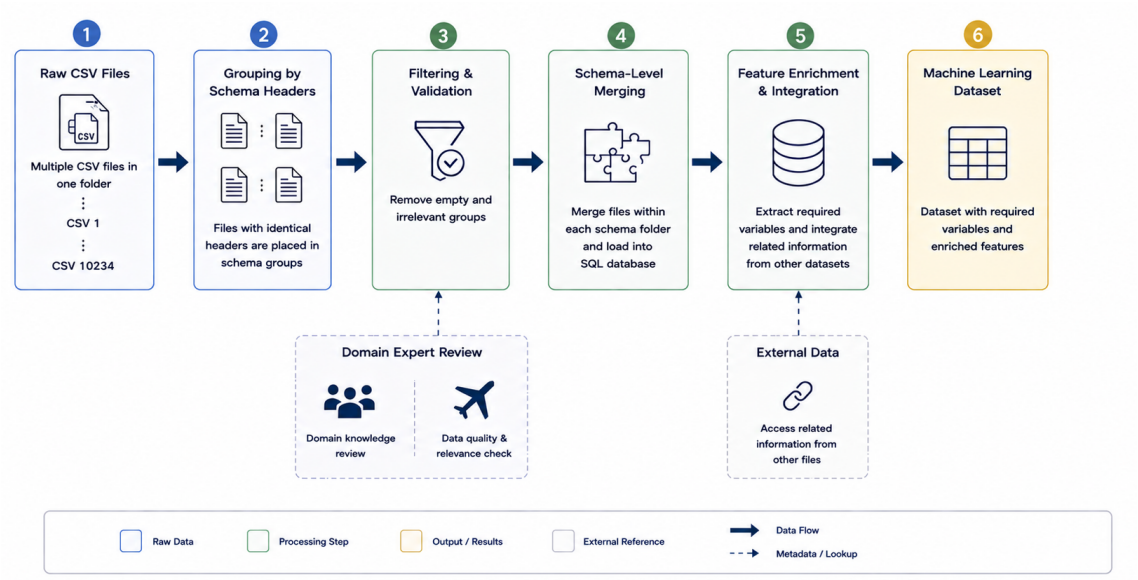


Figure 3.1: Data harmonization workflow.

3.1.1 Schema based grouping

Although 8 724 out of the 10 000 total original files only consisted of a header row, these files still provided critical information. While they did not include any observational data, the header row was present in all files, defining the schema of each file. This was crucial for understanding the structure of the data and ensuring consistent data integration.

Furthermore, the filenames contained structured metadata such as the recording date and the type of data exported from the underlying Structured query language (SQL) database. During closer inspection, it became clear that files that shared the same file name, except for the dates, also shared identical headers. This showed that they originated from the same database, but were recorded on different dates. This information enabled schema based sorting of the files, where the files where grouped in folders according to a strict schema-based matching script. Only if two files contained the exact same header, were they placed in the same group. Each folder contained all CSV files with identical headers, resulting in the 10 000 files being divided into 169 schema groups.

3.1.2 Data cleaning and merge

After the schema-based grouping, additional filtering steps were performed to remove groups that were either empty or irrelevant for taxi-out time prediction. First, an overview analysis of the schema groups was conducted using a summary script to determine the amount of recorded data in each group. Groups consisting entirely

of empty files (including files containing only header rows and no observations) were removed, which were 35 groups consisting of 2 236 files. Next, the remaining schema groups were reviewed in consultation with domain experts and data analysts at the external commercial actor. Based on this review, 51 additional groups with in total 4 879 files were identified as irrelevant for taxi-out time analysis and were therefore excluded from further processing.

After removing both empty and irrelevant groups, 37 schema groups, consisting of a total of 3 119 files remained. These groups were still structured as folders containing all files with identical headers. The files within each schema group were merged into a single CSV file to increase readability. For each group, the header from the first file was retained, and all available data rows from the remaining files were appended. During this merge, there was also a new column introduced to save the original filename of each row, to be able to track the information back to its original filename. However, this was only implemented for the files that contained observational data, not the files that only contained header rows.

3.1.3 Dataset selection

A script was created to identify unique ids and shared header fields across the 37 remaining CSV files, such as flightid or trackid. This enabled inspection of which files contained flight-level information, and how the information in the 37 files were connected. Another script was created to gather all available information for one flight across the 37 files, using its unique ids. Out of the 37 tables, 10 tables contained the flightid and 11 tables the trackid for one specific flight. Overall, 17 tables existed that contained any or both of these ids and could be connected to each flight.

A timeline of the flight was then made using timestamps connected to the flights unique ids, to see the order in which the different files recorded information on the flight and how the flight moved across the airport. This made up a complete timeline of important information. It could be seen that some of the files activated one hour before the flights estimated off-block time, and then again one hour after. Other files only activated during the actual progression of the flight, with columns such as 'occurred'. This suggested that some files were summary files, aggregating information after the flight, while others were event-based files which logged real-time flight information.

Through this inspection, one schema group emerged as containing the most relevant variables for taxi-out prediction. These included flightid, trackid, atot, aobt, runway, stand and aircraft type (Tab. 3.2). The variables atot and aobt were crucial for creating the target, ground truth label AXOT. Stand and runway describe the locations of the start and end of the taxi-out journey. Aircraft type were stored as ICAO codes, which mapped different airplanes specifications and configurations.

Table 3.2: Required variables as an initial dataset for actual taxi-out time prediction.

Category	Variable	Description
Identifiers	flightid	Unique id of a flight
	trackid	Unique id of flight track data
Label construction	aobt	Actual off-block time (AOBT)
	atot	Actual takeoff time (ATOT)
Features	runway	Assigned runway of departure
	stand	Assigned stand at start
	aircraft type	Three letter airline ICAO code

From this schema group, only rows that had data in all of the required variable columns were extracted. This meant that arrival flights or canceled flights were not included in the final dataset. The extraction meant that 14 382 rows out of the 45 627 total rows available in the schema group were added to the final dataset. Initial inspection showed that these rows were recorded over different periods of time (Tab. 3.3).

Table 3.3: Inspection of dates of dataset selection observations.

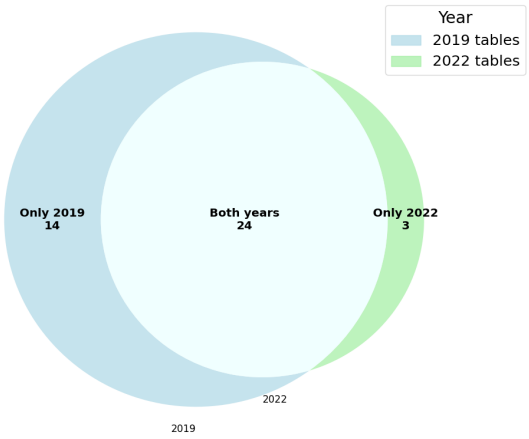
	Year	# observations	Month	# Days
	2019	649	July	1
	2022	13728	October	23
	2022	5	September	1
Total		14382		

Tab. 3.3 showed a difference in distribution between the amount of observed flights between the different years, as well as between the different months. The five flights in September were only recorded for one day, the 30th of September. From the reconstructed timeline it was possible to see that there were tables that activated one hour or more before a flights scheduled departure. As such, the five flights belonging to September were most likely flights with a departure date in October. The five flights from September were dropped from the dataset.

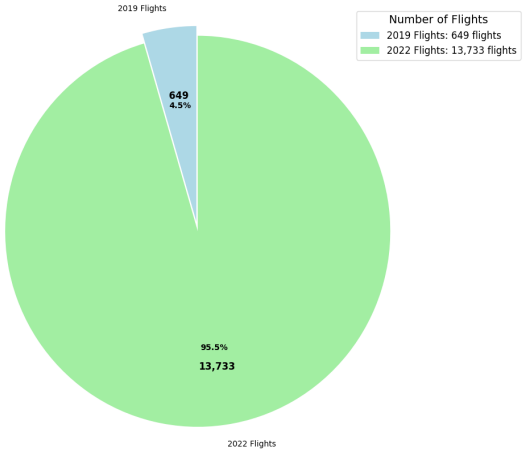
Several variables required for feature construction (Tab. 3.4) were not directly available in the chosen schema group. These variables were supplemented from other schema groups using the flightid or trackid as keys.

3.1.4 Temporal selection and information inconsistency

During inspection it was observed that certain data, such as weather information, seemed to be available from only one day in 2019. This discrepancy was investigated further.



(a) Venn diagram of schema groups tables available across the years for 2019 and 2022.



(b) Pie chart showing the distributions of flights in the data, for 2019 and 2022.

Figure 3.2: Flight data available across years

Fig. 3.2a illustrates the discrepancies in available schema group tables between 2019 and 2022. Out of the total 52 tables in the database, 14 tables existed for 2019 but not for 2022. These tables contained the data necessary to construct weather and spatial features. However, Fig. 3.2b revealed that flights from 2019 make up only 4.5% of the total observed flights in the filtered data. Given that the intent was to use temporal features such as weekday, this would have introduced a bias towards the weekday Monday.

Additionally, the previous timeline script was compared for the flights in 2019 and flights in 2022. In 2019, 17 tables existed that could be connected to a flights flightid or trackid. In 2022, that number dropped to three. Only the summary tables remained, the event based table data were all connected to 2019.

Since the weather-related features and spatial mapping data were only available for the 2019 data, which constitutes a small fraction of the dataset's observations, these features were discarded. Additionally, the spatial mapping data lacked comprehensive information about all runways at the airport, making it unsuitable for generalization across the entire dataset. Consequently, all flights associated with the year 2019 were removed from the dataset to ensure the robustness of the data.

3.2 Feature construction

After selecting the required variables as an initial dataset, the next step was to construct all features used for predicting actual taxi-out time (Tab. 3.4). Some feature categories needed external information to be completed, such as aircraft, operational and weather features. Other categories such as temporal, as well as traffic and congestion, could be engineered from the schema tables. Each feature setup and origin is further described in the following sections.

After each feature had been constructed and added to the dataset, cleaning, analysis and inspection was performed to ensure the robustness of the dataset.

Table 3.4: Final feature set used for AXOT prediction.

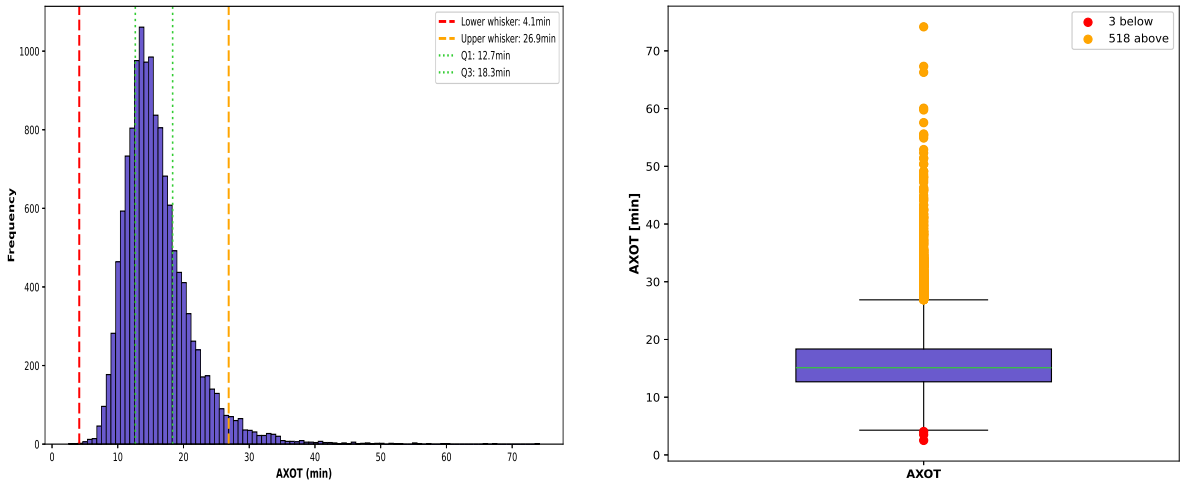
Category	Feature	Description	Encoding
Temporal	Hour	Hour of the day when aircraft taxiing	Numerical
	Weekday	Day of the week when aircraft taxiing	Categorical
Aircraft and operational	MTOW	Maximum takeoff weight	Numerical
Traffic and congestion	Runway queue	Departing aircrafts in queue for a runway at AOBT	Numerical
	Delay previous hour	AOBT more than 5 minutes later than ETD	Numerical
	Taxiing departures	Taxiing departures in airport at AOBT	Numerical
	Taxiing arrivals	Taxiing arrivals in airport at AOBT	Numerical
Spatial and infrastructure	Stand	Assigned stand segment area	Categorical
	Runway	Assigned runway of departure	Categorical
Weather	Rain	Precipitation	Numerical
	Wind speed	measured 100 metres above ground	Numerical
	Wind direction	measured 100 metres above ground	Numerical
	Cloud coverage	Coverage of low altitude clouds	Numerical

3.2.1 AXOT

The target variable, AXOT, had a wide spread of values. To investigate outliers, the IQR method was applied (Fig. 3.3). The whiskers were set to the default value in

IQR analysis, $k_{low} = k_{upper} = 1.5$. Consequently, 518 outliers were identified above the upper whisker at 26.9 min. The total amount of discarded AXOT outliers were 521, corresponding to 3.82% of the total dataset.

The removed observations were mainly located in the extreme upper range of the distribution and risk affecting the model's results. By removing these values, the model could act more robust and reduces the risk that a few extreme observations dominate the analysis.

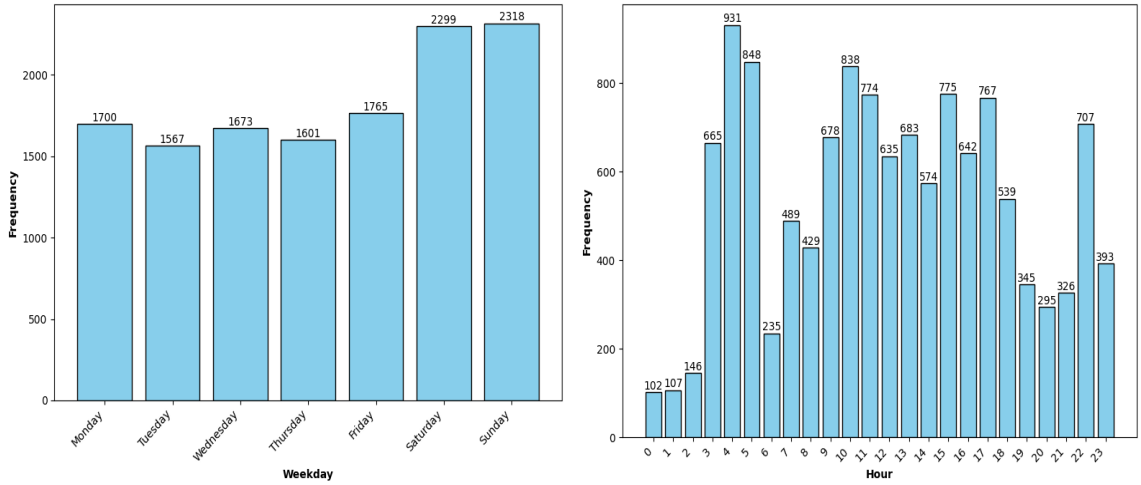


(a) Distribution of AXOT values $k = 1.5$ **(b)** Boxplot $k = 1.5$

Figure 3.3: Observation distributions of AXOT and outliers using the IQR method.

3.2.2 Temporal features

The hour of the day and weekday were included as temporal features, both obtained directly from AOBT. These features capture dynamic congestion patterns, since taxi times may vary depending on the time of day and month. The distribution of the temporal features are shown in (Fig. 3.4). No outliers were removed in the temporal feature category.



