TU WIEN Informatics

Predicting Taxi Time, Runway Assignment, and Deicing Usage at Vienna Airport

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Data Science

eingereicht von

Andreas Scheicher, MSc
Matrikelnummer 0827109

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Assoc. Prof. Dr.techn. Nysret Musliu

Wien, 30. August 2023

Andreas Scheicher	Nysret Musliu

TU WIEN Informatics

Predicting Taxi Time, Runway Assignment, and Deicing Usage at Vienna Airport

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Data Science

by

Andreas Scheicher, MSc
Registration Number 0827109

to the Faculty of Informatics

at the TU Wien

Advisor: Assoc. Prof. Dr.techn. Nysret Musliu

Vienna, 30th August, 2023

Andreas Scheicher	Nysret Musliu

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

Erklärung zur Verfassung der Arbeit

Andreas Scheicher, MSc

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 30. August 2023

Andreas Scheicher

Acknowledgements

First, I would like to extend my gratitude to my advisor Assoc. Prof. Dr.techn. Nysret Musliu for his support and guidance throughout this thesis. He provided invaluable feedback while granting me the freedom to pursue my own ideas.

I also want to thank Christoph Bichler for the initiative and foresight to start this collaboration and for the support.

My gratitude extends to both Oliver Pleyer and Oliver Russ for the support and their insights into airport operations.

I am grateful to my family, who consistently supported my educational pursuits. Without them, this would not have been possible.

Lastly, a special thanks to Caroline for her unwavering love and support.

Kurzfassung

Die Vorhersage von Taxizeiten kann den Flughafenbetrieb unterstützen und möglicherweise Verspätungen reduzieren, und den Treibstoffverbrauch sowie den CO2-Ausstoß senken. Einflussfaktoren, beispielsweise die zugeteilte Startbahn und die Verwendung der Enteisungsservices haben einen großen Einfluss auf die Taxizeiten. Die Entscheidungen darüber werden von Fachexperten getroffen, die eine Vielzahl an Informationen und komplexen Zusammenhängen berücksichtigen und dabei auf jahrelanges Training und Erfahrung zurückgreifen. Die komplizierten Wechselwirkungen der zahlreichen Einflussfaktoren machen dies zu einer geeigneten Anwendung von Machine Learning. Während die Vorhersage der Taxizeiten und Startbahnzuteilung bereits auf verschiedenen Flughäfen untersucht wurde, blieb die Vorhersage der Nutzung der Enteisungsservices bisher unerforscht.

Diese Diplomarbeit untersucht die Verwendung von Machine Learning um Taxizeiten, Startbahnzuteilung und Nachfrage der Enteisungsservices für ausgehende Flüge am Wiener Flughafen vorherzusagen. Eine umfassende Literaturrecherche wurde durchgeführt, wobei die Faktoren identifiziert wurden, welche bei ähnlichen Aufgaben für eine Vorhersage verwendet werden können. Weiters wurden erfolgreiche Modelltypen und nützliche Bewertungskennzahlen ermittelt. Datensätze wurden sowohl aus firmeneigenen als auch aus öffentlichen Quellen gesammelt und verschiedene Feature-Extraktion und Feature-Engineering Methoden wurden angewandt. Das beinhaltet auch einen neuen Ansatz, Vektor-Embeddings zu trainieren, um kategorische Features speichereffizient zu repräsentieren. Das somit entstandene Datenset wurde statistisch analysiert und visualisiert.

Eine Auswahl an Machine Learning Modellen wurde zusammengestellt und am Datenset für die verschiedenen Vorhersageaufgaben trainiert. Um die Modelle zu bewerten, haben wir verschiedene Szenarien definiert. Diese reichen von einer Vorhersage 30 Stunden vor dem Start, über eine Vorhersage direkt vor dem Start, bis zu Bedingungen die verschiedene Studien für einen Vergleich nachahmen. Unsere besten Modelle schneiden bei den relevanten Metriken besser ab als ein Referenzmodell. Im direkten Vergleich mit der Literatur schneiden unsere Modelle besser ab als die besten Modelle von einem Teil der Flughäfen. Schließlich haben wir die wichtigsten Einflussfaktoren für jede der Vorhersageaufgaben identifiziert. Darunter fallen wetterabhängige Faktoren, der Flugzeugtyp, der Zielflughafen, das aktuelle Flugaufkommen, sowie Lärmschutzmaßnahmen.

Abstract

The prediction of taxi-out times can enhance airport operations, potentially reducing delays, fuel burn and carbon emissions. Factors such as runway assignment and deicing service usage greatly influence taxi-out time. The decisions are made by domain experts, who consider an extensive amount of information and its complex interplay, utilizing years of training and experience. The intricate interactions of numerous factors make this an appropriate application for machine learning. While the prediction of taxi-out time and runway assignment has been studied at various airports with differing results, the prediction of deicing usage remains unexplored.

This thesis investigates the use of machine learning in predicting taxi-out time, runway assignment and deicing usage for outgoing flights at Vienna Airport. A comprehensive literature review was conducted, identifying the factors with predictive capabilities on similar tasks, successful model types, and useful evaluation metrics. Datasets were collected from both proprietary and publicly available sources. Feature extraction and feature engineering methods were then applied. This includes the novel approach of using vector embeddings to represent categorical features, which permits a memory-efficient encoding of the information. The resulting dataset was analysed using statistical methods and visualizations.

A selection of machine learning models was curated and trained on the dataset for the various prediction tasks. To evaluate the models, we defined varied scenarios, from predictions up to 30 hours ahead of time with limited information, to a prediction at the time of block-off to conditions that mimic specific research studies for an appropriate comparison. Our best models, when evaluated on the most relevant metrics, outperformed a baseline model. In comparison to the existing literature, our models surpassed the best performing models on a subset of airports. Finally, we identified the most important features for each prediction task, revealing the influence of weather-related factors, aircraft type, flight destination, the current demand at the airport, and noise abatement measures.

Contents

CHAPTER 1

Introduction

This thesis examines the application of Machine Learning (ML) in the context of airport operations. Air traffic volume has been growing steadily over the past decades. While the pandemic temporarily lowered demand, Eurocontrol forecasts 2019 levels to be surpassed by 2025 [Eur22]. This increase comes with significant challenges for airport operations and increases the risk of delays and uncertainties. Predicting various operational characteristics can help mitigate these risks and could aid air traffic controllers as a decision-support tool. Moreover, it could serve as an early warning for potential congestion, allowing for timely mitigation actions [VTJ21]. Furthermore, it could be utilized to improve the efficiency of airport surface movement operations and reduce fuel burn, CO_2 emissions and costs.

One target for such predictions is *taxi time* or *taxi-out time*, the time between an aircraft leaving the parking position and taking off. Numerous factors influence this variable. Some of them are decided by airport or aircraft operators, such as the parking position, the assigned runway or the use of deicing. Other factors are external, such as the noise abatement measures, demand for incoming and outgoing flights, or current weather observations and future weather predictions. The complexity of airport operations and the large number of factors at play make ML a promising solution for such predictions.

In the existing literature, different ML models have been applied to prediction tasks related to airport operations, with the results varying across airports. Diana [Dia18] investigated taxi time prediction at Seattle–Tacoma International Airport (SEA) and found that no algorithm performed best in all cases. In some cases, linear regression outperformed the more complex ensemble models. Lee *et al.* [LCJ19] found ML models not to outperform the baseline of a constant value in taxi time prediction. Conversely, Wang *et al.* [WBW+21] found Random Forest (RF) and Gradient Boosted Regression Trees (GBRT) clearly outperformed linear models.

Airports vary widely in their layout, operations, climate, and connectedness with other airports, leading to differing results. Balakrishna *et al.* [BGS10] remarked an increased difficulty in predicting taxi times at John F. Kennedy International Airport (JFK) compared to Detroit International Airport (DTW) or Tampa International Airport (TPA). Ravizza *et al.* [RAMB13] specifically pointed out differences between North American and European airports.

This thesis explores the application of ML in predicting taxi time, runway assignment and deicing usage at Vienna Airport. We start with a comprehensive literature review on the state of the art of these prediction tasks. Together with insights from conversations with domain experts, we identified the requirements for this type of forecasting, including the forecast horizon, evaluation metrics, and a baseline for a comparison. We compiled a dataset from different sources, applying the necessary preprocessing. This dataset is then analysed, using statistical methods as well as visualizations. A list of ML algorithms is created and each model is applied to the prediction tasks. After the models are tuned and trained on the specific tasks, they are evaluated against each other and against a baseline using an unseen part of the dataset. Following that, we analysed the features used for those predictions and identified the most important ones. The final aim is to arrive at a recommendation for implementing such a system.

In the course of this thesis, the following research questions are answered:

1. How much can the use of machine learning models improve the prediction of taxi time, runway assignment, and deicing usage above baseline?
2. Which features are most relevant for these predictions?
3. Which algorithm is the most appropriate for this task and what are the optimal hyperparameters?

Contribution

Our main contributions throughout this thesis include a new approach in feature engineering, the application of ML on the novel task of deicing usage prediction, and a comprehensive comparison of different ML tasks in various scenarios.

We conducted an up-to-date literature review on the prediction tasks addressed in this thesis, providing an overview of ML application in taxi time and runway assignment predictions. This also includes related tasks, such as airport acceptance rate and runway configuration prediction.

Flight data contains several categorical features of high cardinality. Traditional encoding techniques tend to result in high-dimensional datasets, which cause a lot of memory usage and an increase in training and evaluation times. To solve the problem of high cardinality categorical input data, this thesis proposes the use of embeddings as a low dimensional

vector representation using neural networks. This way we encoded the information in a dense way, which leads to an improvement in memory usage and performance.

This thesis represents the first application of ML for predicting taxi time and runway assignment at Vienna Airport. To the best of our knowledge, this is the first study focused on the prediction of deicing usage on aircraft.

Outline

Chapter 2 provides an overview of the current state of the art. This identifies the approaches that have been tried on the tasks in question so far, gathers the methods and evaluations that have proven useful and identifies gaps in the literature. Chapter 3 describes the data collection and preprocessing, the ML techniques that were applied, and the evaluation metrics. Chapter 4 analyses the generated dataset. It demonstrates dependencies of the target variables and the predictor features and identifies patterns in the data. Since the weather forecast is an essential data source for this task, the quality of the weather forecasts is evaluated. Chapter 5 presents the results obtained from applying ML techniques to the prediction tasks. This includes the optimization of the models, their evaluation, and the identification of the relevant features. The chapter concludes by comparing our findings to the existing literature. Chapter 6 provides a summary of the work.

CHAPTER 2

State of the Art

The prediction tasks of this thesis have been studied to varying extents in the existing literature. While taxi time prediction has a comparatively large existing body of literature, runway assignment prediction is limited to a few papers and deicing usage prediction has not been studied in the literature. In the case of runway assignment, we decided to expand the literature review to include predicting runway configurations and airport acceptance rates. These tasks are related and share similar predictors.

2.1 Taxi time

The available literature includes papers of various different airports in different countries. A large number of techniques from different fields were applied to the task of taxi time prediction. We divided the existing literature into early approaches and statistical models, and papers that have a higher focus on the application of ML.

Early Approaches and Statistical Models

Idris *et al.* [ICBK02] examined taxi time prediction at Boston Logan International Airport (BOS). They used multiple linear regressions and identified important predictor features for this task. They highlighted factors such as the current runway configuration, the amount of traffic on the surface and the current queue size. Their model showed a twenty percent improvement above a baseline of a 14-day running average. Outliers of long taxi time or large queue sizes were removed from the dataset.

Jordan *et al.* [JIR10] modeled taxi time for incoming and outgoing flights at Dallas/Fort Worth International Airport (DFW). They used linear regression and Sequential Forward Floating Subset Selection (SFFSS), which selects a subset of basis functions. The SFFSS model achieves an Accuracy (ACC) within 2 *min* of up to 100% compared to a Linear Regression 97.9%. Among the predictor variables are taxi distance and congestion. This

is only known in certain use cases where the forecast horizon is short. Therefore, the results might not generalize to longer forecast horizons. A limitation of this paper is the evaluation on single days.

Srivastava [Sri11] used high resolution position updates from surveillance systems to evaluate the traffic flow and predict taxi time at JFK. They approached taxi time prediction in two different ways. The first being the Uniform Flow Model (UFM), which treats the entire time between pushback and takeoff as one process. The second being the Split Flow Model (SFM) which divides the taxi time into the movement to the queue and the waiting time in the queue. They applied Linear Regression using the predictor features queue position, runway distance, arrival rates, departure rates, and severe weather among others. They found a better performance of SFM in the case when queues were present, but a more consistent performance of UFM across all different conditions. Their maximum forecast horizon was limited to around 30min, starting when the airplanes first appeared in the surveillance system.

Diana [Dia13] analysed how different factors influence taxi time at JFK using survival and frailty models. They compared the vacation seasons June to August of 2006 and 2007. Observations that had a cloud ceiling below 2000 feet and visibility below 4 nautical miles were removed from the dataset. They found block delay and the percent of airport capacity utilized to have the highest impact in increasing the risk of longer taxi times. Block delay is the difference between actual and scheduled time of an aircraft leaving the parking position.

Machine Learning Approaches

Balakrishna *et al.* [BGSL08] [BGS09] [BGS10] investigated the use of reinforcement learning for taxi time prediction of outgoing flights. They used Markov Decision Processes to predict the average taxi time in 15 *min* intervals. As evaluation metric, they used ACC within 1.5, 3, and 5 *min*. The results of their approach differed among airports, ranging from an average 60% ACC of a prediction within 5 *min* at JFK to 93.7% ACC within 1.5 *min* at TPA. Due to specifics of JFK, they deemed it useful to evaluate the model performance on this airport separately before and after 16:00. This resulted in 65% ACC within 5 *min* before 16:00 and 53% afterwards. These findings suggest airport specifics play an important role in taxi time prediction and results cannot necessarily be expected to generalize.

Chen *et al.* [CRAS11] introduced a Fuzzy Rule-Based Systems (FRBS) approach in airport operations. They applied their method on a dataset from Zurich Airport (ZRH) and achieved 98.8% ACC within 3 *min*, which is an improvement compared to their baseline of linear regression with 95.6%. In a subsequent paper, Chen *et al.* [CWZ+17] extended their model to predict the uncertainty in addition to taxi time. Their approach was tested on a dataset of Manchester Airport (MAN) and generated taxi time predictions with up to 96% ACC within 5*min* and 86% ACC within 3*min*. This, again, highlights the differences between airports.

Ravizza *et al.* [RAMB13] used multiple linear regression analysis to find the most relevant factors having an impact on taxi time. They focused on outgoing flights at the airports Stockholm Arlanda Airport (ARN) and ZRH. They found the distance to be the most important factor. The difference between the number of departures and arrivals played another important role. They noted that departing aircraft often have to wait in a queue and therefore, have a lower average speed compared to arriving aircraft, which have to leave the runway as soon as possible. Their models showed better average performances than Balakrishna *et al.* [BGS09] at TPA and Idris *et al.* [ICBK02] at BOS. In a subsequent paper, Ravizza *et al.* [RCA+14] tested different regression algorithms to predict taxi time at ARN and ZRH. In an initial investigation, they concluded that Decision Tree (DT), k Nearest Neighbors (kNN), Multilayer Perceptron (MLP) and Gaussian Processes did not show promising results and did not qualify for further investigation. Instead, they focused on multiple linear regression, least median squared regression, Support Vector Machine (SVM), M5 model trees and FRBS. The FRBS approach achieved the best results. Their datasets were limited to two days at ARN and eight days at ZRH.

Diana [Dia18] compared several ML models on predicting taxi time. They concluded that models tend to overfit if they are too complex for a given dataset. This reduces their performance on unseen data. They found Cross-Validation (CV) and Root Mean Squared Error (RMSE) to be the most useful for assessing model performance. The best bias/variance balance was achieved by Linear Regression and Ridge Regression, as well as Gradient Boosting Machine (GradBoost) Regression.

Yin *et al.* [YHM+18] noted that previous research on taxi time prediction often focuses on departures alone, while ignoring the impact of arrivals. In reality they are closely linked and both factors need to be considered. The authors used a selection of ML models and a set of features related to aircraft movement, queues, and current demand on the airport. The best performing model in this paper was the Random Forest (RF). When comparing a training set of one day to a set of one month, the longer one led to better performance on unseen data.

Lee *et al.* [LMZ+15] compared taxi time prediction of fast-time simulation with several ML models at Charlotte Douglas International Airport (CLT). Their proposed simulation tool showed a comparable performance to SVM. In their findings, kNN and RF achieved the best performance. The datasets used for the evaluation were created in simulations. In a later paper, Lee *et al.* [LMJ16] tested different ML models using actual traffic data of CLT. The dataset contained three months, from June to August 2014. The train-test split was done by manually selecting four days of different traffic and weather conditions for the test set. They observed linear regression and RF having the best performances but described these performances as “not satisfactory”. They explain this by uncertainties caused by human operators and the external environment.

Lee *et al.* [LCJ19] divided the taxi time into push back time and ramp taxi time using surface surveillance radar data. They analysed the distributions of both times and trained several ML models on predicting them. Separate features were chosen as input for both prediction tasks. They included information about the aircraft and airport operations.

Weather forecasts were not among those features. The authors concluded that the ML models showed a similar performance as their baseline models.

Vargo *et al.* [VTJ21] used ML to predict whether the average taxi time on an airport will exceed a predefined threshold. An increased average taxi time can be a warning sign of flight delays caused by a gridlock. This approach differs from the previous literature, where taxi time was predicted for individual flights.

Wang *et al.* [WBW+21] focused on iterative feature elimination to identify and quantify feature importance. Of the ML models they tested, RF achieved the best result. Training this model on a subset of the most important features led to less than a 1% performance drop-off on ACC within 1, 3, and 5 *min*. The authors noted that applying alternative ML algorithms would be suitable future research. Xia and Huang [XH22] used a Neural Network (NN) for taxi time prediction at a major airport in southern China. They found the best model performance when only using features which have a strong or medium correlation with taxi time. They observed a performance drop-off when adding weakly correlated features. The hyperparameters were tuned using a Genetic Algorithm, as well as a Sparrow Search Algorithm. Their dataset consisted of two weeks in May and June 2019.

Wang *et al.* [WBXZ22] used an informer RF regression model to predict taxi time. The informer model is a deep NN, which contains information about historic taxi times. Their approach outperformed a list of conventional ML models on the dataset of all 2019 flights at Beijing Capital International Airport (PEK). In their conclusion, they noted a lack of explainability of their model.