(a) Day of the week **(b)** Hour of the day

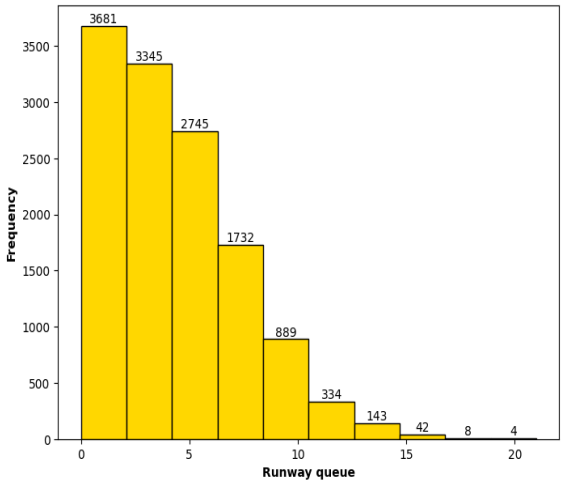
Figure 3.4: Flight observations dataset distribution of the weekday and hour

3.2.3 Traffic and congestion features

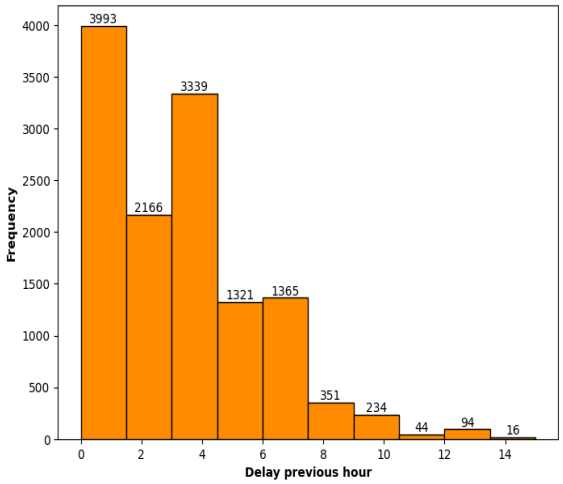
A runway-specific queue length feature was created to represent local congestion at the departure runway, called *runway queue*. It works by counting the number of aircraft already in the taxi-out phase at the time of each flight's AOBT, with the condition that the aircrafts are assigned to the same runway. Specifically, for each flight, all preceding flights heading to the same runway with a block-off time earlier than the given flight's AOBT and without a recorded ATOT were identified. These flights were considered to be part of the departure queue. The feature distribution is shown in Fig. 3.5a.

Additionally, two global congestion features was introduced, *taxiing departures* and *taxiing arrivals*. This represent the overall traffic at the airport surface in both directions. Taxiing departure counts the total number of aircraft in the taxi-out phase at the time of each flight's AOBT, regardless of runway assignment. As in the previous definition, aircraft were included if they had pushed back but not yet taken off. Taxiing arrivals counts the total number of aircraft in the taxi-in phase at the time of each flight's AOBT. This feature was added since the number of arrivals is also indicative of the overall capacity of the airport. The distribution of features values are shown in Figs. 3.5c, 3.5d.

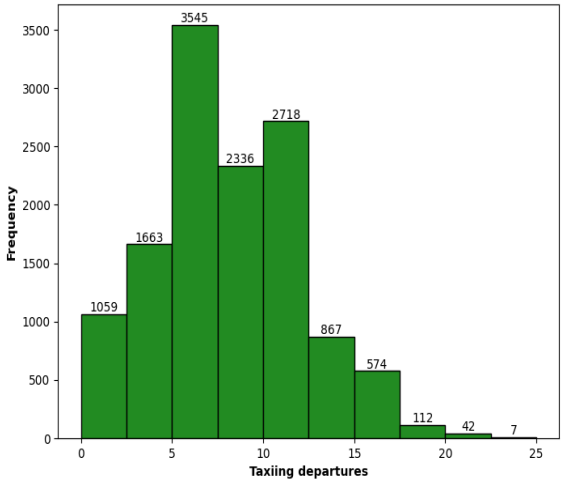
To capture short term temporal dynamics within traffic and congestion that may influence AXOT, a feature *delay previous hour* was introduced. This feature calculates the how many flights that had a departure delay more than 5 minutes of their ETD, within the sliding window of the previous hour. Specifically, it captures the difference between the historical departure flights ETD and ATOT, providing a measure of recent efficiency. The feature distribution of delay previous hour is shown in Fig. 3.5b.



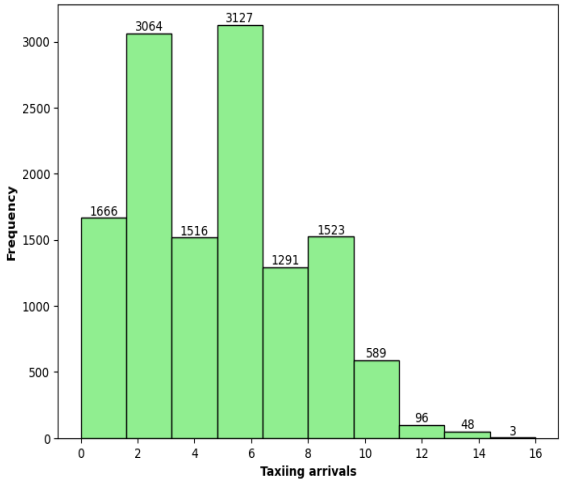
(a) Runway queue



(b) Delay previous hour



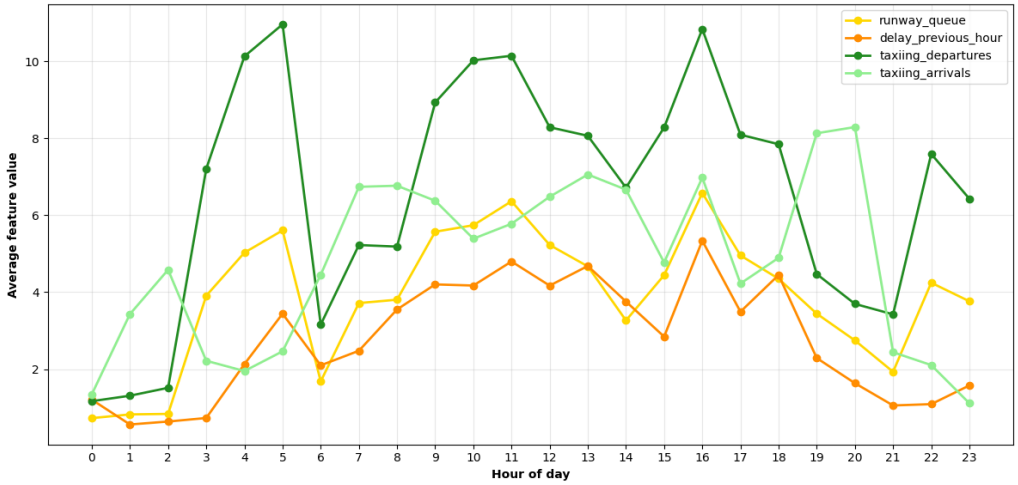
(c) Taxiing departures



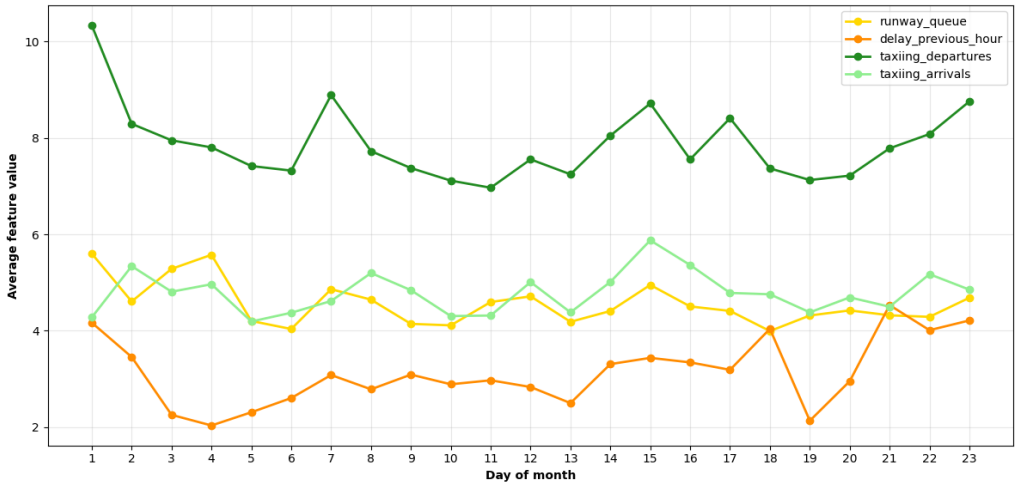
(d) Taxiing arrivals

Figure 3.5: Traffic and congestion features distribution of unique numerical values. The value on top of the bars denotes the number of observations that had the specific feature value.

To illustrate temporal fluctuations, the average feature value of all observations of the hour of day or the day of month is shown in Fig. 3.6. Taxiing departures and taxiing arrivals both have several peaks during the day, although the peaks do not always correspond to each other. Runway queue and delay previous hour follow the same trend on a hourly basis (Fig. 3.6a), but not necessary on a daily basis (Fig. 3.6b).



(a) Average feature values across hour of day



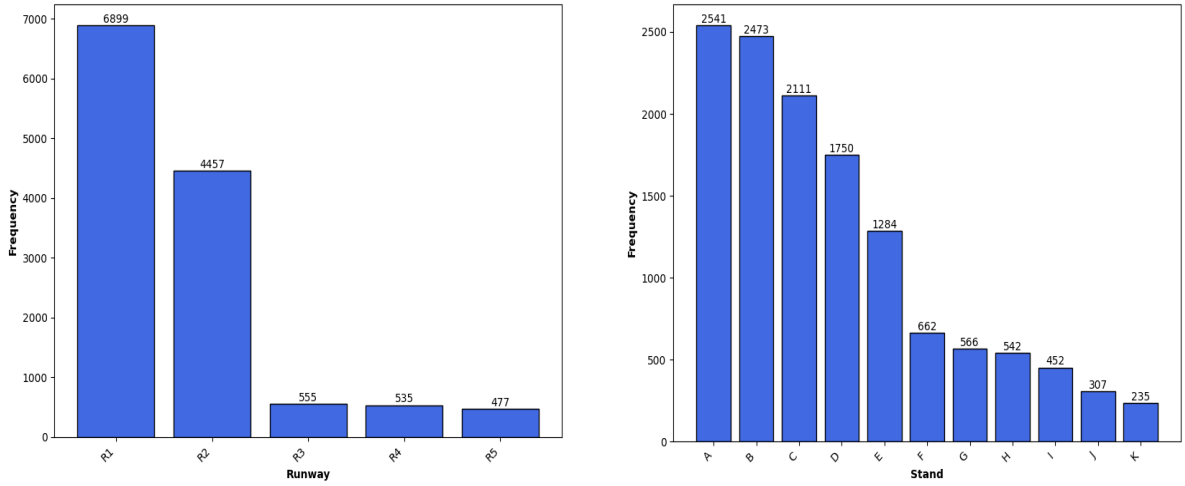
(b) Average feature values across day of month

Figure 3.6: Traffic and congestion features average values across day of month and hour of day

3.2.4 Spatial and infrastructure features

In the literature review, spatial features such as the distance the aircraft traveled during its taxi-out procession were found to be important for the prediction of taxi-out time. Unfortunately, distance data was absent from the 2022 data, and could not be constructed for the flight observations. To still capture information from the taxiing route, the features *stand* and *runway* were included, which is the start and end location for the aircraft before the takeoff. Stand and runway were included as required variables for the flights in the initial dataset (Fig. 3.2). Stand consists of one letter and a number. This means that there are multiple unique values, since there are many spots at the airport where an aircraft can park and start their taxiing from. To lower the number of unique values the stand are generalized to the first letter. The stand and runway distribution of their categorical values were inspected

(Figs. 3.7a and 3.7b).

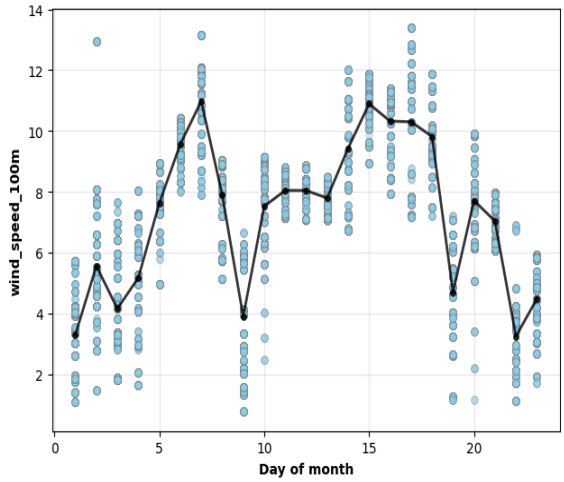


(a) Runway distribution **(b)** Stand distribution

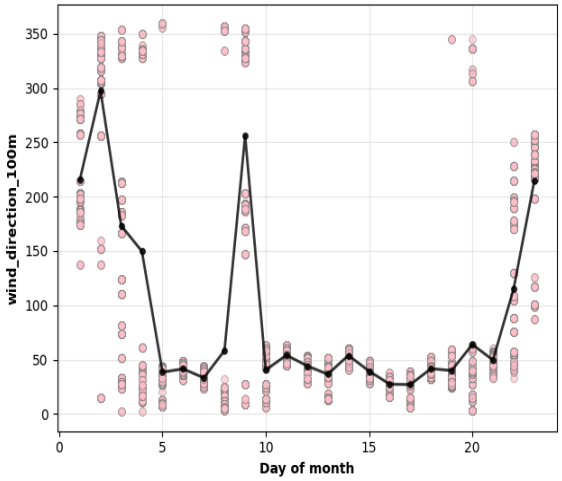
Figure 3.7: Departure flight observations runway and stand distribution

3.2.5 Weather

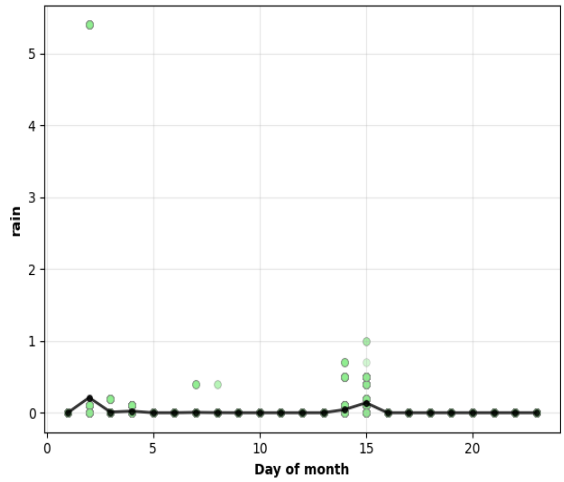
Although the original dataset contained headers for weather features, almost all flights had missing values in these cells. To include weather features, additional external data had to be integrated. This data was collected from Open-Meteo, an open-source software which offers historical meteorological data [32]. Since the Open-Meteo data is available at one hour resolution, the weather features appended to each observation were based on the aircraft's AOBT timestamp rounded to nearest hour. The selected weather features included wind speed, wind direction, rain and low level cloud coverage (Fig. 3.8). Wind speed and wind direction are measured at 100 meters above ground. Rain is defined as all rain that fell in the last hour, including brief local showers and rain from large scale systems. Low level cloud coverage includes clouds and fog up to 2 km altitude and measured in percent [32].



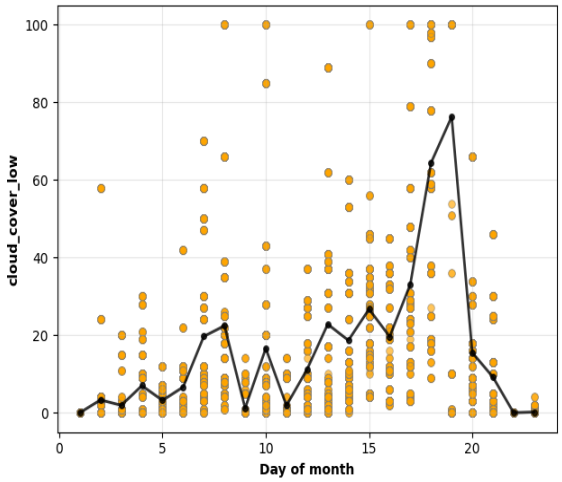
(a) Wind speed



(b) Wind direction



(c) Rain



(d) Cloud coverage

Figure 3.8: Observation distributions of weather features across day of month of the dataset. Black line indicate average of each day.