2.2 Runway Assignment

Prediction tasks related to airport runways focus on various target variables. The runway assignment specifies which runway a particular aircraft uses for takeoff or landing. The runway configuration determines which runways are currently available for being assigned to departing and arriving aircraft. The Airport Acceptance Rate (AAR) represents the number of arriving aircraft in a specific time frame. Each runway has a capacity limit for being used by departing and arriving aircraft. Therefore, the runway configuration restricts the overall airport capacity and AAR. Since these target variables are related, they share common predictor features. Among those target variables, this thesis only focuses on predicting runway assignment. However, due to the relation between all mentioned target variables, it was useful to extend the literature review and include papers that investigate the prediction of runway configuration and AAR.

Early Approaches and Statistical Models

Provan *et al.* [PCC11] introduced a statistical model to predict AAR which uses weather forecasts as predictors. This approach resulted in a 50% lower RMSE compared to a baseline of constant AAR. Buxi *et al.* [BH11] presented a method for generating

probabilistic capacity profiles. These profiles were generated for each day and contained weather information of an entire day. They used Terminal Aerodrome Forecast (TAF) and San Francisco Marine Stratus Forecast System (STRATUS) as input and applied Principal Component Analysis (PCA) and k-means clustering to create the profiles. Their approach improved ground delay programs and reduced costs of delays compared to a weather independent approach. These results emphasize the importance of weather-related features, when generating predictions related to runways.

Dhal and Roy [DRT+14] created a modeling framework to predict airport arrival and departure capacity. The first step of their approach is distinguishing high capacity and low-capacity airports. The second step is predicting the runway configuration. In the last step, the capacity is estimated using the weather. Their work highlights the differences between various airports due to differing demands, baseline capacities and the weather. Tien *et al.* [TRT+15] evaluated Dhal and Roy's model on 35 airports. They used TAF instead of recorded weather, and found a comparable performance for up to 24 hours of lookahead time. The performance varied across airports. They suggested that at airports where the model performed less well, it is due to airport-specific operations.

Ramanujam and Balakrishnan [RB15] developed a statistical model predicting runway configuration selection. They introduced a feature "inertia", which represents the resistance to runway configuration change. Furthermore, they defined a utility function to model the decision-making process. This function was used to determine feature importances. This showed the statistical significance of features such as headwind speed relative to the runway and noise abatement measures. Their discrete choice model achieved good results predicting runway configurations with a forecast horizon of 3 hours. Avery and Balakrishnan [AB15] [AB16] extended this approach to LaGuardia Airport (LGA), San Francisco International Airport (SFO), and Newark Liberty International Airport (EWR). They also introduced limits for maximum tailwind and crosswind components on the runways. Their findings show only a slight performance drop-off when using weather forecast instead of observed weather. They also found that "inertia" was less effective for longer forecast horizons.

Machine Learning Approaches

Wang [Wan11] used a SVM and an Ensemble Bagging DT to predict runway configurations. Performances were evaluated using a 10-fold CV. Due to the large number of runway configurations and their class imbalance, the models were trained to predict either a subset of the most common configurations or one of the most common configurations vs all others. The Ensemble Bagging DT outperformed the SVM and achieved an ACC of 85% on detecting the most common runway configuration vs the second most common one and an ACC of 76% on the most common runway configuration vs all others at EWR. In a subsequent paper, Wang [Wan12] compared the performance of multiple linear regression models with bagging DT for predicting AAR. Again, bagging DT achieved the best results, to which the author concluded the weather has a non-linear impact on AAR.

Nakamura and Jung [NMAJ17] investigated the use of NN to predict differences in runway assignment on subsequent flights. They found differing performances depending on day and nighttime, as well as the direction of traffic. They also observed day to day changes in airport operations, which added difficulty in the prediction task. As a metric, they used ACC and noted that the data set contains a class imbalance. A high ACC in this context may not necessarily represent the usefulness of a model.

Ahmed *et al.* [AAB18] used a multi-layer artificial NN to predict runway configurations. The predictor features included weather and aircraft information. As a preprocessing the dataset was scaled to a standard format to improve the performance of the NN. The model achieved an acceptable ACC according to the authors. A limitation of this paper is the validation being done on a dataset of one day.

Murça and Hansman [MH18] focused on the interdependence of the major airports of the New York metropolitan region. They used Bayesian Regression, RF Regression and Gaussian Process Regression to predict arrival rates. They found the metroplex configuration to be the second most important feature. This interdependence of different airports highlights the complex interplay of the surrounding area and the differences among airports.

Herrema *et al.* [HCH+19] used a GradBoost algorithm to predict the exit an aircraft takes at Vienna airport. The dataset was limited to one runway and one runway configuration. Their features included aircraft specific information, as well as weather conditions and current demand.

Churchill *et al.* [CCJ21] applied ML to predict runway assignment. They tested an Extreme Gradient Boosting (XGBoost) model and found it outperformed Logistic Regression. Their hyperparameter tuning resulted in a small improvement of the model performance compared to the default hyperparameters. They noted that results were more accurate for departing flights than for arriving flights. Their input features include the runway configuration, which is often not known in advance and is a prediction target of multiple other papers. They applied their models on datasets from 2019 and 2020. The performance on the 2020 datasets was worse, due to the reduced demand of air traffic caused by the pandemic. They selected this dataset with the aim to deploy the model amidst the ongoing effects of the pandemic. Furthermore, their train-test split was done randomly. This means instances of the training set can be in close proximity to the test set. Given the tendency of successive flights to occur during the same runway configuration and have similar runway assignment, this may cause an insufficient separation of the train and test data. One caveat to the generalisations of their finding for longer forecast horizons is the use of the runway configuration.

Guang *et al.* [GAAP+21] used a RF Regressor and an adaptive RF Regressor for predicting runway capacity on both departing and arriving flights. They found the most important features to be related to demand and wind.

Khater *et al.* [RKC21] evaluated the use of a RF and XGBoost classifier in a multi-step model for time series forecasting of runway configuration. Each time step takes the output

of the previous timestep as input. They evaluated the model performance over different forecast horizons of up to 6 hours and saw a drop-off in ACC with longer horizons. The performance varied across different airports. The model was evaluated on a 2019 and a 2020 dataset. The results were worse on the 2020 dataset due to the influence of the pandemic. Features related to demand were less important on the 2020 dataset due to the reduced overall demand. The train test split of the dataset was performed on a weekly basis to ensure a better separation between training and testing data.

Wang and Zhang [WZ21] focused on the multi-airport system of the New York Metroplex and used a Convolutional Neural Network (CNN) to predict AAR and runway configuration simultaneously. They used a 29 x 29 point grid spanning an area of 200 x 200 nautical miles and 64 weather forecast features from 19 vertical layers. The input features did not contain demand related information. The model was evaluated over a forecast horizon of 1-8 hours, and they did not find the same drop off in prediction ACC on the same airports as Khater *et al.* The authors concluded that their model outperformed the results of models in previous papers on the same airports.

Raju *et al.* [RMW+21] compared different models on predicting runway configuration and arrival and departure rates. They found that most ML models outperformed a rule-based approach. However, they did not find one model that consistently outperforms all other models. The performance of the models varied across airports and for different forecast horizons. They compared the results of random sampling to forward validation. In random sampling, a data point of the validation set can be from an earlier point in time than a data point of the training set. In forward validation, the data points of the validation set are from a later time than the ones of the training set. Random sampling leads to better results. However, the forward validation is closer to a real-world application, where the model is trained on historic data and evaluated on new data. They found a drop off in performance with an increased forecast horizon.

Kanjanasurat *et al.* [KJPB22] used a multi-layer NN to predict landing runway assignments. They used radar data of individual flights and generated features which contain the demand on the runways. The runways were limited to two classes and other features, such as weather or aircraft specifics were not considered. In a following paper, Kanjanasurat *et al.* [KJLB23] compared the performance of Logistic Regression and RF on runway assignment for arriving aircraft. In their findings, Logistic Regression had the better performance.

2.3 Deicing Usage

There is limited literature regarding aircraft deicing. The available papers focus on increasing the efficiency of airport operations regarding deicing, on it's environmental impact, or on providing a decision support for airlines determining whether they should make use of this service. All of these, while significant in their own context, are not related to the use case in this thesis. While there are no papers available that deal with deicing usage, Srivastava [Sri11] acknowledged the influence of deicing usage and noted

that their proposed models would need to be recalibrated for winter when deicing is active and has an impact on taxi time. To the best of our knowledge, the use of ML to predict deicing demand at airports has not been studied.

2.4 Discussion

The available literature gives a good insight into the application of ML on the taxi time and runway assignment prediction tasks. While this is a promising approach that has led to numerous good results, there are several caveats to the existing literature.

The various datasets in the literature differ in size and features. Many papers evaluated the models on very small datasets. In some instances, the test sets contained only several days. This has the downside of a limited variety in input data, such as weather or demand, making the result less representative. Only a full year contains all seasonal changes and is representative of the weather which can occur at a given airport. Additionally, longer datasets have shown to lead to better results. Some papers used datasets that had outliers or difficult weather conditions removed. The criteria for the removal were not always specified.

In many papers, the train-test-split was either not described or the method of splitting was not specified. An example is Wang *et al* [WBXZ22]. While they used an 80% - 10% - 10% train-test-validation split, it was not specified, how this split was made. Taxi times or runway assignment of consecutive flights are expected to correlate because they are influenced by the same weather or congestion. This relates to the concept of "inertia", introduced by Ramanujam and Balakrishnan [RB15] and confirmed by Avery and Balakrishnan [AB15] [AB16]. It describes a resistance to runway configuration change. In the case of runway assignments, this describes a propensity of consecutive flights to be assigned the same runway. If the dataset was shuffled before making the split, it is expected that consecutive flights are placed in different sets. This might lead to information leaking from the validation to the training data. If the dataset was not shuffled and the validation set consists only of flights which occurred after the flights in the training and test sets, it would lead to a better separation and the results would generalize better to a real-world deployment. This is supported by Raju *et al.*'s [RMW+21] finding where forward sampling leads to worse results but is more applicable in a real-world scenario compared to random sampling.

Wang *et al.* compared their results on runway configuration prediction with previous studies on the same airports and showed their model's improved performance. However, they noted that previous papers were using observed weather instead of weather forecasts, which likely overestimates their performance in real world settings and increases the difference in performance. Numerous other studies use datasets created from weather observations. In the cases where the datasets are created from weather forecast, they often only consider a short forecast horizon of a few hours.

A large number of papers focused their investigations on US airports. According to

Ravizza *et al.* [RCA+14], there are structural differences between US and European airports which cause problems when adopting the findings on US airports to European airports. One example of such differences is the runway queue playing a more important role for taxi time prediction in the US, while the taxi distance is more important in Europe.

Results of the various approaches differ across various airports. This goes as far as some papers reporting good predictive ability of ML, and others reporting similar models don't outperform baseline. Different airports were shown to lead to different model performance, which suggests the necessity of investigating each individual airport. A general problem when comparing different results is the lack of reproducibility due to the use of on proprietary datasets.

CHAPTER 3

Methods and Application of Machine Learning

This chapter describes the methods used to address the research questions in this thesis. The research method describes the overall structure that was followed. The data collection introduces the publicly available datasets which were used along with the proprietary datasets of Vienna Airport. The preprocessing contains steps that were used to transform raw data into a form that can be used by machine learning algorithms. The ML techniques introduce the algorithms and the evaluation metrics, along with the methods used for hyperparameter tuning and feature evaluation.

3.1 Research Method

As the research method for this thesis, we chose the CRISP-DM Cross Industry Standard Process for Data Mining (CRISP-DM) framework [WH00] and modified it to fit the scope of this thesis.

Literature Review

In the first stage, we evaluated the state of the art. This started by identifying literature that deals with similar problems, and researching approaches which have been tried and the generated results. A special focus lies on ML approaches and the methods and evaluation metrics that were used.

Business Understanding

In the business understanding phase, this information was extended by the domain experts working at Vienna Airport. Their input informed the constraints and requirements for

forecasting. Furthermore, it gave an insight into the operation specifics of Vienna Airport and formed the base for the creation of the data sets.

Data Collection and Data Understanding

The data collection phase started by collecting datasets from various sources and analysing them. This phase was closely linked to the previous stage, as the data sources were in part identified by the domain experts and the later data analysis targeted to features.

Data Preparation

In the data preparation phase, we collected the datasets, preprocessed them, and brought them into a form suitable for ML models. This included detecting outliers, handling missing values, adjusting data types and data structures, and selecting relevant features. This also included splitting the dataset into appropriate subsets for the development and the evaluation phase.

Modeling

In the modeling phase, different methods and ML algorithms were applied on the dataset. This includes tuning their parameters to optimize them for the task at hand. The algorithms as well as the strategies for optimizing them were informed by the literature review.

Evaluation

In the evaluation phase, those models were evaluated and compared to determine the best performing model for each prediction task. This included evaluating the dataset and determining which features are relevant.

Recommendation for Deployment

In the last phase, a recommendation for deployment was generated. This specified the used datasets, the preprocessing, the algorithms, and the evaluation metrics for continuous quality control.

3.2 Data Collection

The main datasets utilised in this thesis are the outgoing and incoming flights of Vienna Airport, as well as the weather observations and forecasts of the same time period. The flights dataset contains the International Air Transport Association (IATA) code of the destination. An additional dataset of airports and their coordinates was used to extract the distance and direction of each flight's destination from this code.

Flights

The flights datasets were provided by Vienna Airport in a structured format. They are separated in Inbound and Outbound flights. The datasets of outgoing flights contain 120,502 flights in 2018 and 133,400 flights in 2019. There are a total of 66 features for each flight. The choice of this time frame is in line with the suggestion of Churchill *et al.*, where they suggested using 2018 and 2019 data to exclude the effect of the pandemic. [CCJ21]

Weather

We decided to use Meteorological Aerodrome Report (METAR) and TAF as weather observations and forecast. These weather reports are generated specifically for the use in aviation and are published for airports. The historic reports were obtained from ogimet.com [Val22]. This website features a query tool that allows for accessing historic reports in text form in batches of up to 31 days. The METAR is updated with a frequency of 30 minutes. The TAF is regularly updated every 3 hours, with irregular amendments and corrections in between. Before April 26th, 2018, the source only contained four regular TAF per day. The regular TAF, combined with the amendments and corrections make a total of 2886 for 2018 and 3402 for 2019. There are 17704 METAR for 2018 and 17705 for 2019. An additional TAF were obtained for the last days of 2017 to generate predictions of the maximum forecast horizon for flights on the first days of 2018.

Airports

We used a dataset consisting of 9300 airports, including their acronyms and coordinates [Par22]. This dataset is available under the MIT license.

3.3 Preprocessing

The following section describes the preprocessing that was applied on each of the datasets. This includes filtering the data, extracting new features, and changing datatypes so the datasets can be used by ML algorithms. In the case of weather observations and forecasts, it involved parsing the text-based data into a semi-structured form, and subsequently turning it into a structured form, where each published forecast has a value for each moment in the forecast horizon.

The preprocessing began by removing cancelled flights and empty columns from the data set. The data types were checked and cast into more appropriate data types. This was necessary for later operations on datetime and integer values. Several categorical values were encoded into numerical ones. They include the EU and Schengen status of a flight, pier usage, and handling agent. The gates were separated into pier, push-back, and roll-through positions. The taxi time, which is one of the target variables, was computed by subtracting the “actual time of block off” from the “actual time of take off” [XH22]. The scheduled datetime was used to create several different categorical and

numerical values. After consulting the domain experts, the daytimes were divided into the groups of nighttime (23:30 - 05:30), morning (05:30 - 07:00), daytime (07:00 - 21:00), and evening (21:00 - 23:30). Other such groups are related to weekday vs weekend and to the seasons. The day of the month was used as a numeric feature since it is relevant for noise abatement measures.

The runway usage has target ratios. To generate features, which indicate the actual runway usage ratios compared to the target values, we aggregated the flights over time, counted the number of flights on each runway for each moment in time, and calculated the difference of the ratios of each runway compared to the target values. This was done for both incoming and outgoing flights on both a yearly and monthly basis. Multiplied with the four runways, this led to a total of 16 features.

Yin *et al.* showed a strong correlation between taxi-out time and runway queue. This was described as "the sum of aircraft that land on and take off during the taxi process of any reference aircraft, is an indicator of the runway saturation level and hence the level of congestion at the taxiway" [YHM+18]. In this thesis, the current demand was used instead of runway queue. The runway assignment is unknown at the time the forecast is generated, so the overall demand was considered instead of the demand on each runway. The scheduled incoming and outgoing flights were aggregated for each 10min interval in the data set. These aggregated values were then assigned to each flight as two new features. To consider the demand in a larger time frame, two additional features were created, which include the demand of the preceding and succeeding 10min intervals. Those features include the demand in a 30min interval, where the flight is scheduled between the minutes 10 and 20.

Weather Data

The preprocessing of METAR and TAF started by extracting the reports from the text of the website. Then the individual reports were parsed using the metar-taf-parser-mivek [KPA22]. Each of the reports was in a semi-structured form, with a differing number of features. A TAF starts with a forecast beginning at the moment the TAF is valid. It can then have any number of trends within the forecast horizon of 30 hours. A trend can come in the form of a sudden change, called from (FM), a gradual change, called becoming (BECMG), a temporary change, called temporary (TEMPO), and a change with low probability, called probability (PROB). A PROB comes with a probability and can come with the notion of being temporary. The TAF trends have a resolution of one hour. To generate a tabular dataset, we extracted a forecast for each hour of the validity time span of each TAF.

The forecast contains numerical features, such as the wind speed and direction, temperature range, and visibility, as well as categorical features, such as the weather phenomena rain, snow, fog. The domain experts at the airport gave us a list of phenomena, which cause limitations in the operations. Another feature was created which took the value one if any of those phenomena are present, and zero otherwise. The categorical features

were one hot encoded to comply with the requirements of some of the algorithms. The dew point was found only in the METAR. To include this in the forecast and meet the requirement of having the dataset use only information that would be available in a real world use case, we used the dew point from the metar at the time the taf was published.

While METAR and TAF come in a standardized form, they are intended for the use in real time by pilots and airport personnel. As such, they are not optimized for the data operations of this thesis. This posed some difficulty in handling the data. One of them was a short datetime format, where only the day of the month is included, but not the month. To get the correct datetime of each moment of the forecast, we needed to keep track of whether the end of the forecast horizon was in the same month as the beginning. Another difficulty was caused by the occurrence of both the hour 0 and 24. In different instances, midnight was either expressed as hour 24 of the day before, or as hour 0 for the next day.

The wind direction was given as degrees. This poses the problem of a wind from direction 350° and 10° having directions that are close but take on values that are far apart. In order to solve this issue, we utilized the wind speed and direction and computed the headwind and crosswind components for each runway.

Merging the data set of flights and weather was informed by Churchill *et al.* They created a dataset with an instance for each minute of the forecast horizon, which results in a very large dataset. They then used a subset of this dataset which has the approximate size of the total number of flights during this period [CCJ21]. In contrast to that, we merged the flights and forecast datasets in a way, where each flight was assigned multiple forecasts, but each are from a different TAF. As a result, each row of the dataset is a unique flight-TAF combination

Flight Destination

The direction and distance to the flight destination was computed using the pyproj library [Wsc23] and the coordinates of Vienna airport and the destination airports [Par22].

3.4 Embeddings

A common technique for encoding categorical features is One Hot Encoding [PPP17]. In this technique, each distinct class in a categorical feature is represented by a new feature. The feature corresponding to a given record's class is filled with one, while all other features, corresponding to different classes, are filled with zero. One Hot Encoding can lead to a large number of new features in case of high-cardinality features. Many ML algorithms require the input to be numerical. This technique achieves this goal without loss of information. A drawback of One Hot Encoding is the increased number of features, especially for high-cardinality data. This leads to an increased memory usage, considering the space complexity of a dataset with N instances and M features $O(N \cdot M)$. For a

given model, prediction time is expected to increase at least linearly with the number of features [PVG+11].

The increased number of features leads to an increased volume. This leads to the dataset becoming more sparse, which raises the necessity for larger datasets to train the models. Another effect of a high-dimensional feature space is the increasing distance between data points. This poses a problem for algorithms which rely on a distance measure. These phenomena are often described as the curse of dimensionality [Bel66]. In the example of the flight callsign, there are 1864 unique values in the development set. If One Hot Encoding was applied, this would result in 1864 new features and a more than 10-fold increase of the number of features. This raises the need for a different type of encoding.

There are several other encoding techniques. Examples are target encoding, where each category is replaced by an aggregated value based on the target variable [Mic01]. A drawback of this technique is the potential for overfitting the training data, especially in case of less frequent classes. Another example is frequency encoding, where each category is replaced by its occurrence in the data set. While these techniques don't lead to an increase of features, they do cause a loss of information.

To solve the problem of encoding high-cardinality categorical data, we decided to train vector representations as proposed by Bengio *et al.* [BDVJ03]. In this technique, we map the categorical feature to real-valued vectors, which are called embeddings. The dimension of the vectors can be chosen freely and is usually significantly lower than the cardinality of the feature, which allows for a dense representation.

The embeddings are trained as the first layer of a NN. Among the advantages of NN is their ability to learn complex, non-linear relationships. The weights of the embeddings are learned from scratch and are initialized using a uniform distribution. The input for the embedding layer is label-encoded features, where each class is mapped to an integer value. The embeddings layer is followed by a dense layer with a Rectified Linear Unit (ReLU) activation function and dropout is used for regularization. The last layer is the output layer which has a separate output for each of the target variables. For the regression and binary classification tasks, the output consists of one neuron each. For the multiclass classification, the output consists of four neurons, each representing one of the four runways. The activation functions of the output layer differ for the type of learning task. *Linear* was chosen for regression, *Sigmoid* for binary classification, and *SoftMax* for multi-class classification. The loss function is the sum of the individual loss functions of each task. They are Mean Squared Error (MSE), binary cross entropy, and categorical cross entropy, respectively. The embeddings are trained for the three tasks of taxi time, runway assignment and deicing usage prediction simultaneously. We decided to use the *Adam* optimizer for its computational efficiency and low memory requirement [KB15].

The embeddings are trained on the development set. To address the case of a new class appearing in the evaluation set, we added an additional record to the dataset which contains an "unknown" class. If a class is only present in the evaluation set, but not in the development set, it will be assigned this value. The target values for this "unknown"

record are calculated from the development set as the median of taxi times, and the majority classes of the deicing usages and runway assignments. This is a preventative step to avoid errors in case of unseen categories. The model performance on unknown classes is expected to be lower since they are not represented in the training data.

The dimensions of the embedding vectors are hyperparameters which were optimized by a method analogous to the one proposed by Gu *et al.* [GTAR21]. We started by training embeddings using a dimension equal to the number of classes. This dimension is assumed to be sufficiently large and its loss serves as a reference L_∞. The optimal dimension of the embeddings $d_0(\epsilon)$ is determined by the maximal difference ϵ between L_∞ and $L(d)$ as shown in equation 3.1.

$$d_0(\epsilon) = \arg\min_d (L(d) - L_\infty < \epsilon) \tag{3.1}$$

To visualize the embeddings, a 2-dimensional t-Distributed Stochastic Neighbor Embedding (t-SNE) of the embeddings for the feature "callsign" was computed and can be seen in figure 3.1. This visualization illustrates how the data points are distributed. While it doesn't retain information about the distances between points, it does indicate how the data points are distributed in the higher dimensional space and how clusters are formed. An example of a cluster is the data points of t-SNE Dimension $1 > 70$. These data points have a different distribution of runway assignment, as illustrated in figure 3.2. They have a lower rate of being assigned the majority class runway "RW29" in favor of the less frequent runways "RW16" and "RW36". Flights in this cluster are 6.5 times more likely to use deicing and have 1.67 times longer mean taxi time. A model recognizing this or similar clusters is expected to have a better performance.

3.5 Machine Learning Techniques

The list of algorithms was informed by the literature. We considered the models which were previously tried on similar use cases and generated the list of algorithms for this thesis. These considerations were based on the performance as well as on the experiment setup. SVM and kNN, although featured in some of the available literature, were not included becuase of the datasets size and hardware constraints. Diana measured a comparatively bad performance of SVM on the task of taxi time prediction [Dia18]. In a 2022 paper, Grinsztajn *et al.* compared the performance of tree-based and neural network based models on tabular data. In their experiments they investigated the impact of removing features and adding random features. They found an overall better performance of tree based models. As the reason for this superior performance, they concluded that "irregular patterns in the target function, uninformative features, and non rotationally invariant data where linear combinations of features misrepresent the information" [GOV22]. Based on a report on the current state of competitive machine learning, we placed a special emphasis on tree-based ensemble models [Car23].

The list of algorithms applied on the prediction tasks of this thesis consists of:

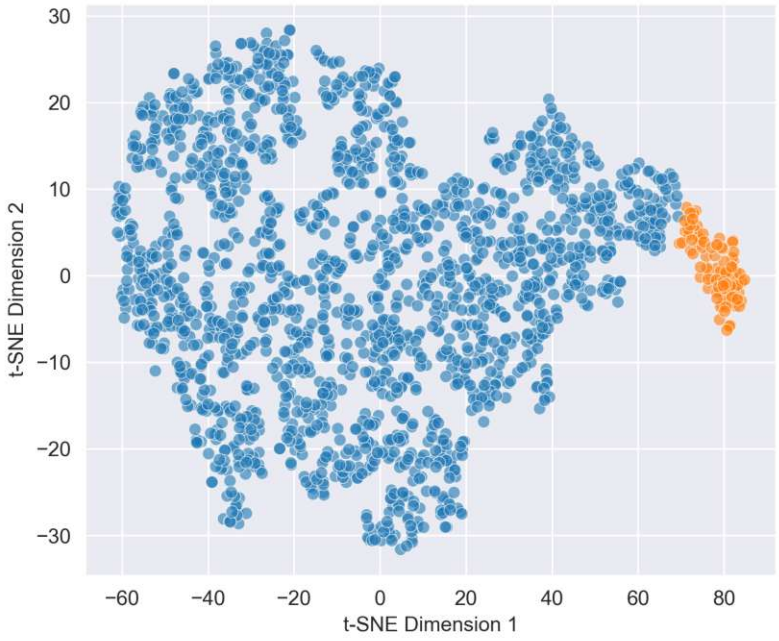

Figure 3.1: t-SNE of callsign embeddings, one cluster highlighted

Figure 3.2: t-SNE distribution of runway assignment comparing entire development dataset to a cluster in the t-SNE visualization

Decision Tree (DT) [Qui86]: A DT recursively splits the dataset into subsets. The resulting tree-like structure contains several branches for the splits and a prediction on each of the leaves. Important parameters of DT are the criterion for making a split, the maximum number of features to consider for the split, and features that limit the size of the tree and prevent overfitting, such as the maximum depth, the minimum number of samples for making another split or the minimum number of samples on a leaf. The advantages of a DT include interpretability, the ability to visualize the model and a low computational cost. Disadvantages include the possibility of overfitting and small variations in the training data resulting in very different trees. A popular solution to mitigate these disadvantages is to create ensembles of trees.

Random Forest (RF) [Bre01]: A RF is an ensemble of DTs, trained on random subsets of the instances of features of the original dataset. The number of trees is an important parameter, in addition to the parameters of the DT.

Adaptive Boosting (AdaBoost) [FS97]: This algorithm is an iteratively generated ensemble of weaker estimators, such as DT. Each instance of the training data has a weight assigned. With each iteration, the weights are adjusted to increase for wrong predictions and decrease for correct predictions. Therefore, difficult samples gain more weight and influence with successive iterations. The learning rate is an important parameter, along with the number of estimators.

Bagging Estimator (Bagging) [Bre04]: A Bagging Estimator (Bagging) predictor is a general ensemble of predictors. As an ensemble of DT it shares similarities with a RF, but uses the entire set of features for each estimator. It therefore has less randomization.

Extremely Randomized Trees (Extra-Trees) [GEW06]: This algorithm is based on RF with the difference of the split being generated randomly for each feature. Compared to RF, Extra-Trees usually results in larger models with more leaf nodes, but has a lower computational time.

Gradient Boosting (GradBoost) [Fri01]: Gradient Tree Boosting is an iteratively generated ensemble of weak estimators, where each successive estimator is trained to predict the error of the current model.

Extreme Gradient Boosting (XGBoost) [CG16]: The XGBoost builds on Gradient Boosting Machine (GradBoost) and contains some additional features, such as regularization of the number of leaves in each DT, early stopping or parallelization.

3.5.1 Scaling

A preprocessing step often necessary or beneficial for ML is scaling the dataset. Several studies are using Minmax scaling and outlier removal. However, a large part of the available literature does not specify the preprocessing. In addition to Minmax scaling, we added Standard scaling, and Quantile transformer to the list of preprocessing, we tested when applying ML to the prediction tasks. We identified outliers, and removed them only in case they came from an error in the dataset.

3.5.2 Evaluation Metrics

Evaluation Metrics are used to assess the performance of a prediction model. In this thesis, we examine the use of ML models on different classification and regression tasks and evaluate them. This raises the necessity of choosing appropriate metrics for the different tasks. The choice of the performance metric varies with the use case and depends on the dataset as well as on the practical application of the model. In the following section, a selection of performance metrics is discussed. This selection is based on the literature review in chapter 2.

Regression: Taxi time Prediction

In the regression tasks of taxi time prediction, the most used metrics are RMSE, Mean Absolute Error (MAE), and ACC within a certain time frame Accuracy within 1 *min* (ACC±1*min*), Accuracy within 3 *min* (ACC±3*min*), Accuracy within 5 *min* (ACC±5*min*).

RMSE and MAE are based on the deviation of the predicted values from the actual values. RMSE measures the root of the average squared deviation, as in equation 3.2. This way large deviations have a disproportionate effect on RMSE compared to small deviations. MAE is the mean of the absolute values of deviations as shown in equation 3.3. An increase in the deviation of any predicted taxi time from the true value has the same effect on the overall MAE, irrespective of the magnitude of the deviation. Therefore, large deviations don't have a disproportional effect.

$$RMSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{3.2}$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N} \mid y_i - \hat{y}_i \mid \tag{3.3}$$

For instance, an increase in prediction error from 20 to 21 min has a much larger impact on RMSE than an increase from 1 to 2 min. In MAE, the increase in both cases has the same effect on the metric.

According to the domain experts consulted during this thesis, small deviations of the predictions from the actual taxi time can be tolerated. It is more important, that the majority of prediction errors stay within a certain threshold. This makes the ACC within a margin a useful metric.

Classification: Runway Assignment, Deicing Usage

ACC is widely used in tasks of Runway Configuration Prediction. One major drawback with this metric is the "Accuracy Paradox" [BDMS20]. This describes how in highly imbalanced datasets, a classifier which only predicts the majority class, can achieve high

scores without being of any actual use. The distributions of Runway assignments in the given datasets of this thesis are imbalanced. For this reason, ACC will not be used as the main classification metric in this thesis. It will, however, still be evaluated to compare our results to the literature. Some studies worked around the problem of imbalanced data sets by filtering all minority classes and defining the classification task as only detecting one class out of a subset of classes [Wan11]. This approach will not be used in this thesis, since it won't reflect the actual real world use case sufficiently. The models were not tuned to optimize for ACC.

Another popular choice for evaluating the performance of a model in a classification task is F1-score. This metric is based on Precision and Recall, which in turn are based in True Positive (TP), False Negative (FN) and False Positive (FP). Precision is the ratio of correct classifications among all positively classified instances, as shown in equation 3.4. Recall is the ratio of positive instances that were recognised correctly, as shown in equation 3.5. A classifier which is optimized for precision, will be very selective and will only classify instances as positive if there is a high confidence in the prediction. This classifier will accept missing some positive instances to avoid false negatives. Such a classifier has high specificity. In contrast to that, if a classifier is optimized for recall, it will be more likely to classify instances as positive, even at the cost of getting some false positives. This is called high sensitivity. There is a trade-off between these two metrics. The F1-Score is the harmonic mean between the two, as shown in equation 3.6. To achieve a high F1-Score, a classifier needs to balance precision and recall, or specificity and sensitivity. If either of them is very low, the F1-Score will be low.

$$Precision = \frac{TP}{TP + FP} \tag{3.4}$$

$$Recall = \frac{TP}{TP + FN} \tag{3.5}$$

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{3.6}$$

Precision, Recall, and F1-score are defined for binary classification. For multi-class problems, there are various methods to decompose the problem into binary classification tasks and average the results. One such way is micro-averaging, where each instance of the data set has the same weight, independent of the class balance. Here the TP's, FN's, and FP's are summed up across all instances and Precision, Recall and F1-score are computed. Another strategy is macro-averaging, where Precision, Recall, and F1-score are computed per class and averaged. This way, each class has the same weight.

In the case of a balanced dataset, micro and macro average yield relatively similar results. In the case of an imbalanced dataset, the majority classes have a disproportionate effect on the micro-average, while minority classes have a comparatively low effect. In

macro-average, each class contributes equally to the overall score, independent of the number of instances. For the task of predicting the runway assignment, it is important to predict all runways, independent of the imbalance. For this reason, macro-average will be used as the main evaluation metric of the runway assignment prediction.

3.5.3 Hyperparameter Tuning

The strategy for the hyperparameter optimization was largely informed by Yang and Shami [YS20] and Bischl *et al.* [BBL+23]. In their papers, they provided a comprehensive overview of the most common ML Algorithms, along with their most important hyperparameters and recommended search spaces. Appropriate methods for optimizing the hyperparameters were given for each algorithm. These methods depend largely on the type of hyperparameters.

As the method for finding the best hyperparameters, we chose a Bayesian optimization with a tree-structured parzen estimator. The search spaces for each algorithm are synthesized from [YS20], [BBL+23] and [PBB19] and give in table 3.1. Grinsztajn *et al.* [GOV22] used 400 iterations for their hyperparameter tuning with a random search strategy. In our search strategy, we expect a quicker conversion, but this number serves as an orientation.

3.5.4 Automated Machine Learning

In addition to the hyperparameter tuning, we decided to apply the automated ML tool TPOT [OBUM16]. This uses a genetic algorithm to optimize a ML pipeline. The pipeline includes different methods for scaling, dimensionality reduction, feature selection, different estimators and hyperparameter optimization. The algorithm starts out with a population of pipelines as the first generation. After applying all to the dataset, the pipelines are evaluated using CV. A new generation is generated by using attributes of the best performing pipelines and introducing random mutations. This new generation is applied to the dataset and the cycle is repeated until the specified maximum number of generations is reached or an early stopping is triggered.

3.5.5 Feature Importance

During data analysis, mutual information was used to estimate the feature importance for each prediction task. This measures the dependence of pairs of variables. While it doesn't capture more complex relations, it allows for estimating the feature importance independent of a model.

Once the models were developed, we quantified the feature importance using permutation importance. In this method, the models, which were trained on the development set, are applied to make predictions on the validation set. Then each feature of the validation set is randomly shuffled and the drop off in performance is measured [Bre01]. Multiple rounds of shuffling each feature allow for a more robust measurement, analogous to CV.

ML Model	Hyperparameter	Type	Distribution	Search Space
DT	max_depth	Discrete	Uniform	$[1, 100]$
	criterion	Categorical		Reg: [sq error, friedman, poisson]
				Clf: [gini, entropy, log loss]
	max_features	Discrete	Uniform	$[1, p]$
	min_samples_split	Discrete	Uniform	$[2, 11]$
	min_samples_leaf	Discrete	Uniform	$[1, 11]$
RF	n_estimators	Discrete	Uniform	$[10, 500]$
	max_depth	Discrete	Uniform	$[1, 100]$
	criterion	Categorical		Reg: [sq error, friedman, poisson]
				Clf: [gini, entropy, log loss]
	max_features	Discrete	Uniform	$[1, p]$
	min_samples_split	Discrete	Uniform	$[2, 11]$
	min_samples_leaf	Discrete	Uniform	$[1, 11]$
AdaBoost	n_estimators	Discrete	Uniform	$[10, 500]$
	learning_rate	Continuous	Log-uniform	$[2^{-12}, 2^{12}]$
Bagging	n_estimators	Discrete	Uniform	$[10, 500]$
Extra-Trees	n_estimators	Discrete	Uniform	$[10, 500]$
	max_depth	Discrete	Uniform	$[1, 100]$
	criterion	Categorical		Reg: [sq error, friedman, poisson]
				Clf: [gini, entropy, log loss]
	max_features	Discrete	Uniform	$[1, p]$
	min_samples_split	Discrete	Uniform	$[2, 11]$
	min_samples_leaf	Discrete	Uniform	$[1, 11]$
GradBoost	n_estimators	Discrete	Uniform	$[10, 500]$
	learning_rate	Continuous	Log-uniform	$[2^{-12}, 2^{12}]$
	max_depth	Discrete	Uniform	$[1, 100]$
	criterion	Categorical		Reg: [sq error, friedman]
				Clf: [gini, entropy, log loss]
	max_features	Discrete	Uniform	$[1, p]$
	min_samples_split	Discrete	Uniform	$[2, 11]$
	min_samples_leaf	Discrete	Uniform	$[1, 11]$
XGBoost	n_estimators	Discrete	Uniform	$[10, 500]$
	max_depth	Discrete	Uniform	$[1, 100]$
	learning_rate	Continuous	Log-uniform	$[2^{-12}, 2^{12}]$

Table 3.1: Important hyperparameters and their recommended search spaces for a selection of algorithms. “clf” and “reg” indicate a hyperparameter is only relevant for classification or regression.