3.2.6 Aircraft characteristic features

The initial dataset (Fig. 3.2) included aircraft type, but it did not provide additional aircraft characteristics required for feature construction. To address this, the aircraft type was used as a key to map other more specific aircraft characteristics from ICAO codes using a data mapping file provided by Eurocontrol's BADA team. This mapping enabled the inclusion of several aircraft specific attributes, including engine type, number of engines, operating empty weight (OEW), wake turbulence category (WTC), maximum takeoff weight (MTOW) and descriptive properties such as length, wingspan, wingspan category and wing area (Tab. 3.5).

Table 3.5: Initial aircraft characteristics derived from ICAO codes using Eurocontrol's BADA mapping.

Aircraft attribute	**Description**	**Unit**
Engine type	Type of engine	Category (Jet or Turboprop)
Number of engines	Number of engines	Count (2,3 or 4)
OEW	Operating empty weight	Tons
WTC	Wake turbulence category	Category
MTOW	Maximum takeoff weight	Tons
Length	Length of the aircraft	Meters
Wingspan	Length between aircraft wing edges	Meters
Wing area	Area of each aircraft wing	Square meters

Among these attributes, WTC represents a critical operational constraint. Wake turbulence is a side-effect generated by lift-induced wingtip vortices [48]. To avoid hazards of wake turbulence effect, standardized separation minima between successive leading and trailing aircraft are enforced. This separation minima is currently defined based on aircraft weight categorization. WTC is an operational classification established by ICAO, based primarily on an aircrafts MTOW (Tab. 3.6) [49]. Aircrafts are separated into four distinct categories ranging from Light (L) to Super (J), which serve as the foundation of separation of aircrafts.

Table 3.6: ICAO Wake Turbulence Categories (WTC)

Wake Turbulence Categories (WTC)

Code	**Description**
J (Super)	Airbus A380-800 with MTOW of 560 ton.
H (Heavy)	MTOW $\geq$ 136 ton.
M (Medium)	MTOW between 7 ton and 136 ton.
L (Light)	MTOW $\leq$ 7 ton.

When aircraft depart from the same runway, these categories serve as safety buffers and dictate the maximum capacity of the runway system. Tab. 3.7 provides the distance-based and time-based separation minima between leading and trailing pairings as required by ICAO [48]. When a wake turbulence separation is not required between leading and trailing aircraft, the separation minima defaults to a minimum radar separation (MRS) of approximately 3 Nautical miles (NM) or 2.5 NM, or the calculated time-equivalent based on aircraft ground speed.

Table 3.7: ICAO Wake Turbulence and separation minima on departure

Distance-based separation minima on departure

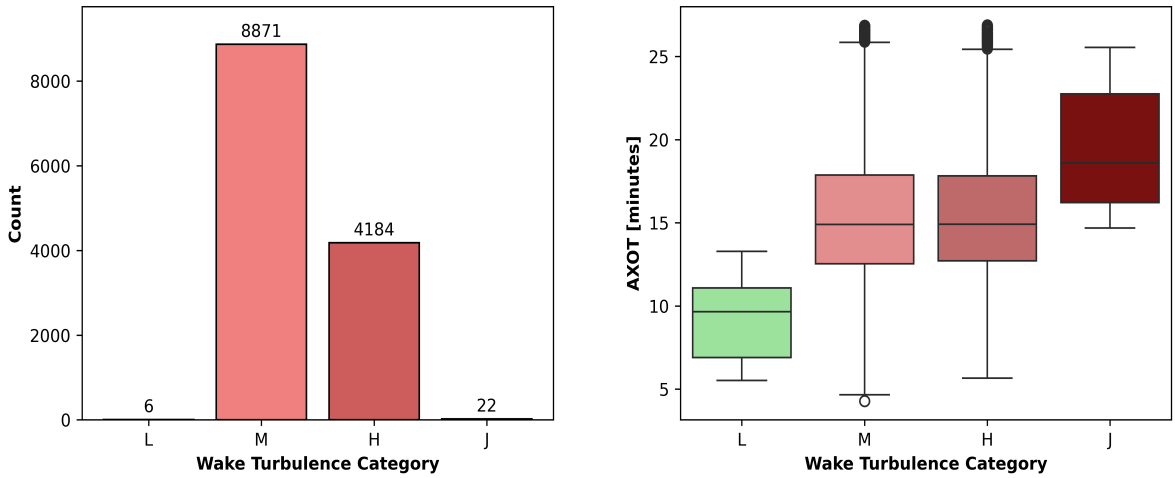
Leading/Trailing	J	H	M	L
J (Super)	MRS	6 NM	7 NM	8 NM
H (Heavy)	MRS	4 NM	5 NM	6 NM
M (Medium)	MRS	MRS	MRS	5 NM
L (Light)	MRS	MRS	MRS	MRS

Time-based separation minima on departure

Leading/Trailing	J	H	M	L
J (Super)	-	2 min	3 min	3 min
H (Heavy)	-	-	2 min	2 min
M (Medium)	-	-	-	2 min
L (Light)	-	-	-	-

As illustrated in Tab. 3.7 the separation depends entirely on the relative weight disparity between the leading and trailing aircrafts. For example, a Medium aircraft following a Super aircraft requires a 3 minute buffer, while homogeneous pairings such as a Medium following a Medium require no additional wake-related delay beyond standard radar separation.

Fig. 3.9 illustrates both the distribution of WTC in the dataset, as well as their relationship with the target variable AXOT.



(a) WTC distribution **(b)** WTC vs AXOT

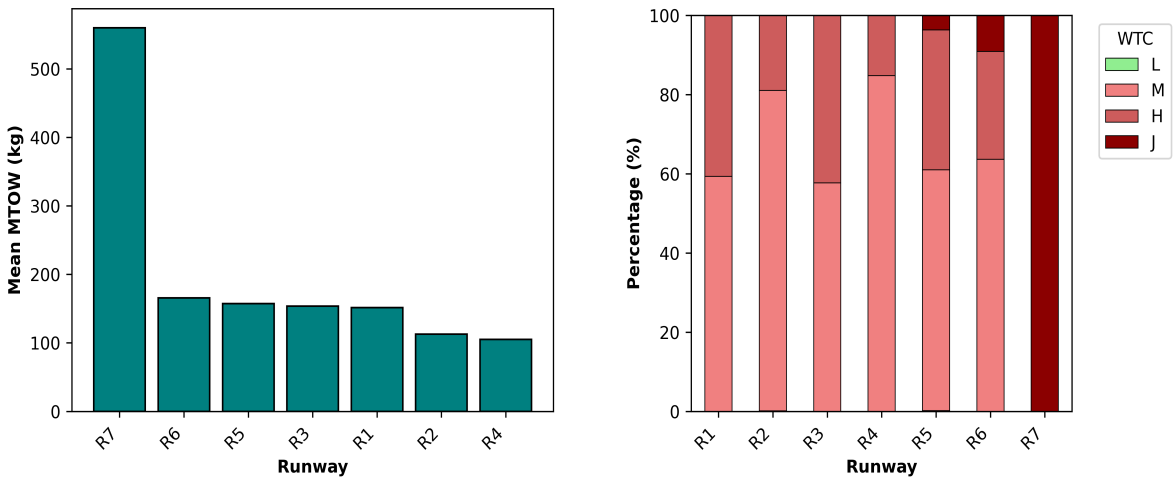
Figure 3.9: Aircraft geometry features plotted against the runways in the dataset.

The dataset overwhelmingly consists of aircraft in the categories M (8571 aircrafts) and H (4184) compared to L (6) and J (22). As such, the dataset is mostly comprised of aircraft within the medium weight range.

Fig. 3.9b confirms that the target variable AXOT has a direct relationship with these operational categories. Category L features the lowest median AXOT at approximately 10 minutes, whereas category J presents the highest at 18 minutes. Categories M and H display mid-range profiles with near-identical medians hovering around 15 minutes. These also exhibit the largest variance, indicating substantial operational diversity within these two brackets. The distribution for L is comparably tighter, between 5 and 11 minutes, without any upper-bound extreme outliers.

The presence of identical upper limits for outliers at around 27 minutes across M and H point towards a systematic or structural issue. Because these outliers persist, it indicates that the maximum AXOT values may be driven by other constraints rather than purely aircraft-specific wake characteristics.

To further investigate these structural dependencies, the distribution of MTOW and WTC were mapped against specific airport infrastructure elements. Fig. 3.10 illustrates the distribution of mean MTOW and the proportional breakdown of WTC classes across the runways available in the dataset.



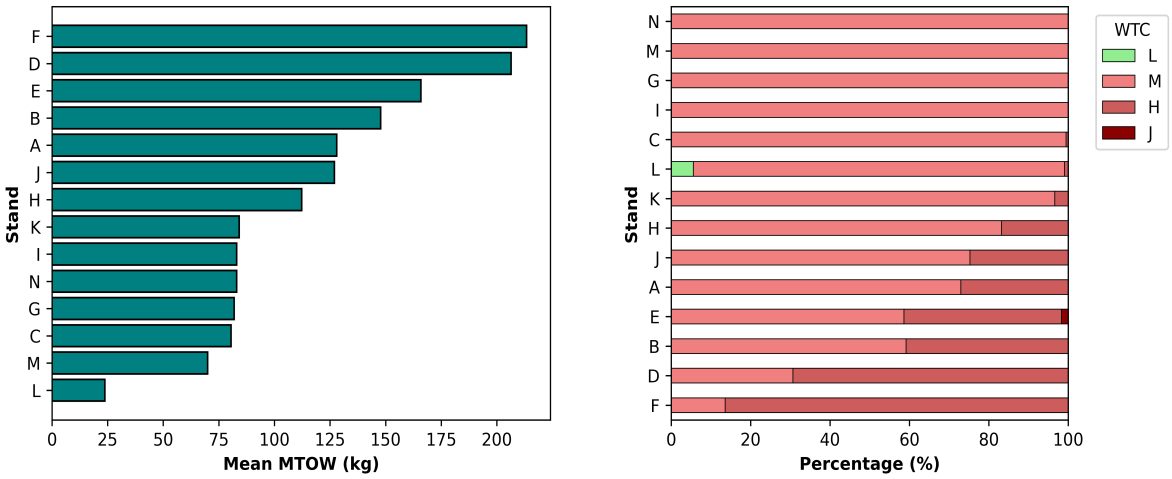
(a) Mean MTOW **(b)** WTC vs runway

Figure 3.10: Aircraft geometry features plotted against the runways in the dataset.

Fig. 3.10 illustrates a clear split of MTOW and WTC along the available runways. Runway R7 exclusively handles aircraft categorized as J. Consequently, flights assigned to R7 exhibit drastically higher mean MTOW. While R7 acts as a specialized threshold for the heaviest aircrafts, the data indicates that a smaller proportion of category J flights are also distributed across runways R5 and R6. While difficult to ascertain, category L flights are also exclusively distributed in one runway, R5. As such, R5 is the only runway which handles all operational categories.

A similar behavior is observable when analyzing MTOW and WTC for the stands in the dataset. Fig. 3.11 shows the distribution of flights along the airport's stand

groupings, with the mean MTOW and WTC.



(a) Mean MTOW **(b)** WTC vs stand

Figure 3.11: Aircraft geometry features plotted against the stands in the dataset.

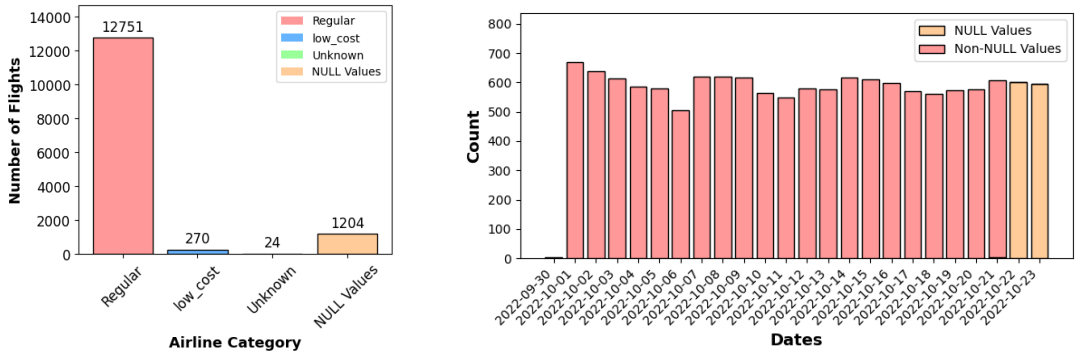
Fig. 3.11 indicates that there are physical infrastructure limitations regarding wingspan and ground maneuvering. Aircraft belonging to WTC J are exclusively restricted to stand E, indicating that this is the only stand equipped to handle that category of aircraft. Conversely, L is entirely isolated to stand D.

This strict separation implies that AXOT is heavily connected with runway and stand, through its relationship with WTC and MTOW. The high median AXOT in Fig. 3.9b observed for J could be explained by the combined effect of time-based wake separation intervals or the fixed physical taxi distance required to transit from stand E to runway R7.

3.2.7 Aircraft operational features

In addition to aircraft characteristics, an operational feature indicating whether an airline was a low-cost carrier was constructed, called *low-cost airline*. Previous studies have implemented this feature by classifying specific airlines as low-cost carriers [7]. This data was not included in the dataset and therefore a mapped and classified from an external source, based on an ICAO report identifying low-cost carriers [50]. To apply this classification to the dataset, airline ICAO codes were initially compiled from ICAO CORSIA page [51]. The compiled list was then matched against the ICAO low-cost carrier list to assign a binary indicator, where a value of 1 represents a low-cost airline and 0 otherwise. During this process, it was observed that a substantial number of airline codes in the dataset were not covered by the initial mapping. To improve coverage, the alphabetical registry from the Opennav online database was incorporated [52]. The sources were then combined to create a comprehensive mapping between airline identifiers and ICAO codes.

In cases where airline codes in the dataset still do not match with any from the constructed mapping, additional manual verification is performed using ICAO and aviation code databases [53, 54]. The process results in a mapping file containing 5449 airline ICAO codes, of which 243 are classified as low-cost. This mapping file is then used in the pipeline to categorize whether a flight in the database is a low-cost airline or not, and add the resulting feature to the dataset for use in the predictive model (Fig. 3.12a). While the majority of the data consists of non-low-cost airlines, this is expected for a major airport. However, 8.4% of the data had missing values, indicating that the airline type was not categorized in the dataset. All the missing values originated from the last two available days in the data (Fig. 3.12b). Given that they represented a substantial amount of flights, these entries cannot simply be dropped, as it would decrease the size of the dataset significantly. While these could be categorized as not being low-cost due to lack of information, this impacts the generalizability of the feature. As such, the feature is dropped due to missing data.



(a) Airlines categories. **(b)** Airline distribution by date.

Figure 3.12: Departure flight observations low-cost airline distribution.

Additionally, the *engine type* was not considered a feature, as only 18 outliers exist that have a different category than Jet, specifically Turboprop. Given that the airport is a large hub primarily serving bigger aircraft, the turboprop flights, which constitute only 0.1% of the total dataset, were considered to be outliers and constituently dropped from the dataset. As a result the engine type feature was dropped from the dataset.

Similarly, the *number of engines* feature was dropped due to the imbalance of the distribution of the data. 98.8% of the flights in the dataset had 2 engines, and only 1.1% and 0.1% of flights had 4 and 3 engines, respectively.

3.3 Feature selection with multicollinearity analysis

Multicollinearity can negatively impact the stability and interpretability of both LR and SHAP-based feature importance estimates. Therefore, a multicollinearity analysis was conducted to identify and remove redundant predictors.

3.3.1 Numerical feature redundancy

A correlation analysis for the *numerical features* in the dataset was performed using a correlation matrix. While correlation does not imply causation, it is a critical metric for identifying multicollinearity. Fig. 3.13 reveals clusters of redundancy, particularly among the aircraft geometry features, MTOW, Length, OEW, wing span and wing area, with near perfect correlation scores ($r > 0.96$).

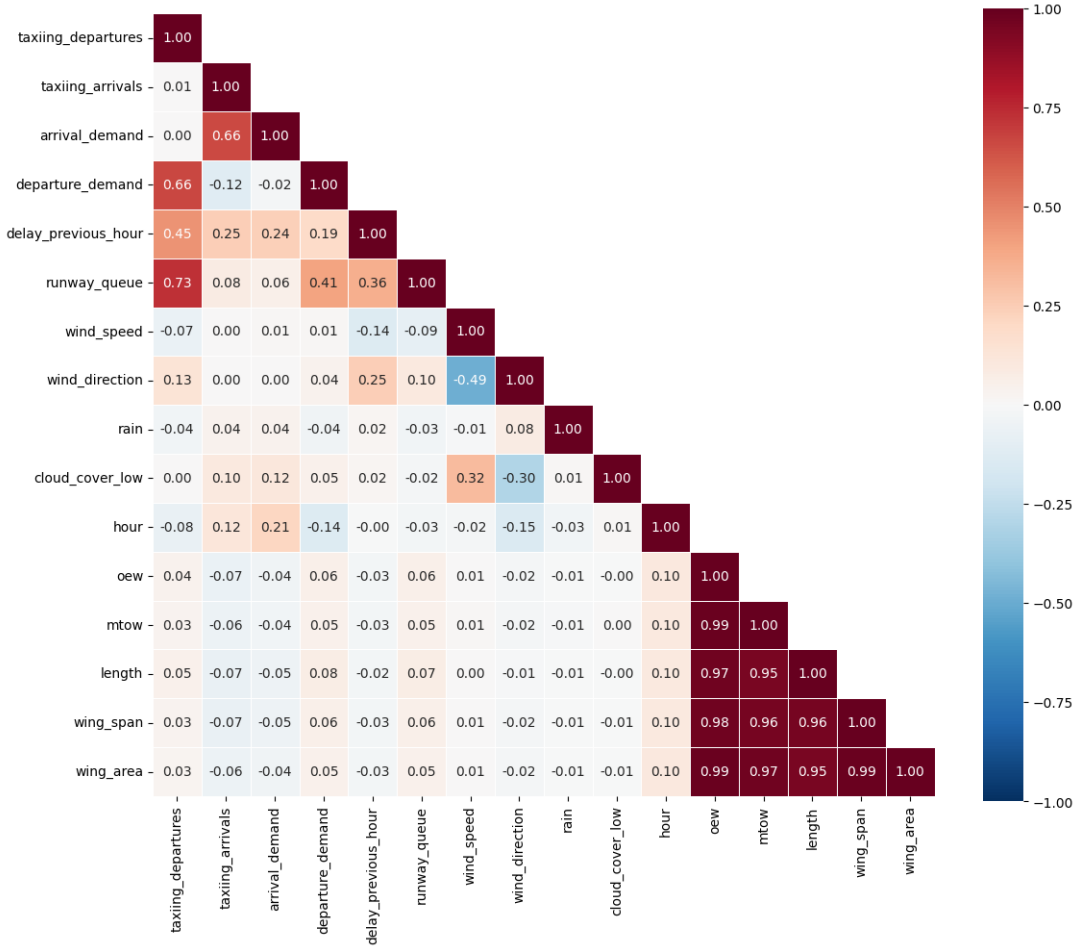


Figure 3.13: Correlation matrix of all numerical features in the dataset. The values range from -1 to 1, indicating the strength and direction of the linear relationship between variables.

While the correlation matrix effectively identified explicit pairwise relationships, it

cannot detect multi-way linear dependencies. To quantify systematic multicollinearity, the VIF was calculated. A VIF value of above five means that features should be handled carefully, but a VIF value exceeding 10 signals severe multicollinearity which needs remediation. The VIF for the numerical features are presented in Tab. 3.8.

Table 3.8: Table of calculated Variance Inflation Factor (VIF) for the numerical features in the dataset, containing all numerical features with a VIF > 5.

Feature	**VIF**
wing_span	648.36
length	438.57
oew	415.46
wing_area	404.45
mtow	126.13
arrival_demand	21.72
departure_demand	20.22
taxiing_departures	17.68
wind_speed	9.27
taxiing_arrivals	7.43
runway_queue	6.67
hour	5.47

The aircraft geometry features, wing span, length, OEW, wing area and MTOW all show extreme multicollinearity (VIF > 100). In order to mitigate the multicollinearity, features were pruned using an iterative VIF elimination process. In each iteration, the feature displaying the highest individual VIF score was dropped, and the VIF matrix was then recomputed for the remaining numerical features - until a maximum threshold of 20 was reached. Through this iterative process, the features wingspan, OEW, wing area and length were dropped from the dataset.

Interestingly, removing these features revealed a secondary group of highly correlated features. These consisted of departure demand, arrival demand, taxiing departures and taxiing arrivals. Because these features retained VIF values between 10 and 19, these were also pruned. The feature pair with the highest VIF, departure demand and arrival demand, were dropped from the dataset.

Following the completion of these removals, the feature taxiing departures exhibited the highest VIF of 10.55. While this value borders on the threshold, this feature was kept due to operational domain logic. As a global metric quantifying the airports overall surface congestion, taxiing departures contains vital signals that complement the taxiing arrivals feature. It was therefore intentionally retained.

3.3.2 Categorical feature redundancy

Dependencies and associations between the *categorical features* was evaluated using Cramer's V. The Cramer's V coefficient is bounded between 0 and 1, where 1 means there is a perfect association and 0 means there is no association. A value above 0.5 is considered to signal strong association.

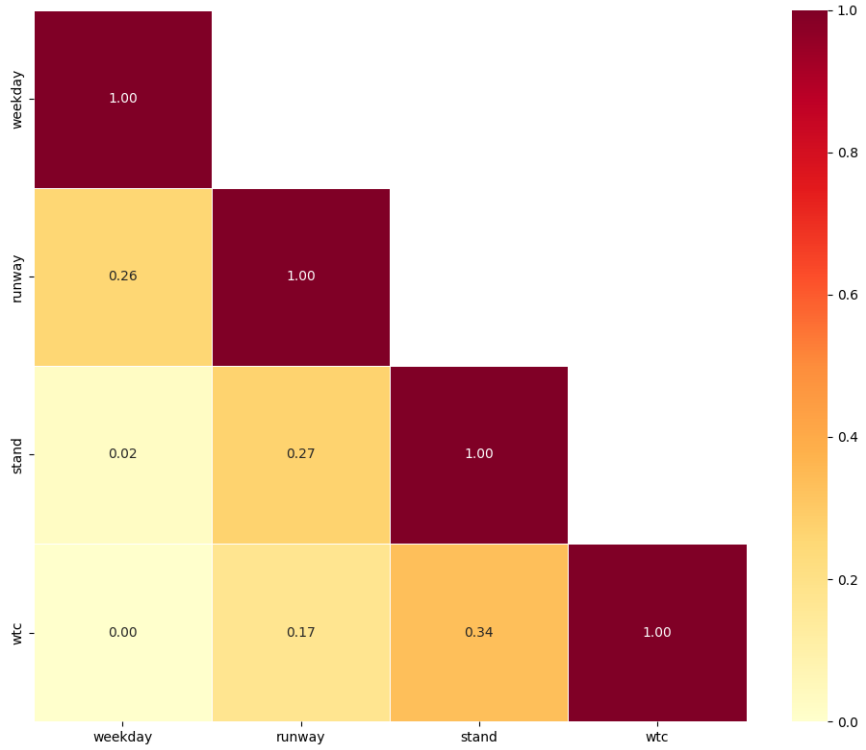


Figure 3.14: Cramer's V matrix of the categorical features. The values range from 1 to 0.

As shown in the resulting matrix, Fig. 3.14, the categorical features in the dataset displayed weak mutual associations, indicating no requirement to drop features. As such, all categorical features were retained and kept in the dataset.

3.3.3 Cross-type redundancy

Because standard statistical matrices are bounded by data types, a manual and domain-specific cross examination was executed to identify redundancy between the categorical and numerical features. A conceptual overlap exists between the numerical aircraft geometry feature MTOW and the categorical feature WTC, since WTC is explicitly derived from MTOW.

Retaining both features would risk introducing severe cross-type information redundancy. From previous exploration, (Fig. 3.10a) revealed that MTOW and WTC follow similar trends. However, as a continuous metric, MTOW preserves high-resolution physical variance, whereas WTC simplifies this variance into a few discrete

bins. To avoid losing data resolution and keep the feature with the most amount of information, WTC was dropped from the final dataset in favor of MTOW.

A complete summary of the outcomes from the multicollinearity phase is formalized in Tab. 3.9.

Table 3.9: Features affected from multicollinearity analysis

(a) Redundant dropped features

Feature	Method
arrival demand	VIF
departure demand	VIF
length	VIF
oew	VIF
wing area	VIF
wing span	VIF
WTC	Manual cross-type

(b) Overlapped kept features

Feature	Method
MTOW	VIF
taxiing departures	VIF
taxiing arrivals	VIF

3.4 Machine learning models and evaluation

Based on the findings of previous taxi-out prediction studies, LR and RF were selected as models for evaluating the predictive performance of the proposed feature set. Prior to model training, categorical variables are transformed using one-hot encoding in order to represent them as 'numerical' input features, which means each categorical variable becomes multiple binary columns, where each column contains either 0 or 1. This allows categorical attributes such as aircraft type to be incorporated into the machine learning models. The machine learning models and preprocessing methods in this study were implemented in Python using the open source Scikit-learn library (sklearn), including LR, RF, Randomized Search for hyperparameter tuning, and StandardScaler for feature scaling [55].

3.4.1 Linear regression

To predict taxi-out time, a LR model is chosen as the baseline model to use on the created dataset. The purpose of selecting LR is to establish a simple and interpretable benchmark performance against which more advanced models, such as RF, can later be compared. The target variable AXOT is separated from the features. Missing values in the features are imputed with zeros. To ensure that the features contribute equally to the model, they are scaled using StandardScaler. This process standardize the features to have a mean of zero and a standard deviation of one, which is helpful when interpreting the feature importance based on the magnitude of the coefficients. The dataset is then split into training and testing sets using a 80/20 split. The LR model is trained on the scaled training data, and its performance is evaluated using RMSE, R^2 score and MAE.

3.4.2 Random forest

RF algorithm is set up by using the RandomForestRegressor, where the initial model was implemented with the default values. To achieve a better performance by the model, hyperparameter tuning was implemented. The parameters that were used in the optimization is shown in Tab. 3.10. The optimizer method is called RandomizedSearchCV and randomly samples parameter combinations from a search space, which is fast and efficient for larger parameter spaces. Cross-validation was integrated into the hyperparameter tuning process through the cv parameter in RandomizedSearchCV. Specifically, 5-fold cross-validation was used, where the dataset was partitioned into 5 equal parts. For each parameter combination tested, the model was trained on 4 parts and validated on the remaining part, repeating this process 5 times so that each part served as the validation set once. This cross-validation approach ensures that the hyperparameter selection is robust and not dependent on a particular train/validation split, providing a more reliable estimate of model performance and preventing overfitting to any single data partition. RF full configuration are shown in App. A.1.

Table 3.10: RF hyperparameters fed to the optimizer and final values chosen.

Parameter	Description	Final value
n_estimators	number of trees in the forest	200
max_depth	maximum depth of each tree	40
max_features	maximum features considered for each split	0.3
min_samples_leaf	minimum number of observations required in each leaf.	2
min_samples_split	minimum number of observations to allow a new branch.	5

3.4.3 K-fold cross validation

To further evaluate RF model performance, the train-test split was switched out in favor of k-fold cross validation. The dataset was partitioned into 10 folds of approximately equal size. For each iteration, nine folds were used for training and one fold was held-out for validation. This process was then repeated until each fold had served as the validation fold once. Performance metrics MSE, RMSE and R^2 were computed for each fold.

The performance metrics of the RF model were then aggregated over the folds, in order to obtain an average estimate of the model performance.

3.4.4 Feature importance using SHAP

For each test sample in each fold in k-fold cross validation, the SHAP values (one per feature), the base value (expected value), the feature values for that sample, the model prediction, and the actual value were saved. All of this information was then

stored in a concatenated dataframe.

After the k-fold cross-validation, the SHAP plots were generated. The global SHAP plots were created by concatenating all out-of-fold SHAP values and their corresponding feature data. The distribution and importance of the SHAP values for each feature across all test samples were then visualized.

For the local plots, a particular row of interest was selected, the largest overestimate and the largest underestimate. A SHAP Explanation object was then created for that row using its saved SHAP values, base value, and feature data from the dataframe.

4
Results

This chapter presents the results achieved through the procedures defined in the methods chapter. First, the predictive performance of the initial baseline comparison between multiple LR and RF is presented. Then, the evaluation metrics of the robust k-fold cross validation approach applied to the final RF model are shown. Lastly, the findings from the post-hoc SHAP explainability framework for feature importance is presented.

4.1 Initial evaluation

The initial evaluation established a performance baseline by comparing the predictive accuracy and error distributions of the LR model against the RF model.

4.1.1 Performance metrics

The performance metrics for the initial evaluation, MAE, RMSE and R^2, for both models are presented in Fig. 4.1. The average magnitude of errors, represented by both MAE and RMSE, was systematically higher for the baseline LR model than for the RF model across both the training and the test set. However, the RF model shows a distinct performance discrepancy between the training and the test sets. It demonstrates significantly higher predictive accuracy on the training data than on the unseen test data.

Through the R^2 score, the performance gap of the LR model and the RF model grows. The baseline LR model shows a R^2 score of 28% for the training set, and 32.4% for the test set. In contrast, the R^2 score for RF model for the training set demonstrates near-optimal performance, although the score for the test set is lower. But despite this clear reduction in generalization capabilities on unseen data, the RF model's test R^2 score consistently outperforms the baseline LR model.

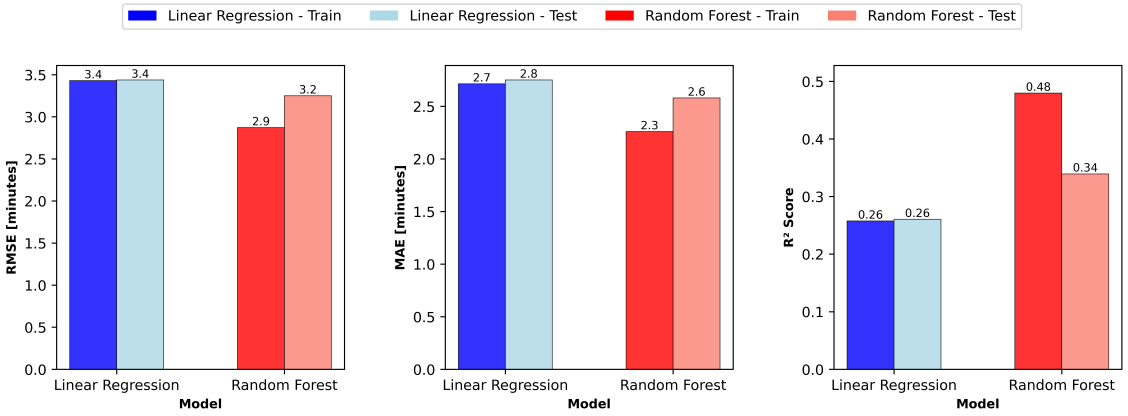


Figure 4.1: Performance metrics for the initial evaluation of the models multiple LR and RF models.

4.1.2 Actual VS predicted

Figs. 4.2 and 4.3 illustrates the distribution of actual versus predicted AXOT values for both the LR model and the RF model across their respective training and test subsets. The dashed line embedded in each of the plots represents the ideal $y = x$ line. Observations falling closer to this diagonal line indicate higher prediction accuracy. For the training dataset, the prediction cloud of the LR model exhibited significantly greater variance and scattering than that of the RF model. The tighter alignment of the dots along the line for the RF model indicates that it achieves better accuracy than the LR model (Fig. 4.3).