Aside from identifying the most important features, this method also allows for identifying features which have a negative impact on model performance and should therefore be removed from the dataset. The quality of the result of permutation dependence depends

on the quality of the model, and how well it performs on the dataset.

CHAPTER 4

Data Analysis

Numerous factors have an influence on taxi time, runway assignment and deicing usage. This chapter offers a statistical analysis of the relevant variables, aiming to create an insight into the dataset. Analysing distributions of different features in the datasets also allows for making an informed decision about the necessity of further preprocessing and scaling.

This chapter will start with a general overview and description of the dataset. Then it will continue to focus specifically on the weather data and evaluation of the weather forecast. Finally, the dependence between the predictor and target variables will be analysed using mutual information.

4.1 Describing the Data Set

The combined dataset of flights and weather forecasts contains 1,299,770 records for 2018 and 1,663,502 for 2019. For each of the 120,490 flights in 2018, an average 10.8 weather reports are available. These consist of an average of 9.8 TAF entries of various forecast horizons and one METAR. For each of the 133,396 flights in 2019, there is an average of 12.5 weather reports.

The distribution of taxi time approximates a positively skewed normal distribution. The 2018 dataset has a mean taxi time of $9.96min$, with a standard deviation of $5.04min$ and a skewness of 2.66. The median taxi time is $9min$. The 2019 dataset has an increased average taxi time, with a mean of $10.16min$ and a median of $10min$, and a slightly lower standard deviation of $4.59min$ and skewness of 2.63. To visualise the influence of runway assignment on taxi time, figure 4.1 displays the taxi time distributions across the different runways. This indicates that Runway 29 typically yields shorter taxi times, especially in comparison to the longer taxi times of Runway 34. One underlying cause for this disparity are the different taxi distances of both runways. This supports the findings

of Xia and Huang [XH22], who reported a weak correlation between taxi time and taxi distance. Each of the distributions is normalized. This indicates that Runway 11, while having a lower peak, has a wider distribution compared to Runway 29, which has a higher peak and a narrower distribution.

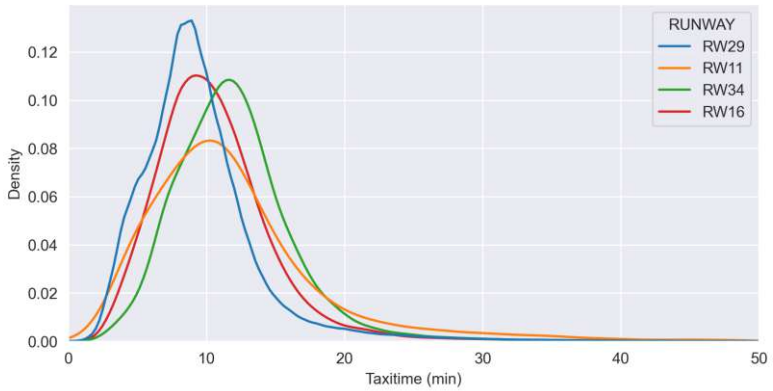

Figure 4.1: Taxi time distributions across runways

Another notable factor that influences the taxi time is the aircraft size. The different sizes are grouped into six categories (A-F), where five of them (B-F) are present in the dataset of this thesis. Figure 4.2 presents the distribution of taxi times across varying aircraft sizes. The figure illustrates that smaller aircraft, represented by letters earlier in the alphabet, tend exhibit shorter taxi times.

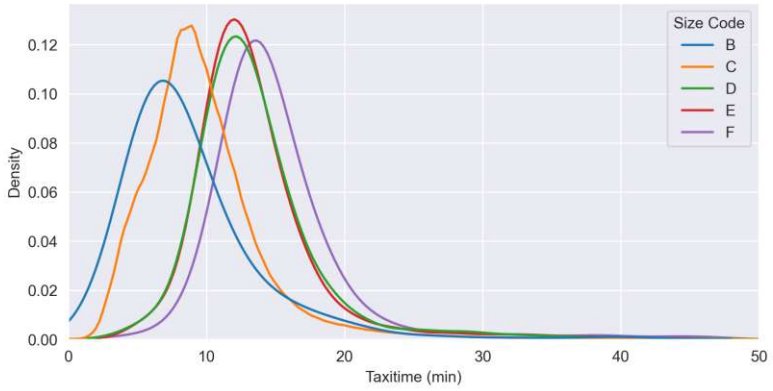

Figure 4.2: Taxi time distributions across size codes

Figure 4.3 displays the influence of deicing usage on taxi time. Aircraft using the deicing services have a mean taxi time of 20.7 *min*, which is approximately double the mean taxi time observed if these services are not used. While deicing usage only occurs in

3.4% of 2018 flights and 2% of 2019 flights, the impact is so large that it motivates the investigation of using machine learning to predict it.

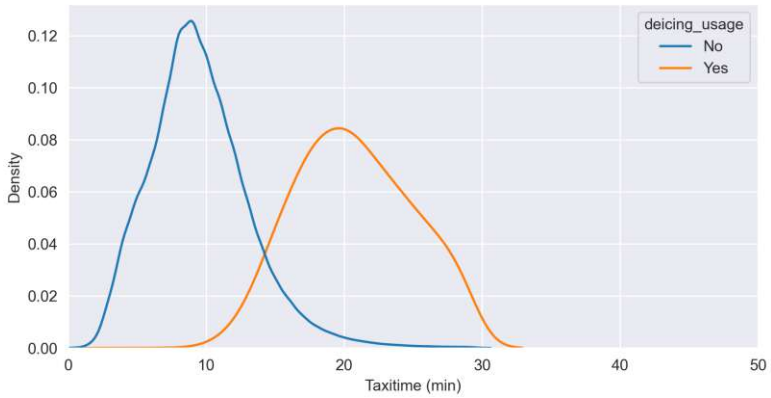

Figure 4.3: Taxi time distributions depending on deicing usage

Figure 4.4 compares the change of average taxi time and the number of flights over a single, randomly selected, day. The two variables are weakly correlated with a Spearman rank correlation coefficient of +0.15. The plot illustrates varying levels of demand over the day, with multiple peaks.

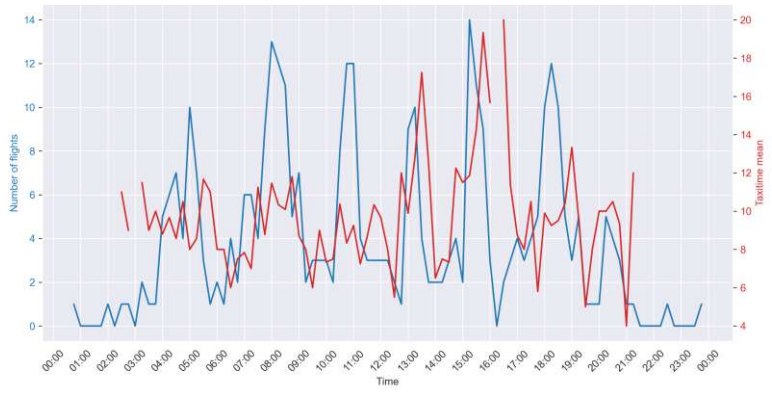

Figure 4.4: Taxi time averages in 15min intervals on 2018-06-26

Figure 4.5 shows the average taxi time over the course of a day. Lower values can be observed in the early morning and late night. Higher levels and peaks occur throughout the day. The peaks are clearer pronounced in the example of a single day compared to the average across all days. This observation suggests that periods of increased taxi times fluctuate throughout the day and year, which raises the need for including features into the dataset which contain temporal information.

Figure 4.5: Taxi time averages over daytime

Figure 4.6 presents the mean taxi time across all days of the dataset. Various changes and peaks are visible throughout the year. Some of them occur in both years, such as an increase in the beginning of the year, while others are only visible in one year, such as the peak in the beginning of October in 2019. Changes across larger time spans, especially differences between 2018 and 2019 will impact the model performance, since they contain differences between the development and evaluation set. This is relevant for the results of this thesis, as well as for a real-world application, where the model is trained on historic data and applied on real time data.

4.2 Evaluating Weather Forecast

Several previous studies use the observed weather instead of weather forecasts for the prediction of taxi time or runway related variables. To determine how comparable these results are, it is useful to analyse the relation of those two data sources. There are few features which depend on the forecast horizon, with features related to the weather making up the majority of them. If a model performance has a drop-off with an increase in forecast horizon, as observed by Khater *et al.* [RKC21], then the predictive power of the weather forecast likely plays a major role in that.

Figure 4.7 displays the change in Spearman rank correlation coefficients between forecast and observed weather over forecast horizon for a selection of numeric weather attributes. The plot demonstrates a strong correlation of approximately 0.8 for visibility and wind speed at a forecast horizon of 1 hour. This correlation drops off with an increase in forecast horizon to around 0.6 at 30 hours. The drop off is slightly steeper in the first half of the forecast horizon and tapers off after that. The temporary wind speed shows a

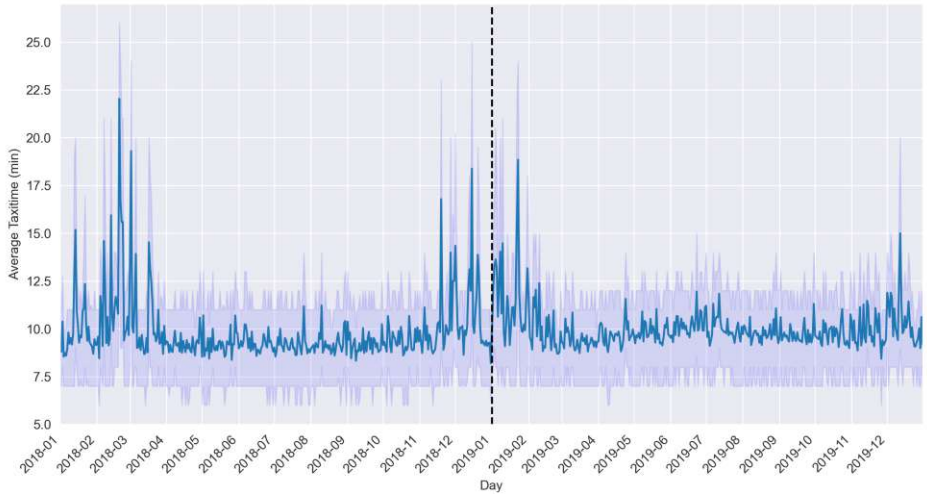

Figure 4.6: Taxi time averages per day

similar pattern. It starts at a correlation of around 0.5 at 1 hour forecast horizon and drops off to around 0.3 for 30 hours. The correlations of wind gusts and temporary wind gusts are below 0.3 for the entire forecast horizon.

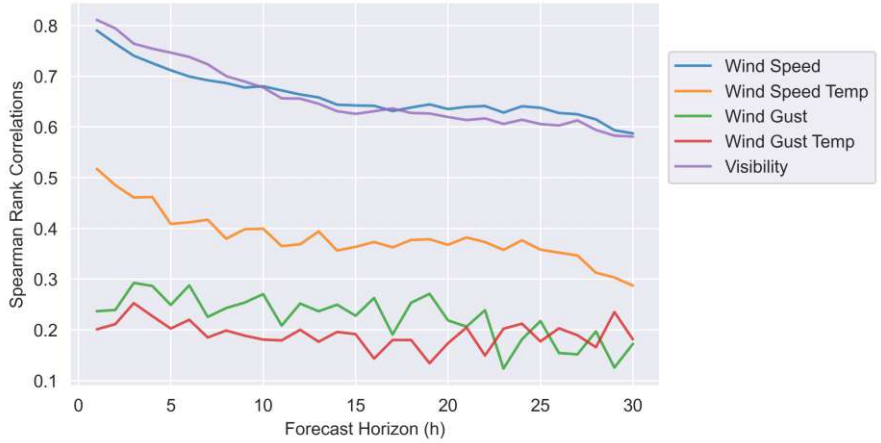

Figure 4.7: Spearman rank correlations of weather forecast and observed weather over forecast horizon

Each TAF has a temperature minimum and maximum for the entire validity of the forecast. Figure 4.8 shows the number of observed temperatures that are between those bounds. The predictions remain stable across the forecast horizon, with approximately 90% of observed temperatures falling within the lower and upper bounds.

The weather phenomena are encoded as acronyms. For example, rain is encoded as

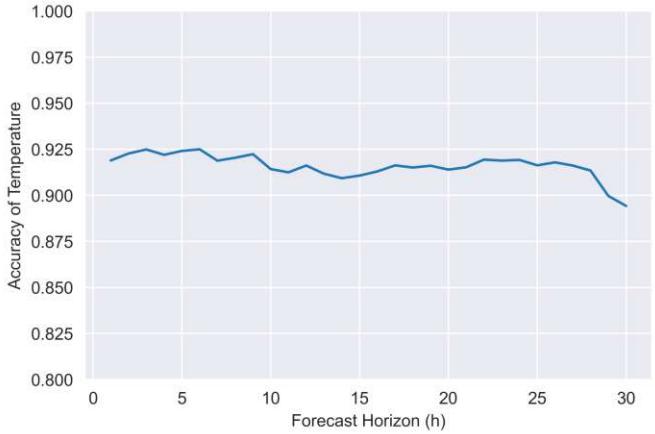

Figure 4.8: Ratio of observed temperatures within forecast temperature range

"RA", while snow is "SN". The precision of the forecast of these phenomena over forecast horizon is shown in figure 4.10. The plot doesn't reveal a strong dependency on the forecast horizon. The SN prediction has the highest precision with approximately 0.6 for most forecast horizons. The precision of RA and BR is below that at approximately 0.4 and DZ and FG have even lower precision across wide ranges of the forecast horizon. Recall in figure 4.11 and F1-Score in figure 4.12 show similar results.

Figure 4.9 shows a confusion matrix of the predicted and observed weather phenomena. The correctly predicted phenomena are visible in the diagonal of the matrix.

	2018			2019		
	mean	std	median	mean	std	median
wind_speed	8.75	4.76	8	9.39	5.33	9
wind_gust	0.42	3.57	0	0.81	5.03	0
visibility	9288	1936	10000	9569	1610	10000
vertical_visibility	1990	133	2000	1989	145	2000
dew_point	7.26	7.29	8	6.64	6.89	7
selected_phenomena	0.04	0.18	0	0.01	0.12	0

Table 4.1: Differences in the distributions of weather predictions across the years

4.3 Evaluating Predictor Variables

Xia and Huang [XH22] found a strong correlation of above 0.6 between the number of departure flights taxiing and the taxi-out time on a major hub airport in central and southern China. This correlation could not be found in the dataset of this thesis. The absolute values of the Spearman rank correlation coefficient between incoming and outgoing demand in the 10-minute and 30-minute windows were below 0.05 for both the

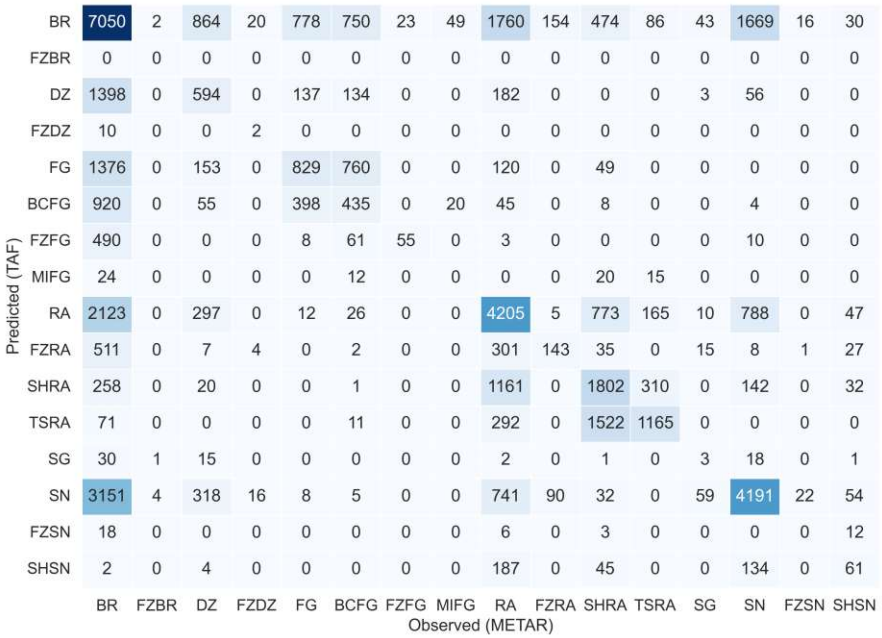

Figure 4.9: Confusion matrix of predicted and observed weather phenomena

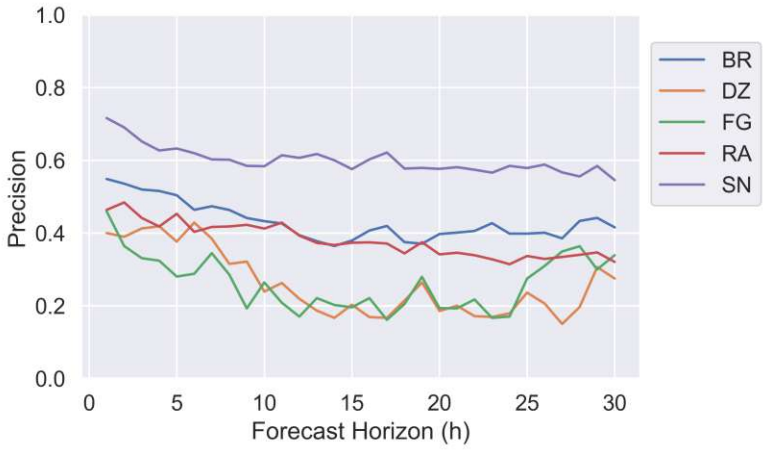

Figure 4.10: Precision of selected phenomena over forecast horizon

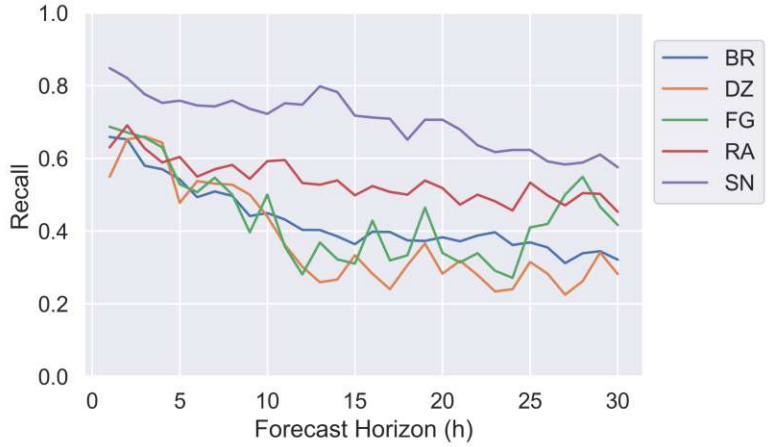

Figure 4.11: Recall of selected phenomena over forecast horizon

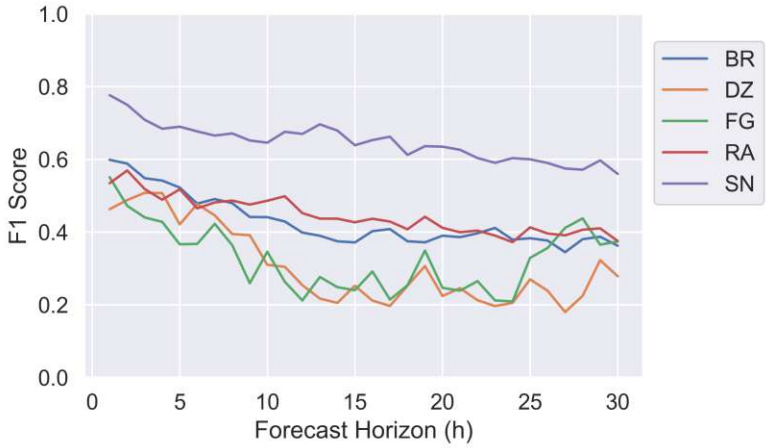

Figure 4.12: F1-Score of selected phenomena over forecast horizon

2018 and the 2019 datasets. A potential reason for this discrepancy, could be congestions forming in the airport that Xia and Huang studied. Such congestions might occur much less frequent at Vienna Airport. In the case of congestion, taxi time increases due to the aircraft spending time in a queue. The varying impact of demand on taxi time suggests that these findings are dependent on the specific airport and don't generalize well across airports.