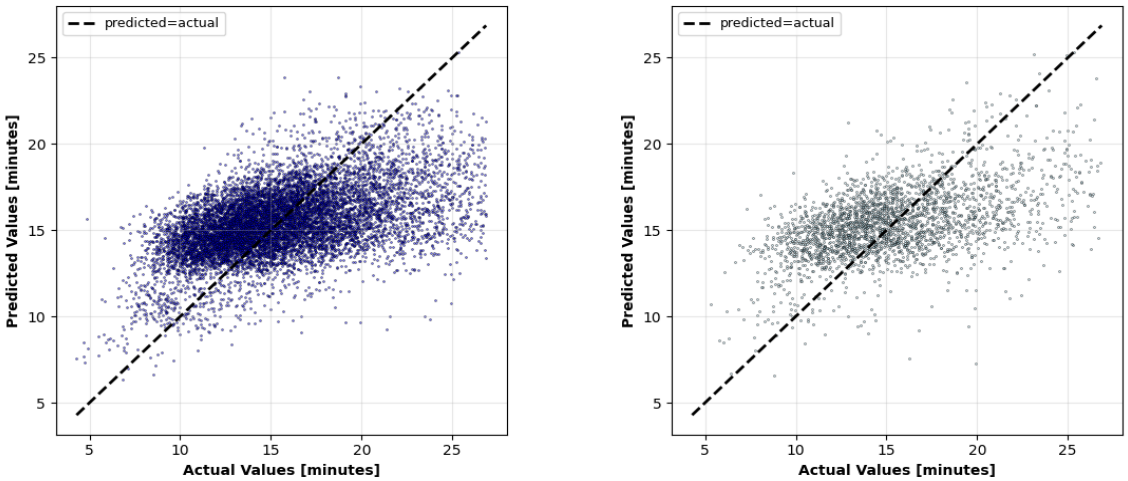


Figure 4.2: Distribution of actual versus predicted AXOT values for LR model.

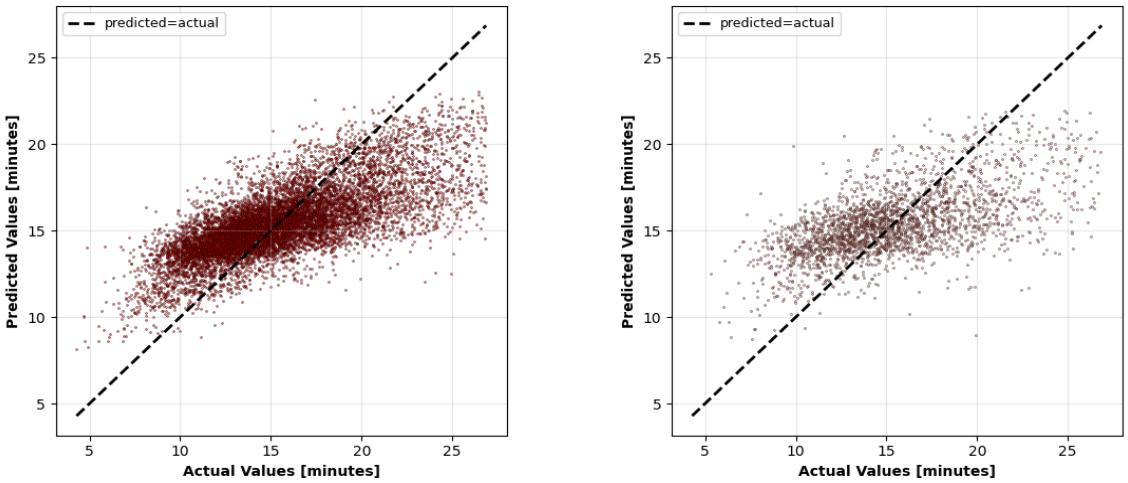


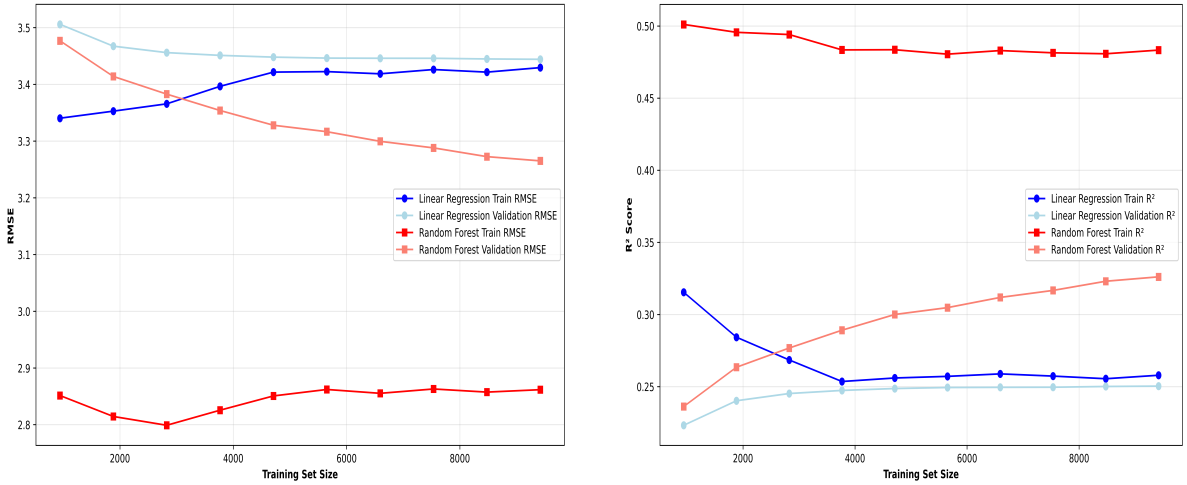
Figure 4.3: Distribution of actual versus predicted AXOT values for RF model.

Additionally, the models exhibit some shared behaviors which persists across all four subplots. When the actual AXOT is low, both models tend to overestimate their predictions, causing the data points to cluster above the dashed line. Conversely, when the actual AXOT is high, both models underestimate the time, so the points instead cluster below the dashed line.

Finally, the spread of the predictions change based on the actual value. For lower AXOT, the predictions stay within a narrow range, but as the AXOT increases, the predictions become significantly more spread out and unpredictable.

4.1.3 Learning curves

The learning curves for both models are presented in Fig. 4.4. Fig. 4.4a shows the RMSE while Fig. 4.4b shows R^2. Both the train and test curves for the LR model display high RMSE scores, lying close to each other with values around 3.4 to 3.5, as seen in Fig. 4.4a. As the size of the training set increases, the training error rises while the test error decreases, causing the curves to converge around a score of 3.4.



(a) Learning curves using RMSE.

(b) Learning curves using R^2.

Figure 4.4: Learning curves showing the LR model and the RF model performance as the training set size increases. Square markers indicate the RF model, while dot markers indicate the baseline LR model.

In Fig. 4.4b, the curves for LR display low R^2 scores. The training curve starts at a higher value and steadily decreases, whereas the test curve starts lower and gradually increases, with both curves converging around 0.10. Overall, the RMSE remains high and R^2 remains low for both the training and test sets.
On the other hand, for the RF model, the training curve remains stable around an RMSE of 2.8 to 2.9, while the test curve starts at 3.5 and decreases as the set size increases. For the R^2 scores, the training curve is consistently high with a score of approximately 0.5 and does not increase with an increased set size. The test curve starts below 0.25 but increases steadily as the training set size grows, leveling out near 0.34. The resulting high test error alongside low training error, combined with high training R^2 and lower test R^2, indicates a persistent gap between the two sets.

4.2 Validation Results

This section presents the performance metrics of the final RF model evaluated across 10 cross-validation folds, followed by an analysis of the model stability across the individual splits. Additionally, the results of the SHAP analysis is presented.

4.2.1 Performance metrics

The averaged cross-validation metrics RMSE, MAE and R^2 are presented as box-plots in Fig. 4.5. Across all three metrics, the training performance consistantly outperforms the test performance. The model achieves a stable training R^2 of approximately 0.79, while the corresponding test R^2 scores decrease to a range between 0.3 and 0.4.

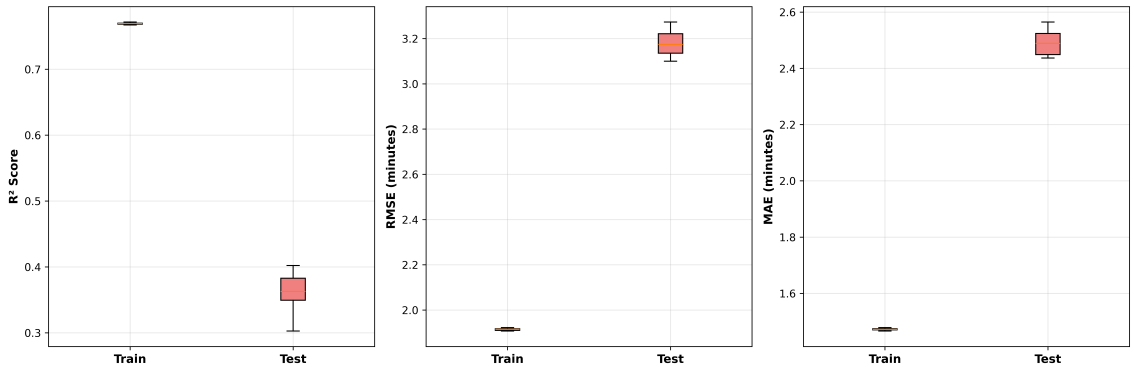


Figure 4.5: Boxplots showing the distribution of performance metrics RMSE, MAE and R^2 across the 10 cross-validation folds.

The error metrics RMSE and MAE show that the training set errors remain low, averaging approximately 1.8 minutes for RMSE and 1.4 minutes for MAE. In comparison, the test errors increased to a range of 3.1 to 3.3 minutes for RMSE and 2.4 to 2.6 minutes for MAE.

Interestingly, the boxplots reveal differences in variance between the splits. The training metric distributions are extremely narrow, indicating high consistency across the folds. Conversely, the spread of the test boxplots are wider, particularly for R^2. This variation suggests that the models performance might be more sensitive to the specific subset the data was tested on, though the boxplots remain free of severe outliers.

In order to evaluate the stability across the splits, Fig. 4.6 displays the performance metrics explicitly for each of the ten individual folds.

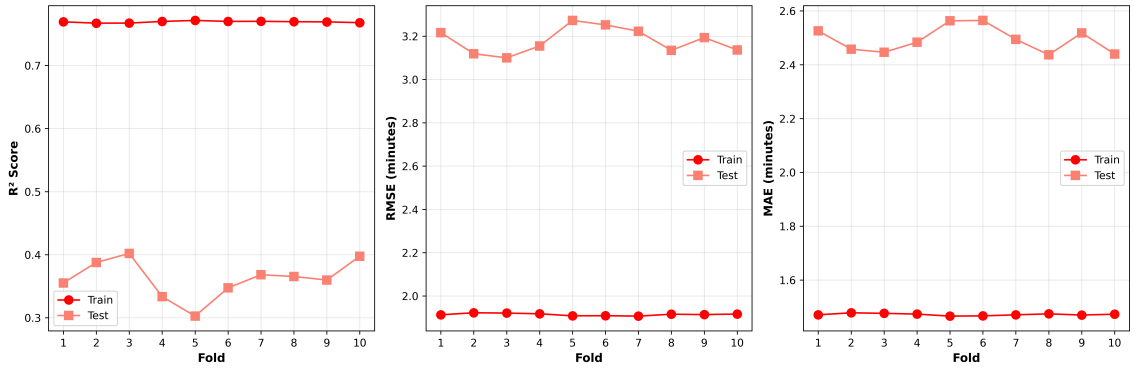


Figure 4.6: Line plot visualizing the difference in training and test metrics across the 10 cross-validation folds.

The lines corresponding to the training metrics remain completely flat across all intervals. The training R^2 stays consistently near 0.79, while the training MAE hovers around 1.4 minutes, and the training RMSE at around 1.8 minutes. This validates the tight consistancy observed in the boxplots in Fig. 4.5.

In contrast, the test metrics display noticeable variability over the folds. Fold 5 sticks out as the lowest performing fold, reaching the lowest R^2 score at 0.3 alongside the highest recorded RMSE and MAE.

4.2.2 Feature importance

This section presents the results of the evaluation of the feature contributions within the RF model to determine which variables drive the prediction of AXOT. SHAP is used as a post-hoc diagnostic tool. Specifically, SHAP provides both global and local explainability metrics for investigation of which specific features the model relies upon to generate its predictions.

4.2.2.1 Global SHAP

Global SHAP was utilized to determine the overall distribution and magnitude of feature influence across the entire set. By calculating the average absolute SHAP values for each variable, the model's feature dependencies become a standardized metric of global importance. This can be visualized using different plots.

Fig. 4.7 shows the resulting bar plot for the RF model. The bar plot ignores whether the impact of a feature was positive or negative. It quantifies, on average, the magnitude of each feature's contribution toward the predicted AXOT. It gives a clear hierarchy, showing which features have the highest impact, by presenting them in descending order.

The bar plot identifies runway R3 as the feature with the highest global impact, with a mean SHAP value of 0.96 minutes. Runway queue is the second highest ranking feature, with a mean SHAP value of 0.7 minutes. These are followed by taxiing departures, runway R2, stand F and wind direction. Hour and mtow display identical importance scores of 0.18 minutes. However, although the bar plots ranking established global importance, it does not account for the direction of the impact.

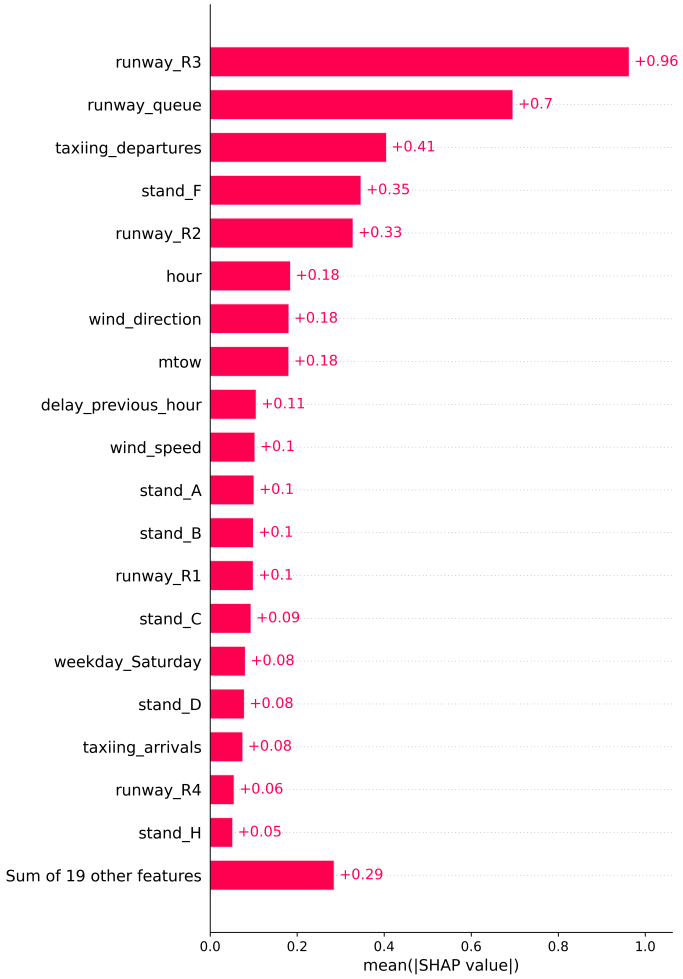


Figure 4.7: SHAP Bar plot showing the mean absolute SHAP values for each variable. Features are ranked by their influence on the predicted AXOT (in minutes).

Fig. 4.8 shows the SHAP beeswarm plot, which complements the bar plot by illustrating why the highest-ranked features are important. The x-axis represents the SHAP value, where positive values indicate an increase in the predicted outcome and negative values indicate a decrease. The colors represent the actual value of the feature. Specifically, red indicates a high feature value, and blue indicates a low feature value. The vertical thickness indicates the density of individual data points at that SHAP value. In this one-hot encoded dataset, red values and blue values correspond directly to the presence or absence of specific categories.

The top feature, runway R3, exerts the strongest overall influence on model predictions. As a categorical feature, red signifies that the flight utilized runway R3, while blue signifies its absence. The red dots for R3 are shifted far to the right, indicating that utilizing this runway increased the predicted AXOT by up to 4 minutes relative to the baseline. The blue dots clustered around zero suggests that not using R3 did not push the prediction down.