To quantify the dependence of different predictor variables on the target variables, we analysed the mutual information. The results are visualized in figure 4.13. The visualization includes all features above a value of 0.01 for any of the target variables. It shows that the monthly and yearly ratios of runway assignments for both incoming and outgoing flights have the largest impact on runway assignment. Furthermore, wind related features, and features containing information about the flight destination play a

role. Features related to the payload or seat capacity, as well as the current demand and the day of month, play a smaller role.

For taxi time, the features with the highest mutual information are related to the position and gate usage, the payload and number of passengers and the destination. The monthly and yearly runway usage play a role too, which is likely related to the impact of the runway assignment on the taxi time, as shown in figure 4.1.

In the case of deicing usage, runway assignment features also play a significant role. There is no obvious explanation for this, since deicing usage does not depend on noise abatement measures. Other relevant features are related to the season, the size of an aircraft, and the weather. Of the weather-related features, the most important ones are temperature, dew point, visibility, and a group of selected weather phenomena.

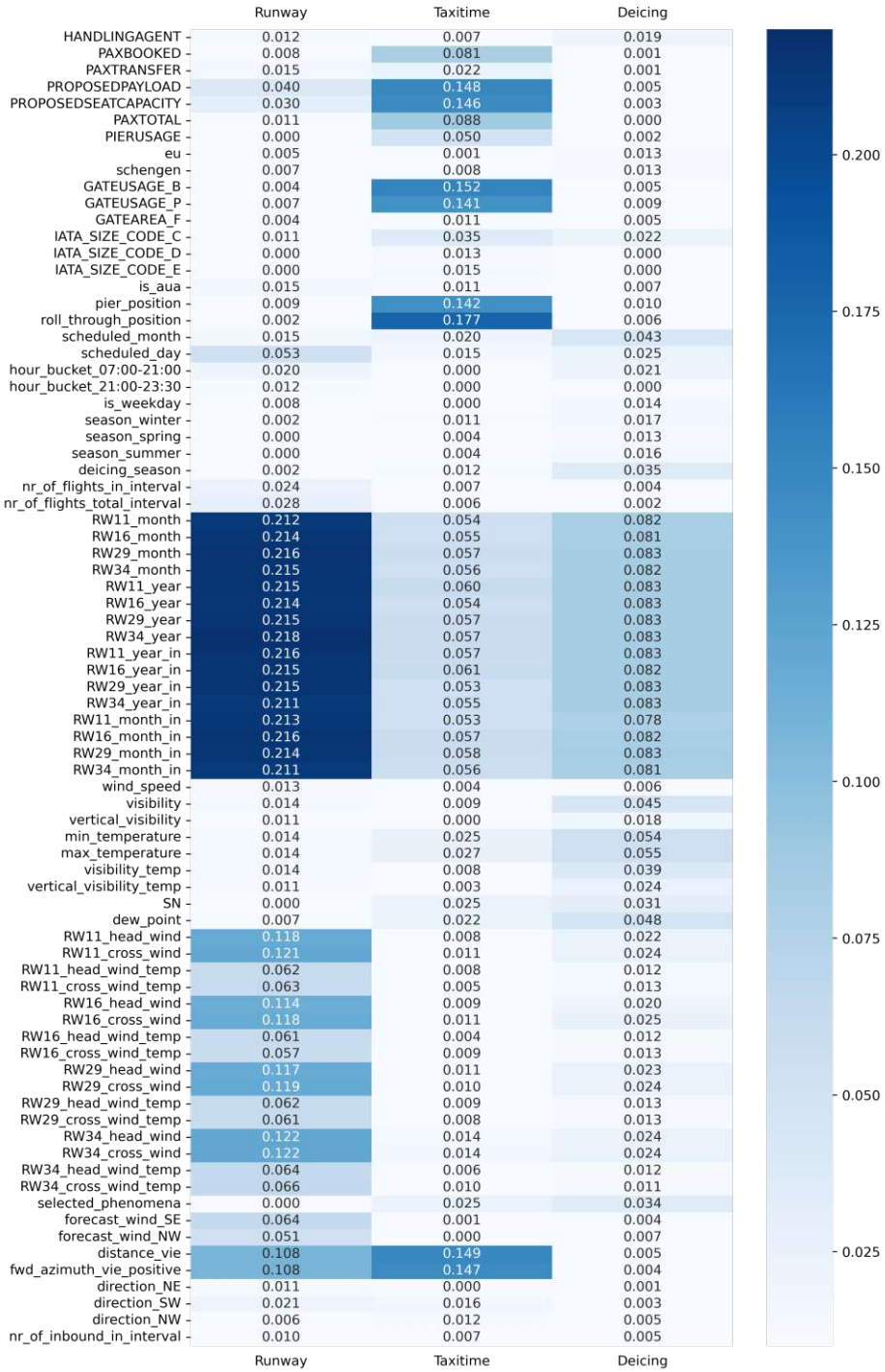

Figure 4.13: Mutual information of the most important features

CHAPTER 5

Experimental Results

This chapter begins by outlining the experiment setup and continues with the results of the various prediction tasks. The primary objective of this thesis is predicting the taxi time of outgoing flights. Additional tasks are predicting the runway assignment and deicing usage, as they have a considerable impact on taxi time. For each task multiple models are trained on the development set and subsequently benchmarked against each other and against a baseline model using the evaluation set. The models are assessed in different scenarios, using specific subsets of the features or data points to replicate real-world deployment scenarios and for comparison with the existing literature. Lastly, the feature importances are analysed for each task.

5.1 Experiment Setup

All experiments were run on a Lenovo Thinkpad X1 Carbon Generation 6 laptop configured with the following specifications:

- Operating System: Microsoft Windows 11 Pro
- Processor: Intel Core i7-8650U
- Memory: 16GB
- GPU: CUDA not enabled

The software environment consisted of:

- Python 3.10.5
- Pandas 1.5.2

- NumPy 1.23.4
- Scikit-Learn 1.2.2
- MLFlow 1.26.1
- XGBoost 1.7.2
- Hyperopt 0.2.7
- TPOT 0.11.7

The models were trained and optimized using the 2018 dataset and evaluated with the 2019 dataset, ensuring a clear separation of the training and evaluation datasets. This emulates a real-world application, where the training data is created prior to the data for which the model will be deployed. This applies to all scenarios, except where it is explicitly stated otherwise.

Hyperparameter Optimization

The systematic optimization of hyperparameters was executed on the development set using 10-fold CV. As an example, figure 5.1 depicts the outcomes of 500 iterations Hyperparameter Optimization (HPO) on a DT regressor for taxi time prediction, each using a distinct hyperparameter combination. The best performing configuration was found in run number 368 with a mean RMSE of 4.01min.

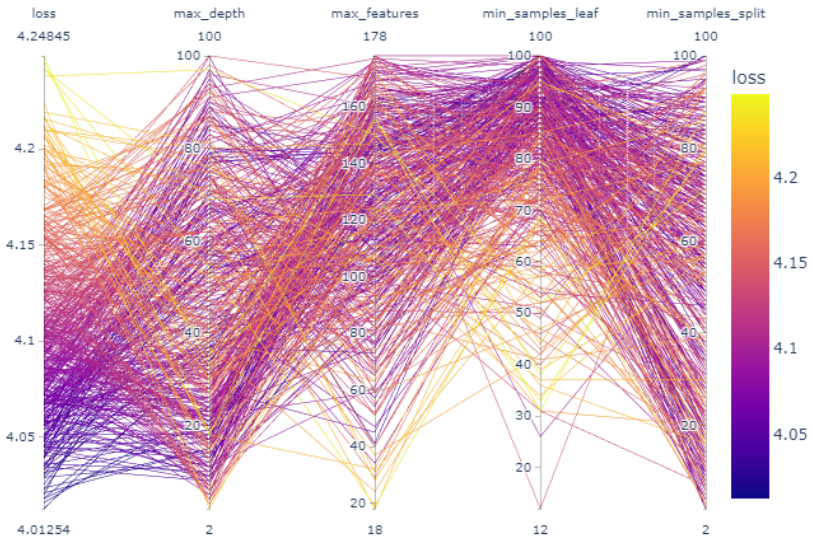

Figure 5.1: Hyperparameter optimization of a DT

Table 5.1 displays the best performing hyperparameters of the same example in the left column. The observed ranges of all combinations with a RMSE in the bottom 10th

percentile are listed in the middle column. In order to gain insight into the distribution of values within those ranges, the median value is presented in the right column. The optimal depth of the tree (max_depth) is 36, with satisfactory results found in the range 8 to 87. Lower values might lead to underfitting, while high values increase the risk of overfitting. The number of features considered for a split (max_features) leading to a good model performance are found in a large range of 41 to 163, with the best performing model at 153. The dataset has 178 predictor features, which is the maximum possible value. The best performing value of min_samples_leaf and min_samples_split are 94 and 69 respectively. All tested scalers and splitting criteria are present in the combinations below the 10th percentile of RMSE. The best performing combination uses the Standard Scaler and friedman_mse as criterion.

	Best Run (368 / 500)	P10 (Lowest, Highest)	P10 Medium
max_depth	36	(8, 87)	39
max_features	153	(41, 163)	129
min_samples_leaf	94	(72, 100)	93
min_samples_split	69	(6, 100)	63

Table 5.1: Best performing hyperparameters of DT

It is important to note that there is interplay between the hyperparameters. For example, a small value of max_depth, or large values of min_samples_leaf and min_samples_split can all act as countermeasures preventing overfitting. The combination of those hyperparameters ultimately determines the performance. The hyperparameter combinations of the best performing models are displayed in table 5.2.

5.2 Taxi time

The performance of our optimized models was assessed on the unseen evaluation set. Among the various evaluation metrics in the literature, RMSE, MAE, and ACC±5min are the most prevalent for taxi time prediction. RMSE and MAE have the same unit as the target variable which makes the result interpretable. Large prediction errors have an outsize effect on RMSE compared to small prediction errors. Depending on the scenario, this can be an advantage if small errors are tolerable, while large errors need to be detected. ACC±5min has the advantage of not being affected by small prediction errors, while simultaneously being robust against outliers.

Evaluation

The taxi time prediction was conducted across four different scenarios, with the models trained and evaluated specifically for each of them. Scenario A is the prediction ahead of time. This was done with a forecast horizon of up to 30 hours, which is the maximum time horizon of the TAF. This scenario only uses information that is available at the time the prediction is generated. In scenario B, the prediction is generated the moment

ML Model	Hyper-parameter	Taxi time	Runway	Deicing
DT	max_depth	36	24	86
	criterion	friedman_mse	log_loss	log_loss
	max_features	153	0.81	0.49
	min_samples_split	69	11	11
	min_samples_leaf	94	11	18
	scaler	Standard	Standard	Standard
RF	n_estimators	377	199	32
	max_depth	39	38	79
	criterion	poisson	entropy	log_loss
	max_features	24	57	56
	min_samples_split	3	3	2
	min_samples_leaf	1	2	1
	scaler	Standard	Standard	Standard
AdaBoost	n_estimators	284	377	375
	learning_rate	0.0001	0.838	1.24
	scaler	Standard	Minmax	Quantile Tr
Bagging	n_estimators	403	281	300
	scaler	Quantile Tr	Standard	Standard
Extra-Trees	n_estimators	145	407	469
	max_depth	59	45	70
	criterion	poisson	gini	gini
	max_features	51	81	88
	min_samples_split	9	3	2
	min_samples_leaf	4	1	1
	scaler	Quantile Tr	Quantile Tr	Quantile Tr
Grad Boost	n_estimators	347	481	283
	learning_rate	0.007	0.053	0.71
	max_depth	72	23	50
	criterion	squared_error	squared_error	squared_error
	max_features	8	55	72
	min_samples_split	11	8	7
	min_samples_leaf	6	8	4
	scaler	Quantile Tr	Standard	Standard
XGBoost	n_estimators	302	384	453
	max_depth	2	36	67
	learning_rate	0.12	0.251	2.02
	scaler	Standard	Standard	Standard

Table 5.2: Results of hyperparameter optimization

the aircraft leaves the parking position. In this scenario, knowledge about the assigned runway, the usage of deicing, as well as the current state of all other aircraft, is assumed. While scenarios A and B differ in the input features used to train the models and make predictions, they are both trained and evaluated on the full datasets containing all available flights. This includes numerous outliers, which have a taxi time that widely deviates from the average. These outliers can for example be caused by deicing usage, extreme or rare weather events. Scenario C excludes data points that are challenging to predict. It uses the same input features as scenario B, but filters flights or entire days that are associated with a lower prediction performance. Scenario D1 and D2 replicate the circumstances of selected papers from the literature, allowing for a better comparison of our models to the state of the art.

For our baseline comparison, we deployed an Ordinary Least Squares (OLS) regression, as used by Ravizza *et al.* [RCA+14] and Chen *et al.* [CRAS11]. Aside from being simple, interpretable, and computationally efficient, this model has the advantage of not relying on a random seed or the choice of appropriate hyperparameters. The same training data therefore, consistently produces the same model.

Scenario A

This scenario simulates the prediction of a flights taxi time in advance without reliance on any information related to the aircraft's parking position or assigned runway. These factors determine the taxi distance, which has a strong influence on taxi time, as shown by *et al.* [RAMB13]. The temporal data is derived from the scheduled times, which are known in advance. All weather information is sourced from the TAF, and therefore contains the uncertainty of the weather forecast, as discussed in chapter 4.

Figure 5.2 presents the evaluation of the models using RMSE. Across all models, there is no distinctive performance drop-off for longer forecast horizons. All models achieve a better score than baseline, with Extra-Trees and RF showing the best performances.

The evaluation using MAE is displayed in figure 5.3. The Extra-Trees model achieves the best score, while XGBoost and Bagging fail to outperform baseline. Contrary to these results, the Bagging model had performed best on the hyperparameter tuning on the development set. This adds to the findings of Yin *et al.* [YHM+18], who observed models exhibiting low performance on the training set but high performance on the validation set. The average taxi time differs between the development set and evaluation set, suggesting that overfitting might be the cause of the Bagging Regressors performance.

Figure 5.4 shows the evaluation results based on ACC$\pm 5min$. All models perform above baseline, with GradBoost and Extra-Trees achieving the best scores across all forecast horizons.

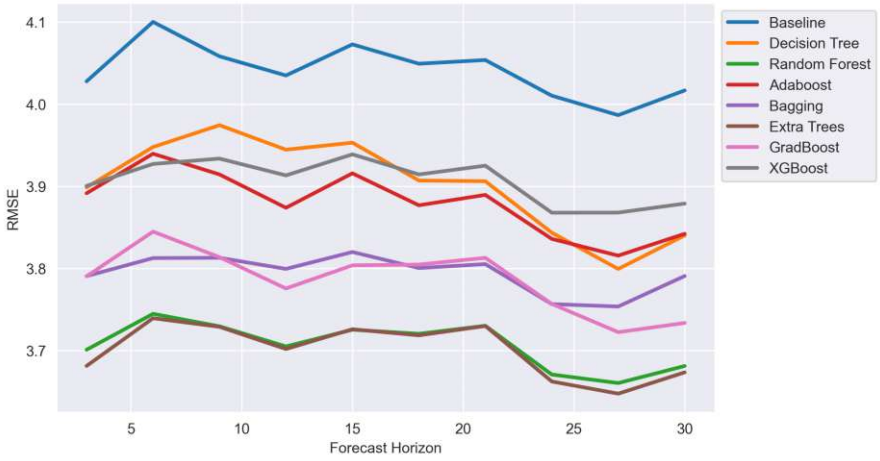

Figure 5.2: Taxi time prediction RMSE over forecast horizon

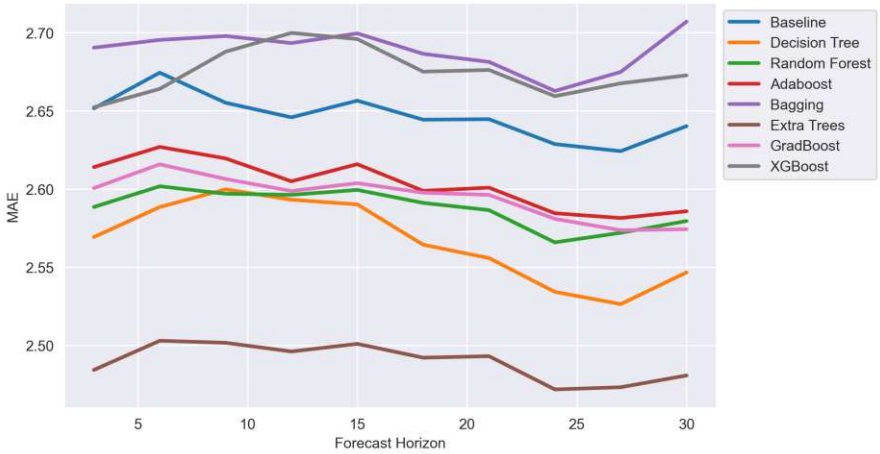

Figure 5.3: Taxi time prediction MAE over forecast horizon

Scenario B

This scenario simulates the prediction of a flight's taxi time at the moment of aircraft block-off. At this point, the information about the aircrafts current location and its assigned runway are available. Additionally, the observed weather taken from METAR is used. While scenario A only uses the demand based on the number of flights scheduled to takeoff or land in each time frame, scenario B relies on actual recorded timestamps. That way, differences between the planned arrival and departure times, compared to the actual times, are accounted for in scenario B. Furthermore, in scenario B the demand is divided by runways since this information is available at that time. Knowledge about the decision of deicing usage is also assumed in scenario B. This contains possible delays, as well as the information about how the incoming and outgoing demand is distributed

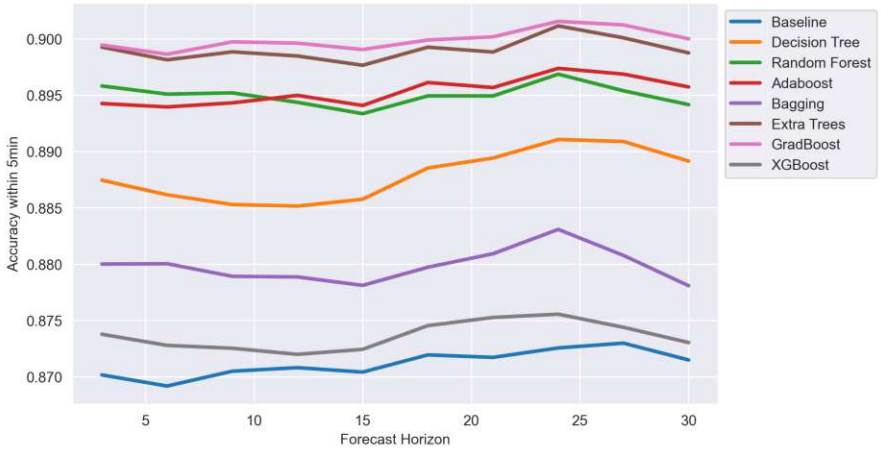

Figure 5.4: Taxi time prediction ACC±5*min* over forecast horizon

among the runways. Table 5.3 presents the performance metrics of the models evaluated in scenario B. Similar as in scenario A, Extra-Trees shows the highest score on most metrics. The RF model achieves the same RMSE as the Extra-Trees. Notably, none of the models outperform baseline in ACC±1*min*.

	RMSE	MAE	ACC±1*min*	ACC±3*min*	ACC±5*min*
Baseline	3.93	2.60	**0.301**	0.718	0.878
DT	3.80	2.52	0.293	0.731	0.892
RF	**3.58**	2.52	0.262	0.708	0.901
AdaBoost	3.81	2.57	0.270	0.729	0.898
Bagging	3.70	2.65	0.251	0.678	0.883
Extra-Trees	**3.58**	**2.43**	0.290	**0.738**	**0.904**
GradBoost	3.69	2.55	0.263	0.710	0.903
XGBoost	3.76	2.57	0.292	0.706	0.879

Table 5.3: Taxi time evaluation scenario B

Figure 5.5 illustrates the predictions of the Extra-Trees regressor over the observed values on the validation set.

Scenario C

This scenario applies the same features as scenario B, but limits the instances to a specific subset. Flights that are difficult to predict were filtered from the dataset. This includes VIP flights, flights without passengers, flights on days where deicing is being used and days of bad weather conditions. The remaining dataset has circumstances that are ideally suited for predicting taxi times. In a potential deployment, the ML model could be applied only for these types of flights, capitalizing on the benefits of cost-effectiveness and scalability, while human experts focus on the more challenging

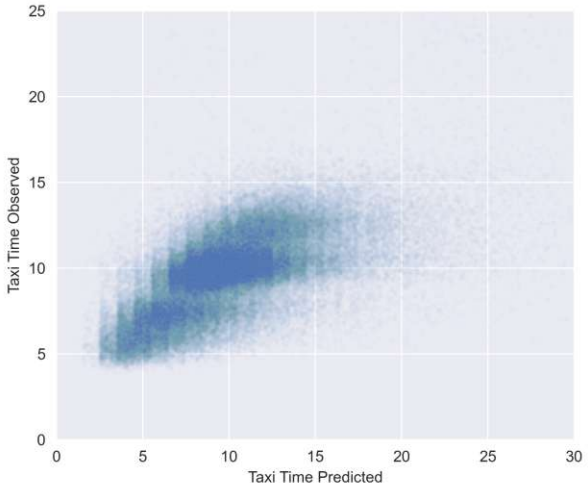

Figure 5.5: Extra-Trees regressor predictions over observed taxi time

scenarios. The model performances of this scenario are improved compared to scenario B. This, however, applies to the baseline as well, lowering the performance gap between our models and the baseline. While in scenario B, the best performing models have a 9% lower RMSE compared to baseline, in scenario C it is only 5% lower. This indicates that in the instances which are difficult to predict, the more complex ensemble models are having an advantage over the linear baseline model.

	RMSE	MAE	ACC±1min	ACC±3min	ACC±5min
Baseline	2.88	2.06	0.333	0.783	0.933
DT	2.90	2.05	0.343	0.782	0.926
RF	**2.72**	1.93	0.352	**0.814**	**0.943**
AdaBoost	2.98	2.13	0.328	0.778	0.925
Bagging	2.88	2.07	0.350	0.782	0.932
Extra-Trees	2.73	1.93	**0.364**	0.809	0.942
GradBoost	2.74	**1.92**	0.363	**0.814**	0.940
XGBoost	2.81	2.00	0.349	0.792	0.935

Table 5.4: Taxi time evaluation scenario C

Scenarios D1 and D2

These scenarios reproduce the conditions detailed in two research papers, allowing for the comparison of our models to the existing literature. Scenario D1 compares to the datasets of Yin *et al.* [YHM+18]. Their models are divided in two cases. Case a uses a training set that consists of one day on October 1st, while the training set of case b uses the entire month of September. Both of their cases are validated on the flights of

October 2nd. The results of comparing our models to Yin *et al.*'s best performing models are shown in table 5.5. Yin *et al.*'s case b model performs best, while all our models show a better performance than their case a.

	RMSE	MAE	ACC±1*min*	ACC±3*min*	ACC±5*min*
Yin *et al.* RF case a	3.92	2.82	0.296	0.646	0.842
Yin *et al.* RF case b	**1.92**	**1.32**	**0.532**	**0.918**	**0.981**
DT	2.42	1.81	0.384	0.803	0.943
RF	2.22	1.68	0.372	0.860	0.963
AdaBoost	2.48	1.86	0.377	0.805	0.951
Bagging	2.46	1.87	0.379	0.818	0.956
Extra-Trees	2.22	1.68	0.387	0.872	0.963
GradBoost	2.26	1.69	0.382	0.842	0.963
XGBoost	2.30	1.76	0.357	0.825	0.966

Table 5.5: Taxi time evaluation scenario D1: comparing to Yin *et al.* [YHM+18]

Scenario D2 emulates the different datasets of Wang *et al.* [WBW+21]. They compared various models and the impact of feature selection on performance. Their study uses datasets from Manchester Airport (MAN), ZRH, and Hong Kong International Airport (HKG). Each dataset has a distinct time frame. We trained and evaluated our models on each specific time frame. HKG does not use deicing services due to the climate of the location. We therefore removed the deicing days for this comparison. Our models show a better performance than the MAN and HKG examples. Conversely, in the ZRH example, Wang *et al.*'s RF model shows better performance.

	RMSE	MAE	ACC±1*min*	ACC±3*min*	ACC±5*min*
Wang *et al.* RF - MAN	3.31	2.25	0.338	0.754	0.899
Extra-Trees	**3.19**	**2.16**	**0.351**	**0.783**	**0.914**
Wang *et al.* RF - ZRH	**3.10**	**1.78**	**0.517**	**0.829**	**0.925**
RF	3.61	2.39	0.323	0.757	0.893
Wang *et al.* RF - HKG	3.10	1.96	0.481	0.778	0.905
GradBoost (no deicing days)	**2.13**	**1.60**	**0.399**	**0.884**	**0.977**

Table 5.6: Taxi time evaluation scenario D2: comparing to Wang *et al.* [WBW+21]

Feature Importance

The feature importance of the best performing model in scenario B was analysed using feature permutation importance. The results are displayed in figure 5.6. The most important feature is an aircraft's current position, followed by weather phenomena, the aircraft size, temperature, the direction of the destination, and the current demand on different runways.

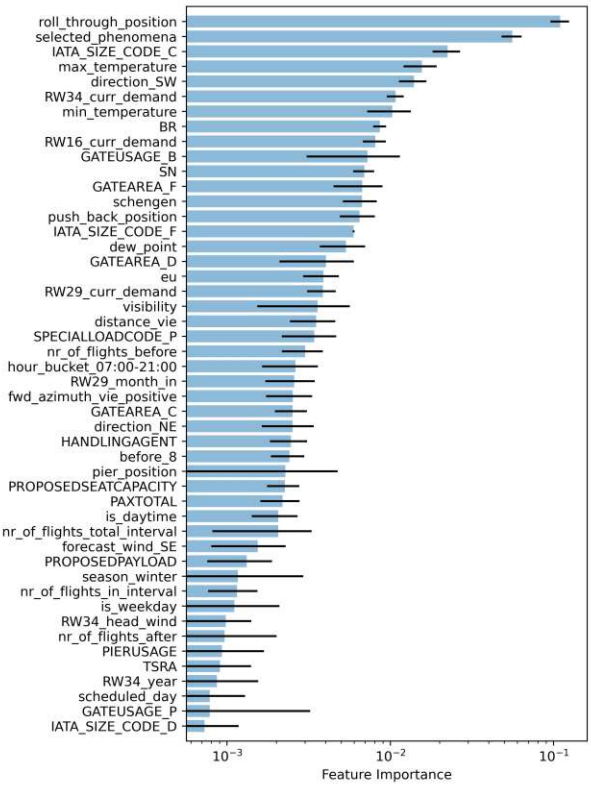

Figure 5.6: Feature permutation importance of taxi time prediction

5.3 Runway Assignment

The runway an aircraft uses for takeoff has a significant impact on taxi time. This motivates the investigation of using ML to predict runway assignment. Notably, while taxi time is a measured time interval, runway assignment is a decision made by human experts. These experts can identify the factors that shape their decisions, but there is no comprehensive set of rules that determines the decision-making process. Instead, their experience and evaluation of the current situation have an impact. The following section evaluates the ability of different ML models in predicting these decisions.

Evaluation

The runway assignment models were evaluated in three scenarios. The scenarios A and B are analogous to the evaluation of taxi time prediction. In both scenarios the flights of

the evaluation set occurred after the flights of the development set. scenario C compares the results to an example of the literature. This includes changing the split between training and evaluation data in a manner in which the flights of the training set and evaluation set are randomly drawn from the same dataset. This type of split led to better results for Raju *et al.* [RMW+21]. The variance in results is caused by a less distinct separation between training and evaluation data.

As the baseline of runway assignment classification, we chose logistic regression for its simplicity and interpretability. This is similar to Churchill *et al.* [CCJ21].

Scenario A

Analogous to taxi time prediction, scenario A represents a prediction ahead of time, using only the information available at the time of prediction. This includes weather forecast and temporal data based on flight schedules, and excludes information about the position and actual demand. Figure 5.7 presents the evaluation of the models measured in macro-average F1-Score. The GradBoost and Extra-Trees show the best performances, followed by RF and XGBoost. All models of this group and to a lesser extent the Bagging model, outperform baseline. A performance drop-off over forecast horizon can be observed. The best models achieve a macro-average F1-Score of 0.68 at a 3 hour forecast horizon and 0.63 at 30 hours, compared to a baseline of 0.55 at 3 hours and 0.51 at 30 hours.

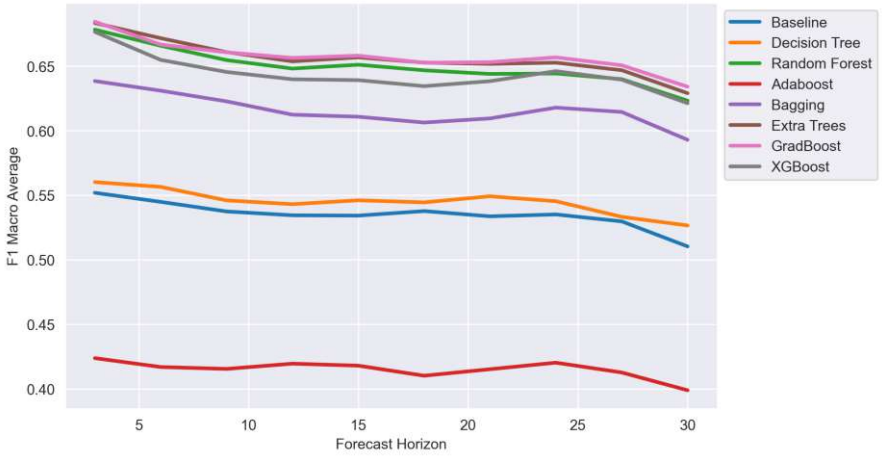

Figure 5.7: Runway assignment macro-average F1-Score over forecast horizon

Figure 5.8 shows the macro-average precision score. The scores of our models are comparable to the F1-Scores but show a smaller difference to baseline. The AdaBoost model shows unstable results, with a high precision for large forecast horizons and scores below baseline for short forecast horizons.

The performances on macro-average recall are displayed in figure 5.9 and show similar trends. The distance of the well performing models to the baseline is greater compared to

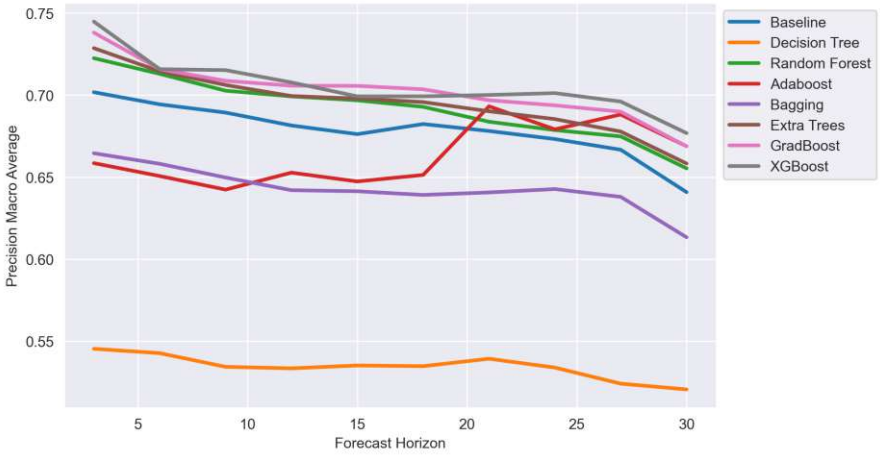

Figure 5.8: Runway assignment macro-average precision over forecast horizon

the results of the precision score. In the ACC score, presented in figure 5.10 the models, which performed best on macro-average F1-Score show very similar ACC scores and are all clearly above baseline. In all metrics, there is a visible performance drop-off with higher forecast horizons.

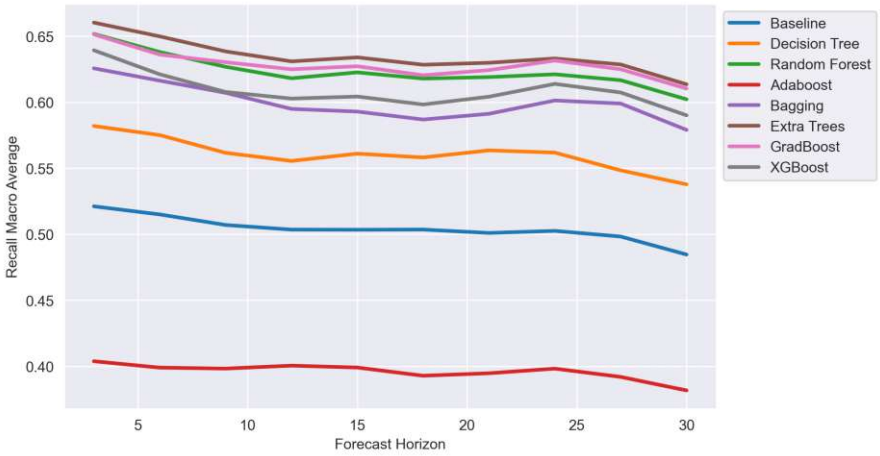

Figure 5.9: Runway assignment macro-average recall over forecast horizon

Scenario B

Scenario B represents a prediction at the time of block-off. Compared to scenario A, this scenario considers the knowledge of the observed weather, the current position of the aircraft, and the actual demand. The results of the evaluation based on Precision, Recall, F1-Score, and ACC are presented in table 5.7.