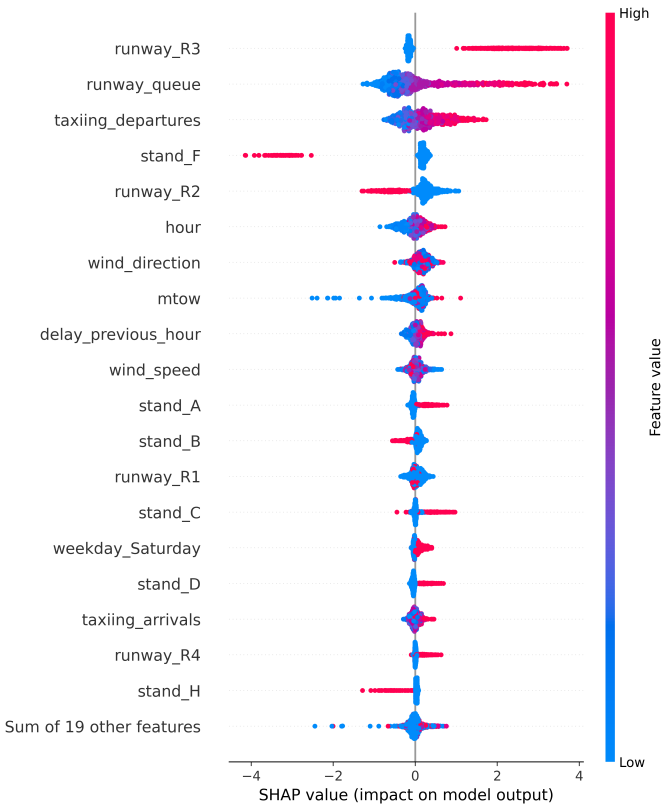


Figure 4.8: SHAP beeswarm plot ranked by mean absolute SHAP value. Individual data points are distributed horizontally according to their SHAP value, with vertical stacking indicating data density.

The second highest ranked feature is runway queue, which shows a clear transition from blue to red, indicating a linear-like positive correlation. As the queue size increases, the SHAP value increases by up to 4 minutes. A similar trend is observed for taxiing departures, which is the third ranked feature. It also shows a clear transition from blue to red, although its maximum SHAP value does not exceed 1 minute. Additionally, the density is thicker at around zero, indicating greater variance in how this efature affects individual flights.

Conversely, there are some features that are shown to be strong negative drivers. For stand F, the red dots are clustered to the far left, at approximately -4 SHAP value. As such, stand F has a massive dampening effect on the AXOT, by up to 4 minutes. A similar, albeit not as drastic, effect on AXOT can be observed when flights utilize runway R2.

For the weather features, wind direction ranks the highest. The values cluster just to the right of zero, with the middle of the cluster seeming more blue, and the outskirts on the left and right of the clusters showing as more red. This suggests a non-linear or more complex relationship, where the high or low values does not have a simple one-way impact.

For the hour feature, the red dots show a slight positive skew, capturing a potential

peak-hour effect. Additionally, the weekday saturday feature shows a consistent, minor positive shift, indicating slightly higher AXOT values on weekends.

Finally, MTOW shows a long tail to the left for blue dots. This suggests that very light aircraft might reduce AXOT, while heavier aircraft shift marginally to the right of zero, increasing the AXOT by roughly 1 minute while still remaining close to the average.

4.2.2.2 Local SHAP

While global SHAP allows for a holistic view of the models logic, local SHAP allows for the decomposition of individual predictions. This approach proves specifically useful for analyzing special cases and outliers, such as instances of significant model error. By analyzing which features drives these outliers, it is possible to identify specific operational conditions where the model's assumptions cannot capture reality.

Two SHAP waterfall plots were generated to illustrate instances where model predicted extreme positive and negative errors. In waterfall plots, $E[f(x)]$ denotes the baseline expected value (the dataset mean), while $f(x)$ represents the final predicted value for that specific observation. The numerical values displayed in gray text to the left of the feature names indicate the actual, raw value of that specific feature for the individual prediction being analyzed.

Fig. 4.9 shows the SHAP waterfall plot for the model's largest overestimation of AXOT. The baseline expected value was 15 minutes, and the model predicted a value of 17 minutes.

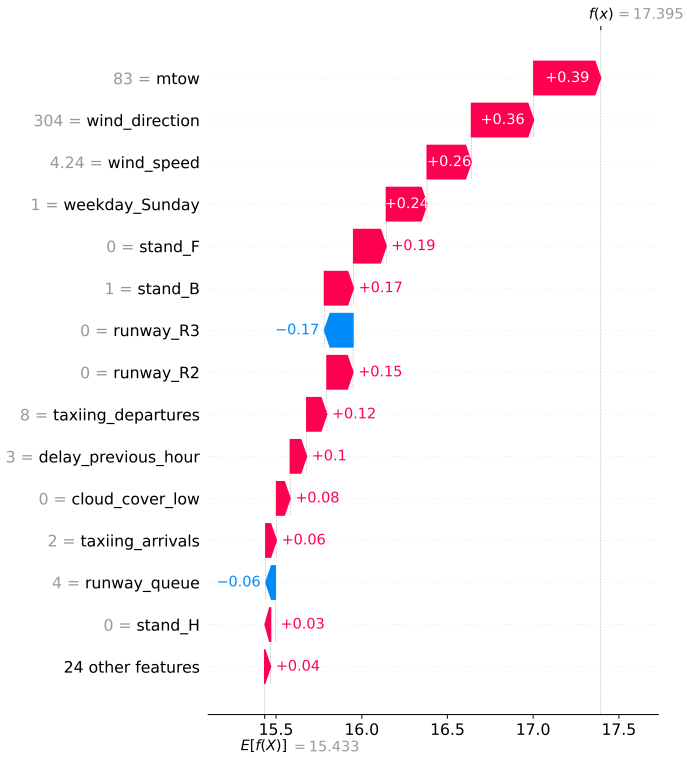


Figure 4.9: Waterfall plot for the prediction with the largest overestimate. Actual AXOT: 8.06 min, Predicted AXOT: 17.395 min (Error: 9.328 min).

Fig. 4.9 reveals that MTOW, with a value of 83 tons, and wind direction, with a value of 304, were the most contributing factors to the model's prediction. They pushed the predicted AXOT up with a value of 0.39 and 0.36 respectively. Furthermore, wind speed also increased the prediction with 0.26 minutes, and having a feature value of 4.24. Other features that had an increasing impact on AXOT was weekday Sunday, stand F and stand B. The two features that had a decreasing impact were runway R3 and runway queue, which is interesting. From the bar plot (Fig. 4.7), it could be seen that these two had the highest importance score. MTOW had a feature value of 83, which corresponds to 83 tons. According to Tab. 3.6, this is categorized as a medium (M) flight.

Fig. 4.10 displays the SHAP waterfall plot of the model's largest underestimation. The baseline expected value for the model was 15 minutes, whereas the actual model prediction was 12 minutes.

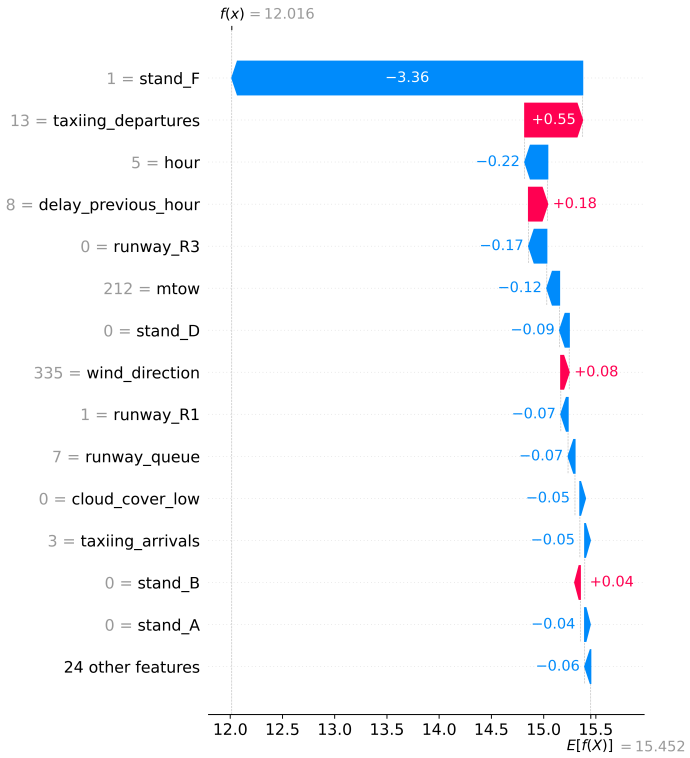


Figure 4.10: Waterfall plot for the prediction with the largest underestimate. Actual AXOT: 24.50 min, Predicted AXOT: 12.00 min (Error: 12.48 min).

The largest contributor to this downward shift is the feature stand F, with a feature value of 1, which reduced the AXOT prediction by 3.36 minutes. Other, albeit much smaller, negative contributions include hour with a feature value of 5 and a SHAP value of 0.22, and runway R3 with SHAP value of 0.17 and a feature value of 0 - indicating that it was not used, and MTOW with a value of 212 tons and a negative effect of 0.12. Notably, Tab. 3.6 would categorize this flight as heavy, since it is more than 136 ton, but not specifically 560 tons which would indicate an airbus A380-800. The second highest contributing feature, taxiing departures, push the AXOT value up by 0.55. This reflects the delay previous hour feature, which also pushes the value up by 0.18. Interestingly, runway queue (consisting of only 7 flights) reduced AXOT for this prediction, with a value of 0.07.

5
Discussion

This chapter discusses the results of the data harmonization, feature engineering and machine learning processes outlined in the previous chapters of the report.

5.1 Data harmonization and limitations

The success of any machine learning model heavily relies on the quality of the data used for training and validation. Despite the effort invested in data harmonization and feature construction, several limitations in the dataset used for predicting AXOT remain. These limitations may impact the model's performance and generalisability.

5.1.1 Temporal imbalances and data coverage

One of the primary limitations was the availability and coverage of the data. The initial dataset contained only a single recorded day of data from 2019, and 23 days of recorded data from 2022. The disparity between the number of flights available for the years was large, which risked introducing bias into the dataset. There were also many inconsistencies identified between the data available for 2019 and 2022 during the temporal analysis. While the 2019 data contained 17 tables which could be connected to a flights timeline through its ids, the 2022 data only contained 3 tables. This discrepancy was due to the lack of event based data tables in 2022.

Furthermore, while the 2019 data contained more information such as runway configuration, weather and data for distance features, it only consisted of 643 flights from a single day. This was an insufficient sample size to capture the variance of those features safely, even though they were available. Retaining it would risk introducing noise or bias into the model for the specific day or the conditions of it. For example, a comparative analysis of airport congestion data between 2019 and 2022 revealed clear structural anomalies, with certain runways exhibiting entirely different congestion patterns across these two periods. Further investigation confirmed that the airport underwent a physical expansion in 2020, where a new runway was built and added, completely changing the physical runway and stand configurations between these years.

Consequently, the physical routes and taxiing constraints of 2019 were fundamentally different from those of 2022. Using data from both of these years would have introduced severe structural bias, as the model would attempt to learn spatial logic

on two different airport layouts.

5.1.2 Schema table differences

In addition to these limitations, challenges related to temporal consistency, data interpretation and semantic meaning of the tables were identified during the harmonization process. Analysis revealed that different timestamp definitions were used across tables pertaining to different years. For example, in one table consisting of only off-block data from 2019, both an 'actual' timestamp and an 'occurred' timestamp for the off-block time were available. These timestamps exhibited a consistent offset of approximately three minutes across all available data in 2019.

When comparing these two timestamps in the off-block event table with the summary tables, it was the 'actual' timestamp that seemed to correspond to the time that was presented as 'aobt' in the summary table. However, the temporal analysis showed that all other tables simply had 'occurred' timestamps, not 'actual'. It remains unclear whether this difference reflects system recording delays during specific years, or distinct operational definitions of the off-block event, which was not used for the other event tables.

One important consideration during the analysis was the fundamental architectural difference in the data present between between 2019 and 2022. Timeline analysis during the data exploration phase revealed that the 2019 dataset utilized dynamic, event-driven logs. These contained columns such as 'occurred'. As a plane moved through the airport, three general summary tables were activated one hour prior to the flight and updated one hour after the flight. During the flight, continuous, live updates were pushed directly to the event tables. As such, the 2019 data contained high-frequency, live updates of the flight's real-time progress.

In contrast, the 2022 data did not contain the same real-time event driven updates. Instead, the 2022 architecture consisted only of the static summary tables. If the 2022 data recordings would have contained event driven logs and detailed timestamps during the taxiing phase, this could have enabled more possibilities when applying machine learning. For example, using the data to simulate the airport as a live system, where the real-time updates would act as new inputs for the model. This would enable the making of iterative predictions.

Attempting to align features across these two time periods would have risked introducing structural imbalance. The summary table for 2019 contained structural inconsistencies with the event based data tables, and no event data was present in 2022. Since there was no documentation on how the real-time tracking architecture was mapped to the summary tables, or why there was a change in the data between 2019 and 2022, there was no way of resolving or explaining the inconsistencies. As such, combining these two time periods was unfeasible. However, the 2019 data was dropped from dataset in favor of the 2022 data. The choice maintained the integrity of the dataset.

5.2 Feature engineering

Many of the features identified in the literature review (Tab. 3.4) were able to be successfully reconstructed using the airport data. Where the airport data lacked information, features were also able to be reconstructed due to the ability of using key ids to supplement the airport data with external sources. However, some key predictive features such as the airport infrastructure features could not be reconstructed through the airport data or the external sources.

5.2.1 Feature construction

The process of recreating the features identified in the literature review, shown in Tab. 3.4, proved to be challenging. Some features, such as the temporal and congestion features, could be recreated from the airport data. However, most of the features could only be reconstructed using external sources, and some could not be reconstructed even through external sources.

The information needed for spatial and infrastructure was explicitly tied to the 2019 data. Considering the differences in airport infrastructure and runway configuration between the two years, using 2019 to construct features such as cumulative turning angle or distance would be detrimental to the model. Effort was made to create these features by supplementing information from external sources, but little to no detailed layout information was available on the airport. There was not enough information to reconstruct features, and as such those features were excluded despite the literature review stating that they were key predictors of taxi-out time.

Despite the successful creation of a feature, it was still not a guarantee that it could be used in the dataset for predictive modeling. During feature analysis, the reconstructed features 'number of engines' and 'budget airline', for example, both showed extremely one-sided distributions. For the budget airline in particular, the two last days of the dataset were even missing (Fig. 3.12a). Since ML models learn patterns, including scewed features risk only introducing noise to the model. As such, these features were dropped from the dataset, despite effort being put into reconstructing them.

Supplementing features through external sources was a time consuming process. Manually mapping the ICAO-identified low-cost airlines through their Corsia webpage, or relying on Eurocontrol to provide excel files on aircraft information, is not an optimal solution. If this information was already available through the airport data, more time and effort could be spent on other key tasks, such as improving the predictive capacity of the models.

The feature analysis also had the added benefit of allowing for insights into feature relationships. Aircraft geometric properties such as MTOW, wingspan and WTC, showed explicit spatial relationships with the features runway and stand. For instance, in Fig. 3.10b, it was possible to see that runway R7 was utilized exclusively

by the largest aircraft class (J) in WTC. The mean MTOW for flights departing from R7 was also drastically higher than the average values observed across all other runways (Fig. 3.10a).

When analyzing these physical properties against the target variable AXOT in Fig. 3.9b, heavy aircraft categories such as J and H displayed a substantially higher mean AXOT compared to lighter aircraft categories such as L. As the physical size and weight of the aircraft decreased, the average AXOT decreased as well. From an ATM perspective, this behavior is expected. The operational constraints are explicitly tied to separation safety standards. Because heavy aircrafts generate powerful wake vortices, air traffic controllers must enforce wider time buffers before allowing trailing aircraft to depart. If there are runways such as R6 which handles both heavier aircrafts and medium aircrafts, some could face longer ground delays as they wait for their required safety windows to clear. This could be why runway R7 exclusively is used for heavier aircrafts, to avoid the buffer times on the other runways.