In general, the performances of the models are higher compared to scenario A. This is true

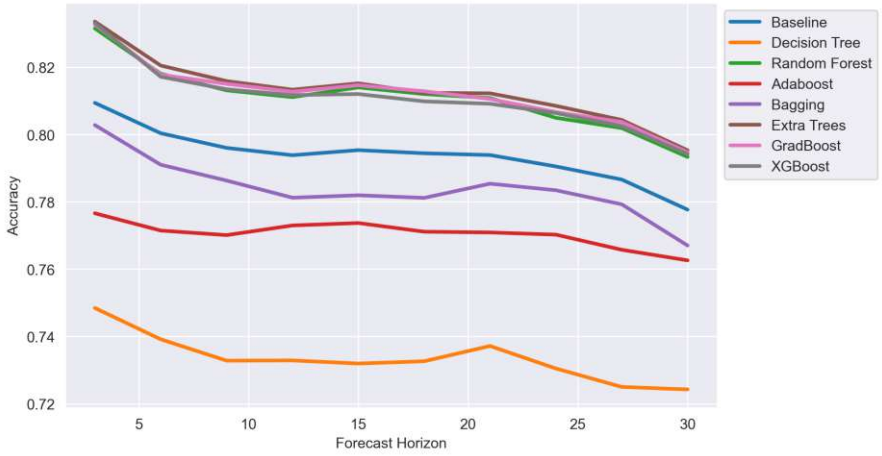

Figure 5.10: Runway assignment ACC over forecast horizon

for the baseline as well. This leads to the conclusion that the additional information of scenario B, and the weather observations instead of forecast contribute to an enhancement in performance. The GradBoost shows the best performance across the majority of scores, with the RF achieving the same ACC score and only XGBoost achieving a higher precision.

	Precision	Recall	F1-Score	ACC
Baseline	0.727	0.535	0.573	0.822
DT	0.592	0.595	0.593	0.771
RF	0.741	0.666	0.695	**0.850**
AdaBoost	0.667	0.405	0.431	0.775
Bagging	0.694	0.634	0.656	0.815
Extra-Trees	0.738	0.671	0.695	0.848
GradBoost	0.750	**0.677**	**0.707**	**0.850**
XGBoost	**0.753**	0.663	0.698	0.849

Table 5.7: Runway assignment scenario B: prediction at block-off time

Figure 5.11 shows the confusion matrix of the best performing runway assignment classification in scenario B.

Scenario C

Churchill *et al.* [CCJ21] primarily trained and evaluated their models on 2020 datasets. They acknowledged that the lowered demand, resulting from the pandemic, had an impact on airport operations, and models might not generalize. Therefore, they trained a model on a 2019 DFW dataset, which in turn makes a comparison to our models possible on similar time frames. In scenario C, we emulated the train test split of the Churchill *et al.* paper and compared the model performances. As a caveat to this comparison, it should be noted that DFW has a greater number of runways. However, the distribution

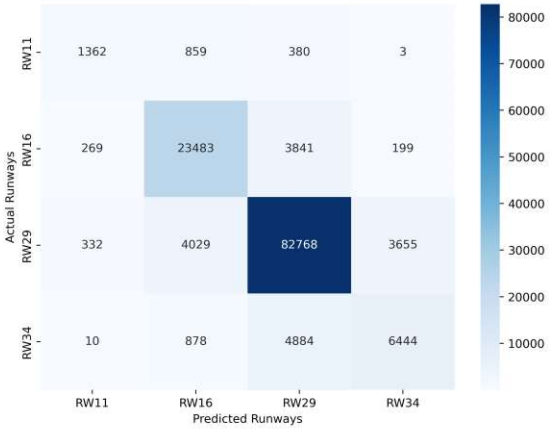

Figure 5.11: Confusion matrix of runway assignment prediction

of the runway assignments is very imbalanced, with several runways being assigned rarely or never. The results are displayed in table 5.8. XGBoost shows the highest ACC score, compared to the rest of our models and the result from the reference paper. XGBoost also scores highest in macro-average F1-Score, although this metric does not allow for a direct comparison to the reference paper, since they did not include F1-Score. When comparing Precision and Recall, it is likely that the F1-Score of their model is higher compared to our XGBoost. Our RF model scored highest in precision. None of our models outperformed Churchill *et al.*'s model in recall.

	Precision	Recall	F1-Score	ACC
Churchill *et al.* XGBoost	0.841	**0.851**	-	0.851
DT	0.706	0.722	0.714	0.849
RF	**0.878**	0.693	0.755	0.892
AdaBoost	0.537	0.619	0.562	0.751
Bagging	0.831	0.742	0.779	0.891
Extra-Trees	0.871	0.695	0.751	0.892
GradBoost	0.811	0.677	0.727	0.870
XGBoost	0.869	0.792	**0.826**	**0.907**

Table 5.8: Runway assignment scenario C: comparison to Churchill *et al.* [CCJ21]

Feature Importance

The GradBoost model, chosen for its superior performance in both scenarios A and B, was used to determine the most important features for the runway assignment classification. The result is presented in figure 5.12. At the top of the most important features are

headwind components of the runways, which confirms the findings of Ramanujam *et al.* [RB15].

Next is the direction of the destination (fwd_azimuth_vie_positive). This shows an emphasis of airport operators, to have flights takeoff into the direction of their destination, which improves efficiency and reduces fuel consumption. The current demand (nr_of_flights_total_interval) is shown to have an influence, indicating that the decision making process changes in times of high demand. Features related to noise abatement measures, such as temporal information (hour_bucket_07:00-21:00) and runway usage (RW29_month_in) are shown to be important as well. The embedding features (IATAFLIGHTDESIGNATOR_n, IATAAIRLINECODE_n) are among the important features too.

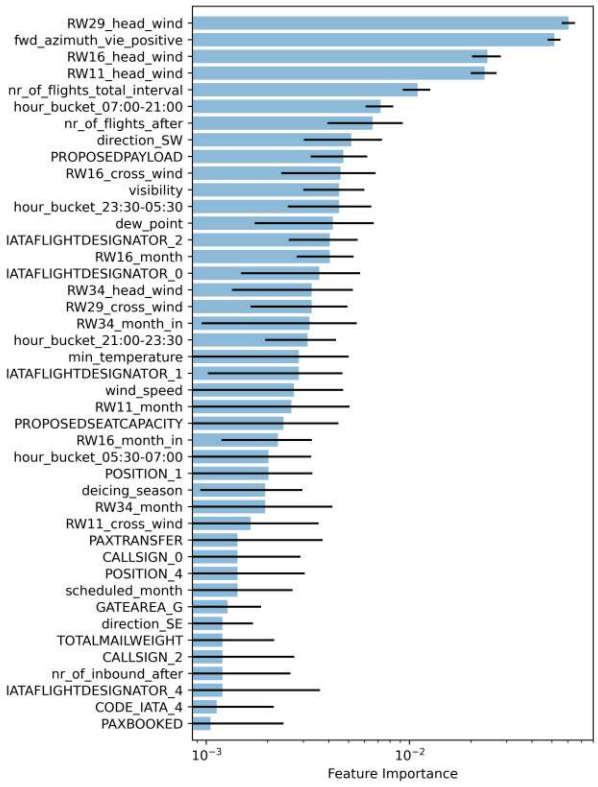

Figure 5.12: Feature permutation importance of runway assignment prediction

5.4 Deicing

The deicing services are used in less than 3% of all flights. Despite this low prevalence, its pronounced impact of doubling the average taxi time motivates the investigation into how well ML can predict an aircraft's deicing usage.

Evaluation

The evaluation of the deicing usage is done in two scenarios. Similar to the previous two tasks, scenario A is a prediction over a 30 hour forecast horizon of the TAF and scenario B is a prediction at the moment of block-off. Deicing usage prediction is a binary classification task. Analogous to the runway assignment prediction, we decided to use a logistic regression as a baseline model. Contrary to the previous prediction tasks, there is no literature to compare the results to. To the best of our knowledge, this is a novel application of ML.

Scenario A

The results of the precision score over the forecast horizon are displayed in figure 5.13. The baseline is very unstable, which impedes an assessment of the scores. The best performing models are RF with 0.6 for forecast horizons up to 16 hours and Extra-Trees with 0.48 - 0.58 for forecast horizons above.

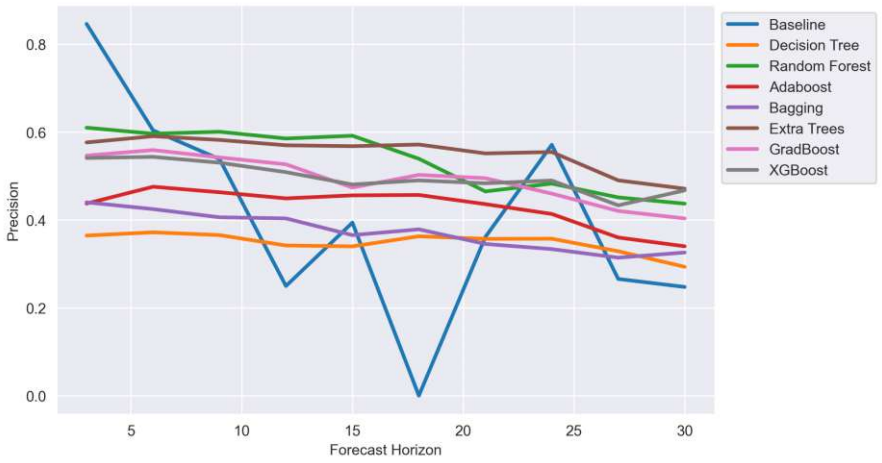

Figure 5.13: Deicing usage precision over forecast horizon

All our models score above baseline in recall, as displayed in figure 5.14, with the DT achieving the highest score across the majority of the forecast horizon. The performance, using the F1-Score, is displayed in figure 5.15. For forecast horizons up to 17 hours, the AdaBoost model achieves the highest score, while the DT shows the best performance at longer forecast horizons.

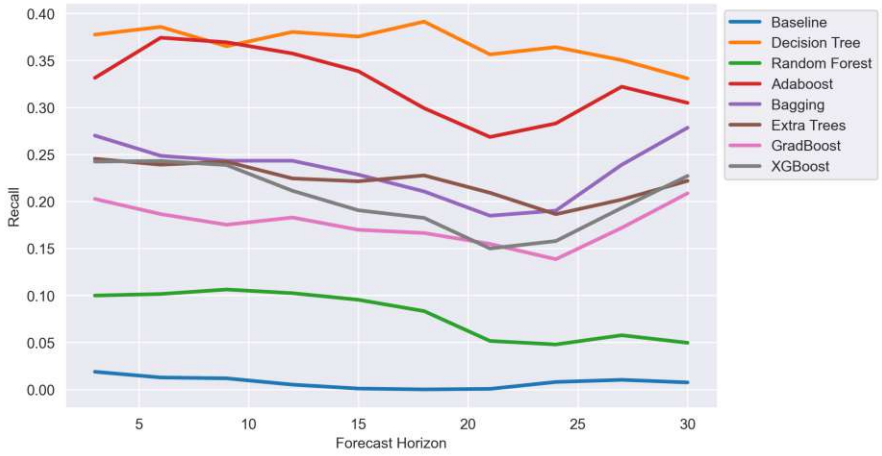

Figure 5.14: Deicing usage recall over forecast horizon

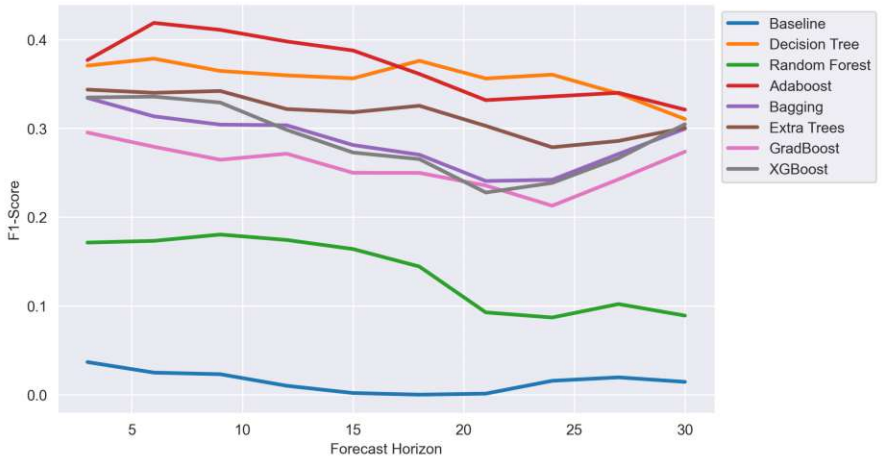

Figure 5.15: Deicing usage F1-Score over forecast horizon

Scenario B

This scenario emulates a prediction at the moment of block-off and assumes knowledge of information that is available at this moment, such as the observed weather. The scores of our models are displayed in table 5.9. As in scenario A, there is no classifier that clearly outperforms on all metrics. The baseline classifier shows a strong bias towards precision, at the cost of recall and therefore F1-Score. None of our model achieves a higher precision score. Conversely, all our models perform above baseline on recall, with the AdaBoost achieving the highest score. The results for F1-Score are similar. The Extra-Trees model scores highest in ACC. Overall, the differences in ACC scores among all models are small, cause by the strong imbalance of the dataset.

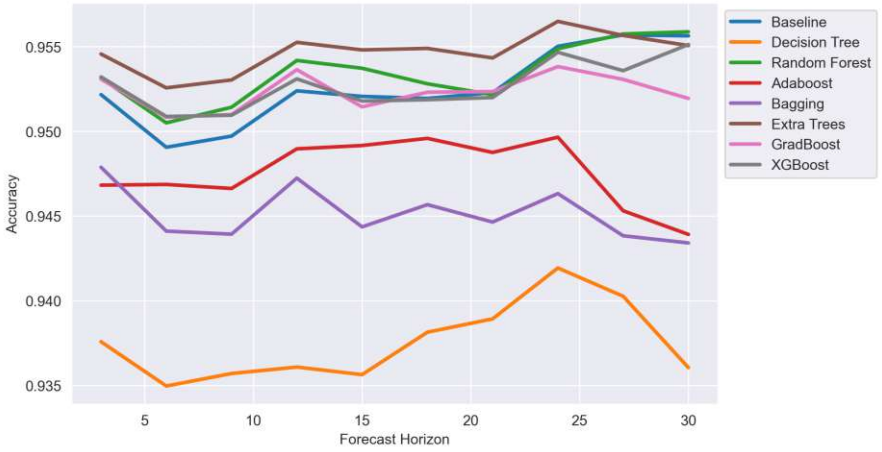

Figure 5.16: Deicing usage ACC over forecast horizon

	Precision	Recall	F1-Score	ACC
Baseline	**0.782**	0.052	0.097	0.956
DT	0.381	0.345	0.362	0.942
RF	0.662	0.153	0.249	0.956
AdaBoost	0.498	**0.391**	**0.438**	0.953
Bagging	0.628	0.296	0.403	0.959
Extra-Trees	0.656	0.295	0.407	**0.960**
GradBoost	0.615	0.220	0.325	0.957
XGBoost	0.632	0.271	0.379	0.958

Table 5.9: Deicing usage scenario B: prediction at block-off time

Feature Importance

The result of the feature permutation importance is shown in figure 5.17. The most important features are weather-related. They include the weather phenomena, temperature, visibility and dew point. Notably, an embedding feature related to the IATA code is among the important features.

5.5 Discussion

The results from the previous sections allow for addressing the research questions of this thesis.

How much can the use of ML models improve the prediction of taxi time, runway assignment, and deicing usage above baseline?

In the scenario of taxi time prediction in advance, with a forecast horizon of up to 30 hours, an Extra-Trees regressor predicts 90% of taxi times within 5 *min*, compared to an

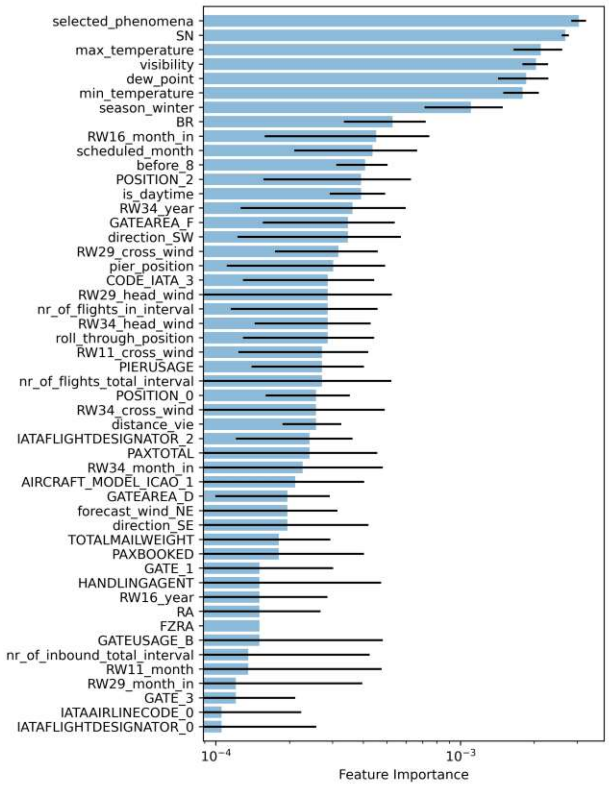

Figure 5.17: Feature permutation importance of deicing prediction

OLS baseline of 87%. This reduces the error rate by more than 20%. The same model achieves a RMSE of 3.7 *min*, compared to a baseline of between 4 and 4.1 *min*, which lowers it by up to 10%. When excluding certain types of flights, which are difficult to predict, the model performance in ACC±5*min* increases to 94% compared to a baseline of 93%. The results of our models are comparable with the literature and achieve good results in a direct comparison on similar datasets of different airports. It should be noted, that airports differ significantly, which has to be considered when comparing different model performances.

For runway assignment prediction up to 30 hours in advance, our RF, Extra-Trees, GradBoost, and XGBoost models achieve the highest F1-Scores with 0.66 to 0.62 compared to a baseline of 0.55 to 0.51. The Extra-Trees and RF models achieve a higher precision and ACC score respectively. In the scenario of a prediction at the time of block-off, GradBoost achieves the highest in F1-Score with 0.707 (baseline 0.573) and Recall 0.677

(baseline 0.535). When comparing our results to the literature, our models achieved a higher precision and ACC, but not a higher recall.

In the prediction task of deicing usage, our models performed significantly above baseline in F1-Score. AdaBoost and DT achieved the highest scores, with 0.31 to 0.42 across the entire forecast horizon, compared to a baseline of below 0.05. In a prediction at the moment of block-off, AdaBoost scored 0.44 compared to a baseline of 0.1.

Which features are most relevant for these predictions?

The most important features for taxi time prediction are an aircrafts current position, different weather phenomena, the aircraft size, temperature, direction of the flight destination, and the current demand on different runways.

For the task of runway assignment prediction, the most important features are the headwind components relative to the runways, the direction of the flight destination, the current demand, and whether the flight departs at nighttime, where the noise abatement measures limit the use of runways.

Which algorithm is most appropriate for this task, and what are the optimal hyperparameters?

Overall, the best performing model for taxi times prediction in different scenarios is an Extra-Trees Regressor, followed by a GradBoost and a RF. We found the best results to be achieved with a hyperparameter configuration of 145 estimators, with a maximum depth 59, reduction in Poisson deviance as splitting criteria, a maximum of 51 features considered at each split, a minimum of 9 samples for a split and a minimum of 4 samples at each leaf. As a preprocessing, a Quantile Transformer has shown the best results.

Our best performing model on runway assignment prediction is a GradBoost Classifier, followed by an XGBoost, Extra-Trees, and RF. The best hyperparameter configuration was found to be 481 estimators, with a learning rate of 0.053, maximum depth 23, squared error as splitting criteria, a maximum of 55 features considered at each split, a minimum of 8 samples for a split and a minimum of 8 samples at each leaf. As a preprocessing, a Standard Scaler has shown the best results.

For deicing usage prediction, the best results were achieved with an AdaBoost classifier, followed by a DT and an Extra-Trees classifier. The best hyperparameters were 375 estimators with a learning rate of 1.24 and a Quantile Transformer as preprocessing.

While we tested large numbers of hyperparameter combinations and evaluated the models using the best combination we found, it should be noted that most hyperparameters show a large range of good performances.

CHAPTER 6

Conclusion

In this thesis, we investigated the use of ML to predict a flight's taxi time, runway assignment, and deicing usage at Vienna Airport. While taxi time prediction and runway assignment prediction have been studied at different airports before, this was a new investigation at Vienna Airport. The prediction of deicing usage, to the best of our knowledge, has never been studied before.

After conducting a comprehensive literature review and corresponding with domain experts, we identified suitable input features for ML models regarding these tasks. We gathered datasets from different sources, including proprietary datasets from Vienna airport, as well as publicly available datasets, such as historic weather reports. A new dataset was created by applying different preprocessing and feature engineering methods. This included the training of embeddings for a vector representation of high cardinality categorical features. This dataset was analysed using statistical methods, as well as visualizations.

We identified a selection of ML algorithms and optimized their hyperparameters for the given tasks. To evaluate the models, we created different scenarios, ranging from emulating a real-world deployment to recreating the dataset sizes of selected papers for a comparison.

Our models for taxi time prediction performed better than the baseline and had a mean absolute error of up to 2.5min for a forecast horizon of up to 30 hours. We identified the aircraft position, weather-related features, the aircraft size, the current demand on the airport and information about the flight destination as important information for such a prediction. In a direct comparison with the literature, our models achieved good results and outperformed some of the models in the available literature.

Our best models predicted a flight's runway assignment with an ACC between 79%, for a prediction 30 hours in advance, and 0.85%, for a prediction right at block-off. We identified the headwind components on the runways, the direction of the flight, the

current demand, features related to noise abatement measures, and visibility as the most important components In a direct comparison with results from the literature, our models performed better on ACC and precision.

In the novel task of deicing usage prediction, our models achieved an F1-Score of 0.31 for predictions of up to 30 hours in advance and 0.44 for predictions at block-off. For prediction ACC, the models scored between 95% and 96% for the entire forecast horizon. The most important features for this prediction were found to be related to weather. This includes weather phenomena, temperature, visibility, and dew point.

Overall, the results show that ML models are useful for the prediction tasks of this thesis. While the decisions of human domain experts may not be modeled completely, the models we presented could be used as an assistance, which allows experts to focus their time exclusively on the cases which are harder to predict.

Our dataset spanned over two years, where the distribution of our prediction targets differed across the years. In a future study, more years could be incorporated to include a wider range of data into the development of the models. Furthermore, the study could be replicated on different airports. The literature has shown varying results among different airports, which raises the necessity to evaluate the findings on each airport individually.

List of Figures

List of Tables

Bibliography

[AAB18] Md Shohel Ahmed, Sameer Alam, and Michael Barlow. A multi-layer artificial neural network approach for runway configuration prediction. In *8th International Conference on Research in Air Transportation (ICRAT 2018)*, Castelldefels, ES, 2018.

[AB15] Jacob Avery and Hamsa Balakrishnan. Predicting airport runway configuration: A discrete-choice modeling approach. In *11th USA/Europe Air Traffic Management Research and Development Seminar*, Lisbon, PT, June 2015.

[AB16] Jacob Avery and Hamsa Balakrishnan. Data-driven modeling and prediction of the process for selecting runway configurations. *Transportation Research Record*, 2600(1):1–11, 2016.

[BBL+23] Bernd Bischl, Martin Binder, Michel Lang, Tobias Pielok, Jakob Richter, Stefan Coors, Janek Thomas, Theresa Ullmann, Marc Becker, Anne-Laure Boulesteix, Difan Deng, and Marius Lindauer. Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges. *WIREs Data. Mining. Knowl. Discov.*, 13(2), 2023.

[BDMS20] Kathrin Blagec, Georg Dorffner, Milad Moradi, and Matthias Samwald. A critical analysis of metrics used for measuring progress in artificial intelligence. *CoRR*, abs/2008.02577, 2020.

[BDVJ03] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3:1137–1155, 2003.

[Bel66] Richard Bellman. Dynamic programming. *Science*, 153(3731):34–37, 1966.

[BGS09] Poornima Balakrishna, Rajesh Ganesan, and Lance Sherry. Application of reinforcement learning algorithms for predicting taxi-out times. In *the 8th USA/Europe Air Traffic Management (ATM) Research and Development Seminars*, Napa, CA, USA, 2009.

[BGS10] Poornima Balakrishna, Rajesh Ganesan, and Lance Sherry. Accuracy of reinforcement learning algorithms for predicting aircraft taxi-out times:

A case-study of tampa bay departures. *Transportation Research Part C: Emerging Technologies*, 18(6):950–962, 2010.

[BGSL08] Poornima Balakrishna, Rajesh Ganesan, Lance Sherry, and Benjamin S. Levy. Estimating taxi-out times with a reinforcement learning algorithm. In *2008 IEEE/AIAA 27th Digital Avionics Systems Conference*, pages 3.D.3–1–3.D.3–12, October 2008.

[BH11] Gurkaran Buxi and Mark Hansen. Generating probabilistic capacity profiles from weather forecast: A design-of-experiment approach. In *9th USA-Europe Air Traffic Management (ATM2011) Research and Development Seminar, Berlin, Germany*, January 2011.

[Bre01] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[Bre04] Leo Breiman. Bagging predictors. *Machine Learning*, 24:123–140, 2004.

[Car23] H. Carlens. State of competitive machine learning in 2022. *ML Contests*, 2023. `https://mlcontests.com/state-of-competitive-machine-learning-2022/`, Accessed 2023-03-10.

[CCJ21] Andrew M. Churchill, William Jeremy Coupe, and Yoon Chul Jung. Predicting arrival and departure runway assignments with machine learning. *AIAA AVIATION 2021 FORUM*, 2021.

[CG16] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In Balaji Krishnapuram, Mohak Shah, Alexander J. Smola, Charu C. Aggarwal, Dou Shen, and Rajeev Rastogi, editors, *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 785–794. ACM, 2016.

[CRAS11] Jun Chen, Stefan Ravizza, Jason A. D. Atkin, and Paul Stewart. On the utilisation of fuzzy rule-based systems for taxi time estimations at airports. In Alberto Caprara and Spyros C. Kontogiannis, editors, *ATMOS 2011 - 11th Workshop on Algorithmic Approaches for Transportation Modeling, Optimization, and Systems, Saarbrücken, Germany, September 8, 2011*, volume 20 of *OASIcs*, pages 134–145. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany, 2011.

[CWZ+17] Jun Chen, Michal Weiszer, Elham Zareian, Mahdi Mahfouf, and Olusayo Obajemu. Multi-objective fuzzy rule-based prediction and uncertainty quantification of aircraft taxi time. In *20th IEEE International Conference on Intelligent Transportation Systems, ITSC 2017, Yokohama, Japan, October 16-19, 2017*, pages 1–5. IEEE, 2017.

[Dia13] Tony Diana. An application of survival and frailty analysis to the study of taxi-out time: A case of new york kennedy airport. *Journal of Air Transport Management*, 26:40–43, 2013.

[Dia18] Tony 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*, 119:149–164, 2018.

[DRT+14] Rahul Dhal, Sandip Roy, Shin-Lai Tien, Christine Taylor, and Craig Wanke. An operations-structured model for strategic prediction of airport arrival rate and departure rate futures. In *14th AIAA Aviation Technology, Integration, and Operations Conference*, Atlanta, GA, USA, June 2014.

[Eur22] Eurocontrol. Eurocontrol forecast update 2022-2028, October 2022. `https://www.eurocontrol.int/publication/eurocontrol-forecast-update-2022-2028`.

[Fri01] Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):1189 – 1232, 2001.

[FS97] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.*, 55(1):119–139, 1997.

[GAAP+21] Lam Jun Guang Andy, Sameer Alam, Rajesh Piplani, Nimrod Lilith, and Imen Dhief. A decision-tree based continuous learning framework for real-time prediction of runway capacities. In *2021 Integrated Communications Navigation and Surveillance Conference (ICNS)*, pages 1–14, 2021.

[GEW06] Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Mach. Learn.*, 63(1):3–42, 2006.

[GOV22] Léo Grinsztajn, Edouard Oyallon, and Gaël Varoquaux. Why do tree-based models still outperform deep learning on typical tabular data? In *NeurIPS*, 2022.

[GTAR21] Weiwei Gu, Aditya Tandon, Yong-Yeol Ahn, and Filippo Radicchi. Principled approach to the selection of the embedding dimension of networks. *Nature Communications*, 12(1):3772, June 2021.

[HCH+19] Floris Herrema, Ricky Curran, Sander Hartjes, Mohamed Ellejmi, Steven Bancroft, and Michael Schultz. A machine learning model to predict runway exit at vienna airport. *Transportation Research Part E: Logistics and Transportation Review*, 131:329–342, November 2019.

[ICBK02] Husni Idris, John-Paul Clarke, Rani Bhuva, and Laura Kang. Queuing model for taxi-out time estimation. *Air Traffic Control Quarterly*, 10(1):1–22, 2002.

[JIR10] Richard Jordan, Mariya A. Ishutkina, and Tom G. Reynolds. A statistical learning approach to the modeling of aircraft taxi time. In *29th Digital Avionics Systems Conference, Salt Lake City, UT, USA*, pages 1.B.1–1–1.B.1–10, 2010.

[KB15] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[KJLB23] Isoon Kanjanasurat, Wasarut Jungsuwadee, Attasit Lasakul, and Chawalit Benjangkaprasert. Comparison of logistic regression and random forest algorithms for airport's runway assignment. *Journal of Physics: Conference Series*, 2497(1):012016, May 2023.

[KJPB22] Isoon Kanjanasurat, Wasarut Jungsuwadee, Boonchana Purahong, and Chawalit Benjangkaprasert. Landing runway assignment by airport traffic using machine learning. In *Proceedings of the 3rd International Conference on Industrial Control Network and System Engineering Research*, ICNSER '22, page 40–45, New York, NY, USA, 2022. Association for Computing Machinery.

[KPA22] Jean-Kevin KPADEY. Metar taf parser. `https://github.com/mivek/python-metar-taf-parser`, 2022. Accessed: 2022-10-25.

[LCJ19] Hanbong Lee, Jeremy Coupe, and Yoon Jung. Prediction of pushback times and ramp taxi times for departures at charlotte airport. In *AIAA Aviation 2019 Forum, Dallas, TX, USA*, June 2019.

[LMJ16] Hanbong Lee, Waqar Malik, and Yoon Jung. Taxi-out time prediction for departures at charlotte airport using machine learning techniques. In *the 16th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference*, Washington D.C., USA, June 2016.

[LMZ+15] Hanbong Lee, Waqar Malik, Bo Zhang, Balaji Nagarajan, and Yoon Jung. Taxi time prediction at charlotte airport using fast-time simulation and machine learning techniques. In *the 15th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference*, Dallas, TX, USA, June 2015.

[MH18] Mayara Condé Rocha Murça and R. John Hansman. Predicting and planning airport acceptance rates in metroplex systems for improved traffic flow management decision support. *Transportation Research Part C: Emerging Technologies*, 97:301–323, 2018.

[Mic01] Daniele Micci-Barreca. A preprocessing scheme for high-cardinality categorical attributes in classification and prediction problems. *SIGKDD Explor.*, 3(1):27–32, 2001.

[NMAJ17] Yoichi Nakamura, Ryota Mori, Hisae Aoyama, and Hyuntae Jung. Modeling of runway assignment strategy by human controllers using machine learning. In *2017 IEEE/AIAA 36th Digital Avionics Systems Conference (DASC)*, pages 1–7, St. Petersburg, FL, USA, 2017.

[OBUM16] Randal S. Olson, Nathan Bartley, Ryan J. Urbanowicz, and Jason H. Moore. Evaluation of a tree-based pipeline optimization tool for automating data science. In Tobias Friedrich, Frank Neumann, and Andrew M. Sutton, editors, *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference, Denver, CO, USA, July 20 - 24, 2016*, pages 485–492. ACM, 2016.

[Par22] Arash Partow. The global airport database. `http://www.partow.net/miscellaneous/airportdatabase/index.html`, 2022. Accessed 2022-10-25.

[PBB19] Philipp Probst, Anne-Laure Boulesteix, and Bernd Bischl. Tunability: Importance of hyperparameters of machine learning algorithms. *J. Mach. Learn. Res.*, 20:53:1–53:32, 2019.

[PCC11] Christopher A Provan, Lara Cook, and Jon Cunningham. A probabilistic airport capacity model for improved ground delay program planning. In *2011 IEEE/AIAA 30th Digital Avionics Systems Conference, Seattle, WA, USA*, pages 2B6–1–2B6–12, 2011.

[PPP17] Kedar Potdar, Taher Pardawala, and Chinmay Pai. A comparative study of categorical variable encoding techniques for neural network classifiers. *International Journal of Computer Applications*, 175:7–9, October 2017.

[PVG+11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[Qui86] J. Ross Quinlan. Induction of decision trees. *Mach. Learn.*, 1(1):81–106, 1986.

[RAMB13] Stefan Ravizza, Jason A. D. Atkin, Marloes H. Maathuis, and Edmund K. Burke. A combined statistical approach and ground movement model for improving taxi time estimations at airports. *J. Oper. Res. Soc.*, 64(9):1347–1360, 2013.

[RB15] Varun Ramanujam and Hamsa Balakrishnan. Data-driven modeling of the airport configuration selection process. *IEEE Trans. Hum. Mach. Syst.*, 45(4):490–499, 2015.

[RCA+14] Stefan Ravizza, Jun Chen, Jason A. D. Atkin, Paul Stewart, and Edmund K. Burke. Aircraft taxi time prediction: Comparisons and insights. *Appl. Soft Comput.*, 14:397–406, 2014.

[RKC21] Juan Rebollo, Shaymaa Khater, and William J. Coupe. A recursive multi-step machine learning approach for airport configuration prediction. In *AIAA AVIATION FORUM*, 2021.

[RMW+21] Ramakrishna Raju, Rohit Mital, Bruce Wilson, Kamala Shetty, and Michael Albert. Predicting runway configurations and arrival and departure rates at airports: Comparing the accuracy of multiple machine learning models. In *2021 IEEE/AIAA 40th Digital Avionics Systems Conference (DASC)*, pages 1–8, 2021.

[Sri11] Amal Srivastava. Improving departure taxi time predictions using asde-x surveillance data. In *2011 IEEE/AIAA 30th Digital Avionics Systems Conference*, pages 2B5–1–2B5–14, Seattle, WA, USA, 2011.

[TRT+15] Shin-Lai Alex Tien, Sandip Roy, Christine Taylor, Craig R. Wanke, and Rahul Dhal. Evaluation of an airport capacity prediction model for strategizing air traffic management. In *95th American Meteorological Society Annual Meeting*, January 2015.

[Val22] Guillermo Ballester Valor. Ogimet.com. `https://www.ogimet.com/metars.phtml.en`, 2022. Accessed: 2022-10-24.

[VTJ21] Erik Vargo, Alex Tien, and Arian Jafari. Airport taxi time prediction and alerting: A convolutional neural network approach. *CoRR*, abs/2111.09139, 2021.

[Wan11] Yao Wang. Prediction of weather impacted airport capacity using ensemble learning. In *2011 IEEE/AIAA 30th Digital Avionics Systems Conference, Seattle, WA, USA*, pages 2D6–1–2D6–11, 2011.

[Wan12] Yao Wang. Prediction of weather impacted airport capacity using ruc-2 forecast. In *2012 IEEE/AIAA 31st Digital Avionics Systems Conference (DASC), Williamsburg, VA, USA*, pages 3C3–1–3C3–12, 2012.

[WBW+21] Xinwei Wang, Alexander E.I. Brownlee, John R. Woodward, Michal Weiszer, Mahdi Mahfouf, and Jun Chen. Aircraft taxi time prediction: Feature importance and their implications. *Transportation Research Part C: Emerging Technologies*, 124(102892):1–23, 2021.

[WBXZ22] Fujun Wang, Jun Bi, Dongfan Xie, and Xiaomei Zhao. A data-driven prediction model for aircraft taxi time by considering time series about gate and real-time factors. *Transportmetrica A: Transport Science*, 0(0):1–28, 2022.

[WH00] Rüdiger Wirth and Jochen Hipp. Crisp-dm: towards a standard process modell for data mining. In *Proceedings of the Fourth International Conference on the Practical Application of Knowledge Discovery and Data Mining*, number 24959, 2000.

[Wsc23] Jeffrey Whitaker and Open source contributors. Pyproj. `https://github.com/pyproj4/pyproj`, 2006–2023. Accessed: 2022-06-02.

[WZ21] Yuan Wang and Yu Zhang. Prediction of runway configurations and airport acceptance rates for multi-airport system using gridded weather forecast. *Transportation Research Part C: Emerging Technologies*, 125:103049, 2021.

[XH22] Zheng-hong Xia and Long-yang Huang. Prediction of departure flights' taxi-out time based on intelligent algorithm optimized bp. *Mathematical Problems in Engineering*, 2022:1–12, March 2022.

[YHM+18] Jianan Yin, Yuxin Hu, Yuanyuan Ma, Yan Xu, Ke Han, and Dan Chen. Machine learning techniques for taxi-out time prediction with a macroscopic network topology. *2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC), London, UK*, pages 1–8, 2018.

[YS20] Li Yang and Abdallah Shami. On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing*, 415:295–316, 2020.