This delay could also be exacerbated by spatial routing constraints. Within the dataset, every single aircraft belonging to the heaviest classifications such as J under WTC originates exclusively from stand E (Fig. 3.11b). This pattern strongly implies that stand E represents the only physical terminal gate able to accommodate the biggest aircrafts. If stand E is located at a large physical distance from runways R6 and R7, the only two runways executing heavy departures, the inflated AXOT observed for large aircrafts might act as a direct proxy for taxiing distance.

5.2.2 Multicollinearity and relations

The multicollinearity analysis was designed to be *model-agnostic*, that is to not depend on any specific machine learning model. This was motivated primarily by the need to protect the stability of the SHAP feature importance rankings. When multiple features carry identical information, SHAP splits the importance scores arbitrarily across them, which dilutes the values and obscures the true impact of the critical drivers. However, the specific methods used to achieve this might have introduced a bias.

Because Pearson's correlation coefficients and VIF rely on identifying linear relationships, pruning features based on these metrics inadvertently might have biased the final dataset in favor of the LR model. Conversely, Cramer's V, which is non-parametric and does not assume linearity, flagged no severe dependencies among the categorical features. This contrast could indicate that the linear diagnostics might have forced a specific structure onto the data. However, it is also worth noting that no clear feature groups were present in the categorical feature set in the same way that the aircraft geometry and surface congestion features were in the numerical feature set.

Regardless, the multicollinearity analysis revealed several critical insights. Ex-

ploratory analysis through the correlation matrix (Fig. 3.13), confirmed that the continuous aircraft geometry features suffered from severe multicollinearity (> 0.9). The VIF showed that these geometric features yielded values exceeding 100, with the wingspan variable peaking at an extreme score of 600 (Tab. 3.8). This was not completely unexpected, as the aircraft geometry features were all constructed by mapping aircraft types directly to Eurocontrol's BADA dataset.

A similar multicollinearity issue was identified among the surface congestion metrics. This matched the feature exploration stage, where the curves for the congestion features over time aligned. While it is intuitive that these features track the same core operation strain, the severity of the VIF (>20) was notable. After pruning the pair of congestion features with the highest combined VIF, arrival demand and departure demand, one of the remaining taxiing features retained a high VIF score of 10.5. Although this still slightly exceeded the traditional critical threshold of 10, the variable was intentionally retained in the final feature set to maintain the pairwise symmetry and operational logic of the congestion tracking system.

Furthermore, attempting to compare the categorical and numerical features against one another proved difficult, as there are no clear, purely statistical methods for directly comparing them against each other. For this reason, non-data driven domain-specific reasoning was used to compare the remaining features. Specifically, a choice had to be made between MTOW and WTC. This introduced an academic trade-off regarding the inclusion of operational categories versus raw numerical features.

This difficulty was heightened by the model-agnostic approach. In hindsight, arguments could be made for dropping WTC in favor of MTOW due to the RF models weakness with high-cardinality categorical variables. Additionally, SHAP feature importance has no clean way to handle one-hot encoded features. However, because a model-agnostic pipeline was used and the model-specific evaluation step occurred later, this was not a justifiable reason for dropping a feature at this early stage.

On the other hand, ATM ground controllers do not calculate takeoff separation using continuous weight values. Instead, they make decisions based on operational categories like WTC. Furthermore, because the exact splitting thresholds chosen by a RF model are deeply embedded within its tree structures, relying solely on continuous features like MTOW might obscure the model's inner logic and lower its overall interpretability.

This trade-off between model complexity and true domain interpretability is further illustrated by the baseline LR model. While it suffered from severe underfitting because it lacked the mathematical flexibility to map the highly interactive, non-linear reality of the airport, it offered completely transparent coefficients.

Ultimately, attempting to use statistical tools such as correlation matrices, VIF, and Cramer's V to drop features creates a potential trade-off. It forces a choice between cutting out similar features to achieve clean, independent interpretability for each

variable, or keeping highly collinear features to potentially maximize a non-linear model like RF's predictive performance.

5.2.3 Feature importance

To gain deeper insights into the internal decision making of the predictive models and investigate which features actively contributed to the predictive power, SHAP was applied to the RF model.

5.2.3.1 Global SHAP

Global SHAP was able to give explanations for the overall model behavior. It showed that runway R3 yielded the highest mean SHAP value (Fig. 4.7), closely followed by the congestion features runway queue and taxiing departures, and stand F and runway R2. Global SHAP indicates that the most critical global drivers of AXOT are a balanced combination of infrastructure and surface congestion features.

In the literature review, it was clear that distance and variables explaining the geometry of the airport had been seen to have a high impact on taxi-out prediction. As such, it is not surprising that the infrastructure feature runway R3 had the highest predictive impact at 0.96, and stand F at 0.35 as the 4th. Looking only at a standard bar plot, it would be easy to conclude that these two simply represent the runway-stand pair that adds the most time to an aircraft's AXOT.

However, the beeswarm plot shown in Fig. 4.8 introduces nuance not captured in the bar plot. The beeswarm plot shows that while utilizing runway R3 actively increases the predicted AXOT, not using it has almost zero impact on the baseline. For stand F, the relationship is entirely inverted, with its presence actually decreasing the predicted AXOT. Instead, runway R2 is the infrastructure element associated with shorter taxi times, while stand A pushes predictions upward. This contrast highlights the value in using XAI tools such as SHAP, particularly for safety critical areas such as ATM, where understanding the directional impact of a feature is just as critical as knowing its overall rank.

The conflicting behaviors of stand F and runway R3 highlight a complex trade-off between physical airport layout and operational flow. It strongly suggests that the model could benefit from more continuous distance features to better connect specific stand locations to their assigned runways, rather than treating them as isolated variables.

The high rankings of runway queue (0.70 minutes) and taxiing departures (0.41 minutes) also align with the literature review. These features directly capture surface delays. A high volume of taxiing departures naturally creates congestion, because aircraft must often share taxiways or cross active paths. Similarly, a long runway queue represents the final bottleneck stage where aircraft might have to wait to take off. The beeswarm plot perfectly illustrates this behavior by illustrating the high

congestion values shift predictions to the right (longer AXOT), while low values cluster to the left (shorter AXOT).

The intermediate features such as hour, wind direction, and MTOW all shared an identical global importance score of 0.18 minutes, representing secondary temporal, weather, and aircraft characteristics. The lower ranking of MTOW could indicate that using static variables often fails to capture the fluid, stochastic nature of surface traffic.

Despite this lower rank, MTOW is still highly relevant. The beeswarm plot shows that low MTOW values, indicating lighter aircrafts, have a strong effect in shortening AXOT, whereas high MTOW values that signify heaver aircrafts exert an upwards pull on the prediction.

Similarly, hour and wind direction act as secondary drivers. The beeswarm plot shows that low hour values, corresponding to late nights or early mornings, consistently shorten AXOT, while higher values increase it, successfully capturing a peak-hour congestion effect. For wind direction, the dots were spread across both positive and negative SHAP regions. This is likely because specific wind vectors dictate active runway configurations, which in turn alter the physical taxi distances.

Finally, the use of one-hot encoding for high-cardinality categorical features, like stands and runways, significantly expanded the dimensionality of the feature space. This expansion runs the risk of introducing data sparsity and noise, which can distort individual SHAP values. While the SHAP framework remained robust, the lack of real-time runway configuration changes due to the data constraints means the feature space is missing key operational signals. If exact physical distance metrics or other critical features could have been reconstructed, they would have eliminated the need to encode individual stands and runways altogether, drastically lowering the model's cardinality and noise.

5.2.3.2 Local SHAP

Local SHAP investigates one single prediction, and explains how the model made that specific prediction. The most interesting cases to look at, is the 'extreme' cases, where the predicted value was the most far away from the actual value. Therefore the largest overestimate, when the model predicted a value that was higher than the true value, and the largest underestimate, when predicted a value that was lower than the true value, was analysed.

One consistent finding across both extreme cases was the behaviour of the stand F feature. In the underestimation case (Fig. 4.10), the presence of the aircraft at stand F drastically reduces the AXOT prediction by 3.36 minutes. Conversely, in the overestimation scenario (Fig. 4.9, the aircraft not being at stand F pushes the prediction up by 0.19 minutes. These consistent patterns indicate that the model associates stand F with faster taxi times.

Additionally, there are also clear differences between the overestimate case and the underestimate case. The overestimation error seems to be cumulative, as no single feature dominates the prediction. Instead, a combination of multiple minor factors such as MTOW at 0.39 minutes, wind direction with 0.36 minutes, and wind speed with 0.26 minutes, all aggregate to drive the predicted AXOT upward. In contrast, the underestimation error is highly localized, driven almost entirely by the critical downward impact of stand F (-3.36 minutes), while other features contribute negligibly.

Interestingly, the directional impact of MTOW in both plots yields a highly counter-intuitive result when compared the trends from the analysis of the dataset (Fig. 3.9b), which indicated that heavier aircraft categories (H and J) experience, on average, higher AXOTs than lighter categories. Yet, the local SHAP plots contradict this. The heavy aircraft (212 tons) decreases the AXOT by 0.12 minutes, while the lighter medium aircraft (83 tons) acts as the primary driver increasing the AXOT by 0.39 minutes.

Looking up the 212-ton outlier in the reference dataset from BADA resolved this paradox, as the specific airframe corresponds to a dedicated cargo freight aircraft. Freight operations often bypass passenger related terminal delays, explaining the negative SHAP contribution despite the high MTOW. This highlights a critical blind spot in the current feature space, the model cannot differentiate between passenger aircraft and cargo aircraft. Consequently, to fully capture the operational realities of the airport, incorporating a binary feature classifying aircraft as either freight or passenger could be considered.

Finally, these outliers illustrate the complex nature of taxi-time prediction, which requires a delicate balance of both 'static' and 'dynamic' features. While static features like infrastructure stand, runway, and physical attributes such as MTOW establish the baseline operational constraints, dynamic features like weather and traffic congestion incorporates more real time information. The local SHAP analysis demonstrates that model failures not only stem from a single feature, but that they can also fail from the cumulation of these static and dynamic operational conditions.

5.3 Model performance and generalizability

The third research question concerned whether machine learning could be applied on the created dataset for taxi-out time prediction, and how the performance was. The initial evaluation with both models demonstrated that there were only small differences between the RMSE and MAE performance metrics for both models. Both models had RMSE of approximately 3 minutes, and LR's MAE was close to 2.8 minutes while RF's was closer to 2.5. RF showed a bigger difference in the coefficient of determination R^2, which means that RF variability is greater and explains the predicted outcome better.

Both models showed a tendency to predict higher AXOT values when the ac-

tual values were low and lower AXOT values when the actual values were high (Figs. 4.2, 4.3). This could be due to the LR model assumption constant variance of errors which can smooth out the extreme predictions. RF on the other hand builds many decision trees where each makes their own predictions. When the trees disagree on extreme observations, the averaging process tends to pull predictions towards the center of the distribution. As a result, RF exhibits a similar pattern of overpredicting low values and underpredicting high values.

Another common observation is that the prediction spread increases for higher AXOT values, whereas the range of predictions is narrower for lower actual AXOT values. One possible explanation is that lower AXOT values correspond to aircraft that taxi relatively quickly to the runway, resulting in more predictable conditions. In contrast, for higher AXOT values, more time has elapsed between prediction and departure, increasing the likelihood that the operational conditions have changed. Consequently, the relationship between the predictor variables and the target variable becomes less stable, which can reduce prediction accuracy for both models.

On the other hand, a difference was demonstrated by the learning curves, between how the LR model and the RF model responded to the engineered dataset. The baseline LR model suffered from structural underfitting, proving the model was insufficient at handling the complex dependencies of airport surface, which is modeled as non linear relations. The RF regressor showed inverse behavior, demonstrating a pronounced tendency to overfit the training data.

Learning theory suggests that overfitting can be mitigated by increasing the amount of training data. For the LR model, the learning curves indicate that this limit was reached at a training set size of approximately 4 000 observations, where the training and validation curves converge, indicating the model has reached it's full potential (Figs. 4.4a, 4.4b). The learning curves for RF suggest a similar pattern, since the curves are getting closer to each other as the training set size is increasing. The improvement in RF curves lies in the validation scores improving for both R^2 and RMSE metrics, while the training score stays more consistent when train set increases.

However, an important question remains: why does the gap between the RF learning curves persist despite the increase in training data? This persistent gap indicates that the overfitting might not only be a function of sample size constraints, but also a reflection of architectural and encoding mismatches. One possible explanation is the presence of high-cardinality categorical variables encoded using one-hot encoding. The runway and stand identifiers resulted in a high-dimensional feature space, which can be challenging for RF models. During the recursive splitting process, the model may learn patterns that are specific to individual runway or stand configurations, rather than capturing more general relationships, such as the effect of taxi distance on AXOT. Consequently, the model may overfit to localized patterns in the training data, limiting its ability to generalize to unseen observations.

The prediction of AXOT is based on the operational state observed at the off-block event, when the aircraft departs from the stand. As a result, the model is inherently limited in its ability to account for unexpected events that occur after taxiing has commenced. Delays caused by factors such as changing runway queues, traffic congestion, or operational disruptions may not be fully captured by the predictor variables available at the time of prediction. Features such as runway queue length, delay during the previous hour, taxiing arrivals, and taxiing departures give a representation of current airport conditions and may indirectly reflect current traffic patterns. Nevertheless, predicting taxi-out time remains challenging when unusual events occur after the prediction is made.

This limitation becomes more pronounced as taxi-out time increases. The longer the time interval between prediction and take-off, the greater the opportunity for airport conditions to change. Consequently, predictions associated with high AXOT values are subject to greater uncertainty than those associated with shorter taxi-out times. This behavior may explain the wider spread of predictions observed for larger actual AXOT values in both the LR and RF models (Figs. 4.2, 4.3).

An alternative modeling approach would be to define the target variable as the remaining taxi-out time and continuously update predictions as new information becomes available. Such a framework would allow the model to adapt dynamically to changing airport conditions throughout the taxi phase. However, implementing this approach would require additional data collected during taxiing, such as updated traffic information or the aircraft's current position.

The choice between a static AXOT prediction and a dynamic remaining-time prediction depends on the intended application. A static AXOT prediction may be more suitable for strategic planning purposes, such as departure sequencing and resource allocation before taxiing begins. Furthermore, this approach is easier to implement because it requires only information available at the off-block event. In contrast, a dynamic remaining-time prediction may be more valuable for real-time operational decision-making, as continuously updated estimates could support congestion management, runway sequencing, and delay mitigation in response to evolving airport conditions. Regardless, the data limitations encountered in this study meant that developing such a dynamic prediction framework would not be possible.

6
Conclusion

This thesis investigated how airport surface operational data could be harmonized and used for ML-based taxi-out time prediction. The study addressed three research questions concerning data harmonization, feature selection, and ML-based predictive modeling.

The first research question examined how airport surface data can be harmonized for ML applications. The results showed that constructing a ML-ready dataset required extensive preprocessing due to inconsistencies in data structure, timestamp definitions, and table availability across years. Despite these challenges, a dataset was successfully created by combining airport operational data with external sources containing aircraft, airline, and weather information. However, several potentially important features, such as taxi route distances and detailed runway configuration data, could not be reconstructed even through external sources due to missing information and limitations in the available data sources. This thesis therefore highlights that data harmonization is a critical and often overlooked step in airport-related ML applications.

The second research question concerned which features should be selected for taxi-out time prediction. The results demonstrated the importance of combining domain knowledge with statistical analysis during feature engineering and feature selection. Multicollinearity analysis identified severe redundancy among several aircraft geometry variables and some congestion related features. Removing these redundant variables improved the interpretability of the feature set while preserving relevant operational information. The feature analysis also revealed that airport infrastructure features, aircraft characteristics, and congestion metrics were closely connected to the taxi-out time. Furthermore, the study showed that feature selection should not only consider predictive performance but also the impact of feature redundancy on explainability methods such as SHAP.

The third research question investigated whether ML could be applied to the created dataset and how well the resulting models performed. The results demonstrated that ML could successfully be applied to the available airport surface data for taxi out time prediction. The baseline LR model showed clear underfitting and was unable to capture the complex non-linear relationships present in the dataset. In contrast, the RF model outperformed the LR model, by modeling these non-linear interactions. However, the Random Forest model also showed signs of overfitting. To improve the RF models generalization, incorporating additional training data and alterna-

tive feature encoding strategies are suggested.

To improve interpretability, SHAP was applied to the RF model. The global SHAP analysis revealed that airport infrastructure and congestion-related variables were among the most influential predictors of taxi-out time. The local SHAP analysis provided additional insight into individual predictions and prediction errors, highlighting both the strengths and limitations of the model. In particular, the analysis identified operational patterns that could not be fully captured by the available feature set, such as differences between heavy passenger aircraft and cargo aircraft.

Overall, the findings demonstrate that airport surface data can be transformed into a ML-ready dataset and successfully applied to taxi-out time prediction. The study further shows that data harmonization, feature engineering, model selection, and explainability techniques all play important roles in developing reliable predictive models for airport operations. Beyond the predictive results themselves, the work provides practical insights into the challenges of preparing airport operational data and contributes a reproducible framework for future ATM-related machine learning research.

6.1 Future Work

Several opportunities for further research have been identified through this thesis.

Importantly, future work should focus on improving data availability, as well as the data harmonization process. A major limitation of this thesis was the absence of consistent airport infrastructure information, such as taxi route distances, runway crossing counts, and runway configuration data. Since infrastructure related features were consistently identified as important predictors during the literature review, access to more detailed airport operational data might significantly improve both model performance and interpretability. This is particularly relevant given that the proxy categorical variables for stands and runways remained among the most influential features in the SHAP analysis. Furthermore, establishing a standardized harmonization framework for airport data would support more robust and reproducible ML research within airport surface operations.

Additional feature engineering approaches should also be explored. The multicollinearity analysis in this thesis relied on methods designed to identify linear dependencies between variables. Future work could investigate alternative approaches capable of capturing more complex non-linear relationships among features. Furthermore, the local SHAP analysis revealed that the model was unable to distinguish between heavy passenger aircraft and heavy cargo aircraft despite their different operational characteristics. Incorporating additional features describing aircraft operation type, together with more continuous infrastructure-related variables, may allow future models to better capture the operational factors driving taxi-out time.

Future research should also investigate alternative feature encoding strategies. The

RF model showed signs of overfitting, which might be explained by the use of one-hot encoding for high-cardinality categorical variables such as stands and runways. Methods such as target encoding or frequency encoding could preserve the predictive information contained within these features while reducing the size of the feature space.

Another promising thing to explore is the development of dynamic prediction models. The current approach predicts taxi-out time using information available at the off-block event, providing a static estimate before taxiing begins. However, the airport surface conditions evolve continuously during the taxi phase. This is especially true for longer taxi-out times, the information on why the time is long might not have been present at the off-block time. Incorporating real-time updates could enable rolling predictions of remaining taxi time and provide more adaptive decision support for ATC.

Finally, future studies could evaluate a broader range of ML models to improve the predictive performance. While RF outperformed LR in this study, other approaches such as regularized linear models or XGBoost might offer improved predictive performance and generalization. Comparing multiple modeling approaches on larger datasets would provide a more comprehensive understanding of their suitability for airport surface operational prediction tasks.

Bibliography

[1] Adobe Stock. (2026) Air traffic control tower at night with sunset view. Stock photo, accessed Feb, 2026. [Online]. Available: https://stock.adobe.com/

[2] A. C. I. World and I. C. A. Organization, "Joint aci world–icao passenger traffic report, trends, and outlook," Montreal, Canada, January 2025, accessed: 2026-02-18. [Online]. Available: https://aci.aero/2025/01/28/joint-aci-world-icao-passenger-traffic-report-trends-and-outlook/

[3] Z. Du, J. Wu, Y. Leng, and S. Wandelt, "Ai4atm: A review on how artificial intelligence paves the way towards autonomous air traffic management," *Journal of the Air Transport Research Society*, vol. 5, p. 100077, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2941198X25000211

[4] J. Yin, Y. Ma, Y. Hu, K. Han, S. Yin, and H. Xie, "Delay, throughput and emission tradeoffs in airport runway scheduling with uncertainty considerations," *Networks and Spatial Economics*, vol. 21, no. 1, pp. 85–122, 2021. [Online]. Available: https://doi.org/10.1007/s11067-020-09508-3

[5] J. Bao, J. Kang, J. Zhang, Z. Zhang, and J. Han, "A dynamic control method for airport ground movement optimization considering adaptive traffic situation and data-driven conflict priority," *Journal of Air Transport Management*, vol. 124, p. 102753, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0969699725000158

[6] F. Wang, J. Bi, D. Xie, and X. Zhao, "A data-driven prediction model for aircraft taxi time by considering time series about gate and real-time factors," *Transportmetrica A: Transport Science*, vol. 19, no. 3, p. 2071353, 2023. [Online]. Available: https://doi.org/10.1080/23249935.2022.2071353

[7] X. Wang, A. E. Brownlee, J. R. Woodward, M. Weiszer, M. Mahfouf, and J. Chen, "Aircraft taxi time prediction: Feature importance and their implications," *Transportation Research Part C: Emerging Technologies*, vol. 124, p. 102892, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X20307920

[8] H. Zhang and K. Zhang, "Big data analysis and intelligent traffic management for airport transportation," in *Proceedings of the 2024 6th International Conference on Big Data Engineering*, ser. BDE '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 74–80. [Online]. Available: https://doi.org/10.1145/3688574.3688585

[9] Federal Aviation Administration, "Airport Surface Detection Equipment, Model X (ASDE-X)," 2024, accessed: 2026-05-15. [Online]. Available: https://www.faa.gov/air_traffic/technology/asde-x

[10] I. C. A. Organization, "Advanced surface movement guidance and control systems (a-smgcs) manual," ICAO, Tech. Rep. Doc 9830, 2004.

[11] Eurocontrol, "Asterix," accessed: 2026-02-23. [Online]. Available: https://www.eurocontrol.int/asterix

[12] EUROCONTROL, "Sustainable taxi operations – concept of operations (conops)," EUROCONTROL, Tech. Rep., 2024, online; accessed 2026-02-24. [Online]. Available: https://www.eurocontrol.int/sites/default/files/2024-07/eurocontrol-sustainable-taxi-operations-conops.pdf

[13] Y. LIM, S. ALAM, F. TAN, and N. LILITH, "Variable taxi-out time prediction based on machine learning with interpretable attributes," *TRANSACTIONS OF THE JAPAN SOCIETY FOR AERONAUTICAL AND SPACE SCIENCES*, vol. 67, no. 3, pp. 136–144, 2024.

[14] F. Herrema, R. Curran, H. Visser, D. Huet, and R. Lacote, "Taxi-out time prediction model at charles de gaulle airport," *Journal of Aerospace Information Systems*, vol. 15, pp. 1–11, 02 2018.

[15] R. Ganesan, P. Balakrishna, and L. Sherry, "Improving quality of prediction in highly dynamic environments using approximate dynamic programming," *Quality and Reliability Engineering International*, vol. 26, no. 7, pp. 717–732, 2010. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/qre.1127

[16] G. Lian, Y. Zhang, J. Desai, Z. Xing, and X. Luo, "Predicting taxi-out time at congested airports with optimization-based support vector regression methods," *Mathematical Problems in Engineering*, vol. 2018, pp. 1–11, 04 2018.

[17] S. Ravizza, J. A. D. Atkin, M. H. Maathuis, and E. K. Burke, "A combined statistical approach and ground movement model for improving taxi time estimations at airports," *Journal of the Operational Research Society*, vol. 64, no. 10, pp. 1347–1360, 2013. [Online]. Available: https://link.springer.com/article/10.1057/jors.2012.123#citeas

[18] T. Diana, "Can machines learn how to forecast taxi-out time? a comparison of predictive models applied to the case of seattle/tacoma international airport," *Transportation Research Part E: Logistics and Transportation Review*, vol. 119, pp. 149–164, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S136655451830543X

[19] O. Lordan, J. M. Sallan, and M. Valenzuela-Arroyo, "Forecasting of taxi times: The case of barcelona-el prat airport," *Journal of Air Transport Management*, vol. 56, pp. 118–122, 2016, growing airline networks -Selected papers from the 18th ATRS World Conference, Bordeaux, France, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0969699716301570

[20] A. Srivastava, "Improving departure taxi time predictions using asde-x surveillance data," in *2011 IEEE/AIAA 30th Digital Avionics Systems Conference*, 2011, pp. 2B5–1–2B5–14. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/6095989

[21] J. Song, Y. Tang, J. He, and W. Zhang, "Importance and prunability analysis of basic features in machine-learned aircraft taxi-out time prediction," in *Smart Transportation and Green Mobility Safety*, W. Wang, L. Jin, and H. Tan, Eds. Singapore: Springer Nature Singapore, 2024, pp. 441–447.

[22] Y. Zhang and Q. Wang, "Methods for determining unimpeded aircraft taxiing time and evaluating airport taxiing performance," *Chinese Journal of Aeronautics*, vol. 30, no. 2, pp. 523–537, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1000936117300286

[23] J. Yin, Y. Hu, Y. Ma, Y. Xu, K. Han, and D. Chen, "Machine learning techniques for taxi-out time prediction with a macroscopic network topology," in *2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC)*, 09 2018.

[24] R. Dalmau, F. Ballerini, H. Naessens, S. Belkoura, and S. Wangnick, "An explainable machine learning approach to improve take-off time predictions," *Journal of Air Transport Management*, vol. 95, p. 102090, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0969699721000739

[25] J. Du, M. Hu, W. Zhang, and J. Yin, "Finding similar historical scenarios for better understanding aircraft taxi time: A deep metric learning approach," *IEEE Intelligent Transportation Systems Magazine*, vol. PP, pp. 2–17, 01 2022.

[26] F. Kato and E. Itoh, "Aircraft taxi time prediction using machine learning and its application for departure metering," in *2023 IEEE/AIAA 42nd Digital Avionics Systems Conference (DASC)*, 2023, pp. 1–9.

[27] X. Tang, M. YE, S. Zhang, and K. Fuellhart, "Prediction of taxi-in time and analysis of influencing factors for arrival flights at airport with a decentralised terminal layout," *Promet - Traffic&Transportation*, vol. 36, pp. 623–638, 08 2024.

[28] N. Antunes Ribeiro, J. Tay, W. Ng, and S. Birolini, "Delay predictive analytics for airport capacity management," *Transportation Research Part C: Emerging Technologies*, vol. 171, p. 104947, 02 2025.

[29] X. Yang, H. Yang, Y. Mao, Q. Wang, and S. Yin, "Multi-task dynamic spatio-temporal graph attention network: A variable taxi time prediction model for airport surface operation," *Aerospace*, vol. 11, no. 5, 2024. [Online]. Available: https://www.mdpi.com/2226-4310/11/5/371

[30] E. Šimić, M. Begović, M. Šabić, and E. Muharemović, "Using machine learning to predict additional taxi-out time as a airport key performance indicator in the eurocontrol zone," in *New Technologies, Development and Application VII*, I. Karabegovic, A. Kovačević, and S. Mandzuka, Eds. Cham: Springer Nature Switzerland, 2024, pp. 237–245.

[31] D. T. Pham, M. Ngo, N. Tran, S. Alam, and V. Duong, "A data-driven approach for taxi-time prediction: A case study of singapore changi airport," in *Air Traffic Management and Systems IV*, E. N. R. Institute, Ed. Singapore: Springer Singapore, 2021, pp. 113–129.

[32] P. Zippenfenig, "Open-meteo.com weather api," 2023. [Online]. Available: https://open-meteo.com/

[33] S. Okwir, K. Amouzgar, and A. H. Ng, "Exploring prediction accuracy for optimal taxi times in airport operations using various machine learning models," *Journal of Air Transport Management*, vol. 122, p. 102684, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0969699724001492

[34] A. Degas, M. R. Islam, C. Hurter, S. Barua, H. Rahman, M. Poudel, D. Ruscio, M. U. Ahmed, S. Begum, M. A. Rahman, S. Bonelli, G. Cartocci, G. Di Flumeri, G. Borghini, F. Babiloni, and P. Aricó, "A survey on artificial intelligence (ai) and explainable ai in air traffic management: Current trends and development with future research trajectory," *Applied Sciences*, vol. 12, no. 3, 2022. [Online]. Available: https://www.mdpi.com/2076-3417/12/3/1295

[35] P. Cunningham, M. Cord, and S. J. Delany, *Supervised Learning*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 21–49. [Online]. Available: https://doi.org/10.1007/978-3-540-75171-7_2

[36] A. Sternberg, J. Soares, D. Carvalho, and E. Ogasawara, "A review on flight delay prediction," 03 2017.

[37] J. W. Tukey *et al.*, *Exploratory data analysis*. Springer, 1977, vol. 2.

[38] B. Iglewicz and D. C. Hoaglin, *Volume 16: how to detect and handle outliers*. Quality Press, 1993.

[39] R. L. Ott and M. T. Longnecker, *An Introduction to Statistical Methods and Data Analysis*, 6th ed. Cengage Learning, 2008.

[40] G. James, D. Witten, T. Hastie, R. Tibshirani, and J. Taylor, *An Introduction to Statistical Learning with Applications in Python*, ser. Springer Texts in Statistics. Springer, 2023.

[41] S. Masís, *Interpretable Machine Learning with Python: Build explainable, fair, and robust high-performance models with hands-on, real-world examples*. Packt Publishing Ltd, 2023.

[42] M. Kearney, *Cramér's V*. Sage, 12 2017.

[43] Y. Liang, S. Fu, K. Wang, Z. Jiang, and X. Wang, "Predicting aircraft taxi time at airport with rrelieff feature selection and machine learning methods," in *2023 China Automation Congress (CAC)*, 2023, pp. 4800–4805.

[44] D. Freedman, *Statistical Models: Theory and Practice*. Cambridge University Press, 01 2005.

[45] L. Gianfagna and A. Di Cecco, *Explainable AI with python*. Springer, 2021, vol. 10.

[46] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems*, 2017.

[47] S. M. Lundberg, G. Erion, and H. Chen, "From local explanations to global understanding with explainable ai for trees," *Nature Machine Intelligence*, vol. 2, pp. 56–67, 2020.

[48] F. Aybek Cetek and E. Aydoğan, "Air traffic flow impact analysis of recat for istanbul new airport using discrete-event simulation," *Düzce Üniversitesi Bilim ve Teknoloji Dergisi*, vol. 7, pp. 434–444, 01 2019.

[49] SKYbrary, "Icao wake turbulence category," https://skybrary.aero/articles/icao-wake-turbulence-category, accessed: 2026-04-13.

[50] International Civil Aviation Organization, "List of low-cost-carriers (lccs)," https://www.icao.int/sites/default/files/sp-files/sustainability/Documents/LCC-List.pdf, 2017, accessed: 2026-03-26.

[51] ——, "Corsia aeroplane operators," https://www.icao.int/CORSIA/AeroplaneOperators#:~:text=If%20the%20Attribution%20Method%

20is%20%22ICAO%20Designator%22%2C,aeroplane%20operator%20is%20registered%20as%20juridical%20person., accessed: 2026-03-26.

[52] Opennav, "Opennav airline codes," https://opennav.com/airlinecodes, accessed: 2026-03-26.

[53] International Civil Aviation Organization, "Icao api data service page," https://dataservices.icao.int/default.aspx, accessed: 2026-03-24.

[54] A. C. Central, "Airline designator / code database search," https://www.avcodes.co.uk/airlcodesearch.asp, 2026, accessed: 2026-03-26.

[55] Scikit-learn developers, "Scikit-learn: Machine learning in python," n.d., accessed: 2026-04-21. [Online]. Available: https://scikit-learn.org/stable/

A

Appendix 1

A.1 Model setups

Table A.1: Random forest settings

Parameter	**Final value**
bootstrap	True
ccp_alpha	0.0
criterion	'squared_error'
n_estimators	200
max_depth	40
max_features	0.3
max_leaf_nodes	None
max_samples	None
max_impurity_decrease	0.0
min_samples_leaf	2
min_samples_split	5
min_weight_fraction_leaf	0.0
n_jobs	None
oob_score	False
random_state	42
verbose	0
warm_start	False

DEPARTMENT OF MECHANICS AND MARITIME SCIENCES
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2026
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY