Taxi Time Prediction at Schiphol Airport

Final Article

C. Vakaet

Unchanged Image by Atamari CC BY-SA 3.0

Taxi Time Prediction with Classical and Auto Machine Learning at Schiphol Airport

Christophe Vakaet [*]
Delft University of Technology, 2629HS Delft, The Netherlands

In Partial Fulfillment of the Requirements for the Degree of Master of Science in Engineering

Under Supervison of:

Jacco Hoekstra [†] and Joost Ellerbroek [‡]
Delft University of Technology, 2629HS Delft, The Netherlands

Ferdinand Dijkstra [§]
Knowledge Development Centre, 1117ZN Schiphol, The Netherlands

Taxi time predictions are used by air traffic controllers to optimally release aircraft from the gate such that efficiency losses due to queuing are minimized, while runway capacity is maintained. More accurate taxi times can therefore result in improved airport surface operations and reduce air traffic controller workload. This article proposes a methodology to develop taxi time predictor and applies this methodology to Schiphol airport. The methodology combines novel data-driven predictors with different improvements and extensive performance evaluation. One such improvement involves using recent taxi time prediction errors to improve upcoming taxi time predictions. During evaluation, this article extends conventional analysis by analyzing different prediction horizons and performance metrics. Applying the methodology at Schiphol airport resulted in a predictor that increased the fraction of flights with a taxi time error of less than two minutes from 64.41% to 67.91% compared to the currently operational manual decision tree predictor.

Nomenclature

Astra	=	Aircraft Positioning Data from the Multilateration Ground Surveillance System
AutoML	=	Automated Machine Learning
cfs	=	Schiphol Runway Configuration Prediction Data
FS	=	Feature Selection

[*] Graduate Student, Faculty of Aerospace Engineering
[†] Professor, Faculty of Aerospace Engineering
[‡] Assistant Professor, Faculty of Aerospace Engineering
[§] Supplier, Centre of Excellence

GA	=	Genetic Algorithm
Google	=	Google AutoML Tables
h	=	Prediction Horizon
HGBR	=	HistGradientBoostingRegressor
ICAO	=	International Civil Aviation Organization
l2	=	L2 Regularization Factor
LAD	=	Least Absolute Deviation
MAE	=	Mean Absolute Error
RMSE	=	Root Mean Square Error
skv	=	Schiphol Weather Forecast Data
t	=	Computation Time
t_eval	=	The Period Under Evaluation
t_train	=	The Maximum Age of the Training Data
w	=	Coefficients or Weights of the Linear Predictor
x	=	Feature Value
z	=	Standardized Value
α	=	Scaling Factor
ϵ	=	Prediction Error
ε	=	Outlier Threshold for the HuberRegressor
λ	=	Learning Rate
μ	=	Mean
σ	=	Standard Deviation

I. Introduction

Air traffic control plan aircraft arrivals and departures such that capacity is maximized, while maintaining safety and minimizing costs. One way to minimize the economic and environmental costs of air travel is to reduce the time an aircraft spends queuing at the runway [1]. It has been estimated that a major U.S.-based airline with an extensive domestic network could reduce its fuel consumption by 1%, if taxi-delay were eliminated based on flight data from 2012 to 2013 [2]. To cancel or reduce this taxi delay, air traffic controllers can tactically hold an aircraft at a gate or parking spot rather than have the airplanes wait in line with engines on. An additional benefit of holding aircraft at the gate is an increase in passenger connection rate by giving passengers more time to catch the flight [3]. However it is important to note that gate availability can often be critical thus giving the benefit to holding aircraft at parking stands [4, 5].

Although the concept of holding might appear simple at first, it is difficult to execute efficiently within the airport operations. Care should be taken that the runway capacity is maintained, the workload for air traffic control is manageable, and no unnecessary delays at the gate are introduced. To manage these problems different ground control systems have been developed and implemented [6, 7]. Many of these systems rely on and are hindered by inaccuracies in the taxi-time predictions [8, 9]. Current predictions rely on simple and often inaccurate taxi-time estimates leading to efficiency losses and added workload for controllers. In addition, improved taxi-time predictions would improve passenger information and allow for further optimization of different ground processes.

The aim of this article is therefore to improve taxi-time estimations at airports by analyzing the practical performance of different prediction methodologies and input parameters for different prediction horizons. While different methodologies and input parameters have already been developed and compared in literature [10], this project intends to focus on different prediction horizons and adapting predictors to act on recent prediction errors to improve upcoming predictions. It is the aspiration of this project to aid decision makers to implement advanced taxi-time predictors in the ground control system.

This article has been subdivided into nine sections. Section II presents a literature review of the taxi-time prediction subject. Next, section III describes the particularities of Schiphol airport, the case-study of this article. Section IV describes the data understanding phase. Subsequently section V details the preparation process of the data. This is followed by section VI specifying the taxi time predicting. The results of the three prior sections are given in section VII, followed by a discussion of the results in section VIII. Finally section IX summarizes the main conclusions of this article.

II. Literature

For this article an extensive literature review has been performed. In the following paragraphs the conclusions from the literature review are summarized. For a more detailed description of the different papers cited below, the accompanying literature report should be consulted [11].

The first conclusion of the literature review is that extensive simulation based predictors do not seem to be able to outperform data driven methods [12–16]. Simulation based predictors are predictors where the taxi time is derived through estimating the individual track of the aircraft over the airport surface.

Next, it is found that a large number of different predictor types have been applied in literature [10, 17–29]. It is therefore believed that developing a new type of predictor is unlikely to generate large performance gains. Instead the most promising predictors in literature should be implemented and compared at the specific airport. In addition, further improvement to these predictors could be made by adding a correction to the predictor based on average prediction error of recent predictions. Such a correction has already successfully been applied in 2006 by Futer et al. but was not found in recent state-of-the-art predictors [30].

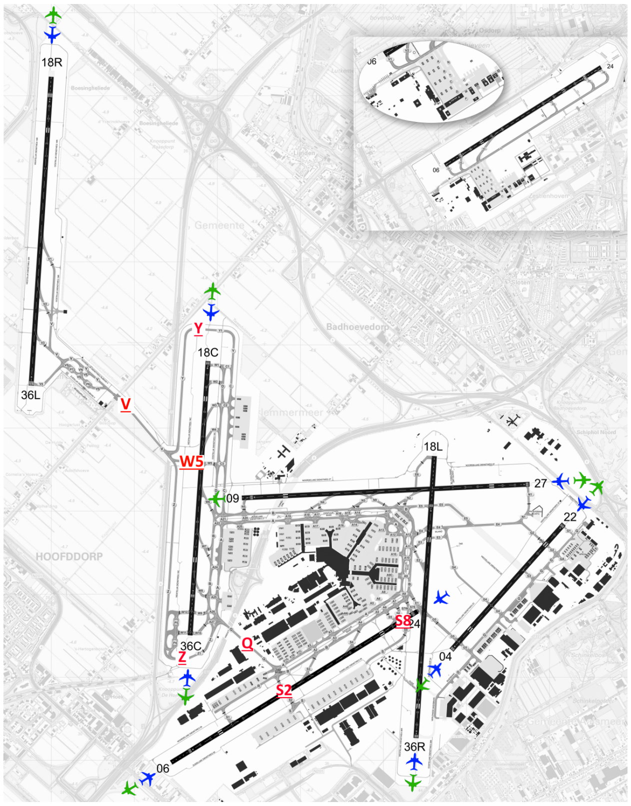

Fig. 1 Schiphol airport aerodrome chart

Thirdly it was found that significant improvements to predictor evaluations can be made by considering different prediction horizons. Additionally with respect to evaluation, it is important to identify the best metrics to capture the efficiency losses on the airport operations. It has been found that in literature a large variety of metrics are found with little analysis on the differences between them.

Finally, the literature study encounters large differences between the results and conclusions of different papers. This shows that predictor implementation, dependent variables, test-design, and the case study are major factors in determining the optimal predictor.

III. Case Study Schiphol Airport

This section highlights the main characteristics of Schiphol airport which could affect the performance of predictors. In figure 1 the six available runways at Schiphol airport can be seen. Additionally the dark and light airplane symbols near the runways indicate the nominal landing and take-off directions for each runway respectively. It is also important to note that runway 04/22 is significantly shorter and is generally only used for general aviation*.

On figure 1 it can be seen that the runways are spread out. Runway 18R/36L is particularly far removed from

*https://www.schiphol.nl/en/schiphol-as-a-neighbour/page/is-my-house-underneath-a-flight-path/ Accessed 13-01-2020 the central hub. Additionally, when runway 18C/36C is in use, aircraft taxiing towards or away from 18R/36L may need to use taxiway Y or Z rather than W5, increasing the taxi distance significantly. This indicates that the runway configuration has a significant impact on the taxi time.

Also shown on the chart are taxiways V and Q. Both taxiways are one way constraining the taxi operations significantly, especially during runway configuration changes. Taxiway Q is of particular interest since closure of the taxiway requires aircraft to make significant detours. The status of taxiway Q could therefore be an important parameter to determine the taxi time.

Lastly taxiways S2 and S8 both cross runway 06/24 and are the only exit for ramp S. It has therefore been found that the taxi time, for aircraft departing/arriving at S, is highly dependent on the ability to cross these two taxiways. Hence the use of runway 06/24 is expected to affect the taxi time for ramp S.

IV. Data Understanding

The data understanding phase consists of data collection, characterization, exploration, and quality analysis. For data collection this article uses the literature study, expert consultations (i.e. with airport air traffic controllers), and brainstorming sessions to identify the key parameters and processes which could affect taxi times at the particular airport. Subsequently a survey of the available data sources is performed to determine the best data sets to capture these parameters and processes. Finally it is important to ensure that the data is available at the required prediction horizon and has sufficient quality.

Through this process four main data sets were identified. The first data set contains logs of the data from each flight available to the tower at any time (flt). This includes the most accurate departure/arrival times, aircraft type, gate number and more. The second data set consists of aircraft positioning data from the multilateration ground surveillance system (astra). It contains the location of all vehicles with active automatic dependent surveillance–broadcast systems with one second resolution. This data set provides the best source to derive the taxi times. The third and fourth data set contain the weather (skv) and runway configuration (cfs) predictions available at the different prediction horizons. Skv is provided to the airport by the Royal Dutch Meteorological Institute. Cfs is published three times per day by the airport in collaboration with air traffic control and airliners. Each data set encompasses a two year period (2018, 2019). The features contained in each data set can be found in the appendix, table 11. More information on each data set can additionally be found in the projects midterm report [11].

V. Data Preparation

The data preparation phase consists of four elements: data cleaning, data construction, data integration, and data formatting. During initial cleaning of each data set the subset of useful entries are extracted (i.e. only departures, commercial flights etc.). This is followed by the construction phase where features identified during the data understanding phase, but not directly available in a data set, are constructed (i.e. taxi time construction from radar data). Next the different data sets are integrated to generate a new data set with all the features from the data sets linked to every individual flight analyzed. Finally the new data set is formatted for use by the predictors and evaluation tools.

For the Schiphol airport case-study initial data cleaning mainly consists of eliminating non-commercial flights, due to their low frequency but high variance. This is performed by eliminating flights with a flight number equivalent to its registration, or flights departing from a non-commercial gate using the Python programming language. In addition flights from the S ramp were ignored as taxi times show high variance and the cause (S2/S8 interaction) could not be captured in data, see also section III. In addition arrival flights were eliminated as the scope of the article is limited to departures, the same methodology can however be applied to arrivals as well. Finally, flights with a predicted outside temperature below 3° C are removed to eliminate taxi delays due to deicing. Note, flights with missing data were not eliminated as the predictor is intended to be used within the operation where missing data is present.

Next, several features were constructed using the data set. One important constructed feature, is the taxi time. The taxi time is calculated by detecting transitions of flights between different polygons (from the exit of the gate polygon to the entrance of the final runway polygon). Due to the size of the positioning data this process is performed using an SQL database, more specifically PostgreSQL. A second set of constructed features attempt to capture the traffic situation. The features quantify the number of different types of flights (departures, arrivals, non commercial) expected to take-off or land within a time range around a flights own departure from the gate. For simplicity the time range is selected based on air traffic controller knowledge. In the future further analysis could be performed to improve the set of traffic parameters. A third feature has been constructed to determine the queue time of each aircraft. This feature identifies aircraft with low speed at holding positions for the take-off runway as queuing. Additionally slow aircraft outside holding positions, but near aircraft identified as queuing are also considered queuing. This propagates the queuing property through a traffic chain outside of usual queuing areas.

With the features constructed within each data set, the different data sets can be combined. For this process the meta data from each flight is extracted from the tower data as a baseline (i.e. unique flight name, time of release from the gate, final recorded take-off runway). Subsequently the taxi time data of these flights are linked with that meta data. Next, the varying prediction horizon offsets are used to generate entries for each flight at every prediction horizon. This is followed by joining the tower flight data available for the flight at that particular time of prediction. Similarly the best weather forecast and runway configuration forecast is appended.

Before continuing to format the data, verification checks are made to ensure accurate joins between the data sets. This includes quantifying and interpreting the amount of failed joins and dropped records. In this step it is found that from the 498,145 flights that are expected in the data, using the annual Schiphol traffic reviews [31, 32], 335,962 flights or 67% are found. Most of the missing flights can be attributed to missing positioning data, with around 100,000 flights having missing records when transitioning from the gate onto the taxiways or from the taxiways onto the runway.

Another cause for excluded flights from the output data set are flights that start taxiing from outside the gate area using remote parking stands causing the taxi start detection to fail. Finally the joins are checked for consistency using manual sampling and ensuring logical connections between the parameters of the different data sets.

Finally the output data set is reformatted for use by predictors. This includes normalizing, more specifically standardizing numerical features, to accelerate predictor training. This standardization is performed with equation (1) using the feature value (x), the mean of the feature (μ), and the standard deviation (σ). Additionally missing numerical values are replaced with the mean of non missing values for predictors which are unable to handle missing values. Next, cyclical features such as minutes since midnight are discretized into bins using quantiles (other methods of encoding for these types of features could be attempted in the future). Subsequently these cyclical features are one-hot-encoded together with the other categorical features. For these features a separate category is made for missing data.

$$z = \frac{x - \mu}{\sigma} \tag{1}$$

Lastly, in table 11 each feature of the final data set is listed together with the type, source and description of the feature.

VI. Predicting

In this section the data set, constructed in the previous section, is used to make predictions. The section has been divided into three subsection: select predictor type, generate test design, developing predictors.

A. Select Predictor Type

With dozens of potential predictor types, the literature study is used to identify two predictor types best suited for this project: linear predictors, and tree-based ensemble predictors. Linear regression is chosen for its simplicity, adequate performance, and use as a reference or baseline in literature. Next the tree-based ensemble regression has been chosen for its consistent performance near the top in literature. Both of these models are implemented using the Python module scikit-learn.

While initially used as a baseline for the design of the classical machine learning predictors, automated machine learning (AutoML) showed impressive performance. Based on these findings more time has been put into developing AutoML predictors using three different AutoML tools: Google's AutoML Tables, TPOT, and AutoKeras. Google AutoML Tables provides an easy to use interface for developing machine learning predictors, it is however closed source. TPOT is an open source alternative which uses scikit-learn predictors, but does not consider neural networks. For this reason also AutoKeras is used to automatically develop neural network based predictors.

Finally these predictors are also compared to simply using the mean taxi time and the manually designed decision tree predictor currently operational at Schiphol airport.

B. Generate Test Design

With the goal of determining the best predictor, it is necessary to first identify the best metrics to compare and optimize predictors with. In literature one mostly encounters root mean square error (RMSE) or mean absolute error (MAE) to compare taxi time predictors. Additionally often the fraction of flights within a certain error bound are given. It is however unclear how the differences between these parameters relate to the operational performance.

For this reason consultations were done with Schiphol air traffic control experts who indicated that an absolute predictions error ($|\epsilon|$) of less than 2 minutes does not require manual intervention. Predictors should therefore be optimized and evaluated for the $|\epsilon| < 2$ min [%] metric. From experiments it is found that predictors which cannot directly be optimized for this metric, the MAE metric should be used over RMSE. RMSE amplifies the importance of large prediction errors which would always require manual intervention. RMSE is therefore believed to be a lesser metric for operational performance compared to MAE. In addition to these three metrics also the fraction of flights with $|\epsilon| < 5$ min and $|\epsilon| < 7$ min are used in evaluations to provide more insight on the performance of the predictor.

Next, correlations exist between the taxi time of different flights that are near in time. When data is randomly split in training, validation, and test data, the predictor can learn from the training data surrounding an evaluated prediction that the taxi time will be longer (i.e. due to construction). An operational predictor is however unable to improve the taxi time using future taxi times as those are not available. It is therefore necessary to evaluate predictors on data which takes place later in time with respect to the training data.

Furthermore, to ensure that the final evaluation provides an unbiased evaluation of the predictor, the data used in the final evaluation cannot be used during the development of the model. Instead during the development of models, intermediate evaluations should use a third data set, the validation data set. Once the final model has been developed it can subsequently be retrained on both the training and validation data and evaluated on the test data.

The data is therefore split into training (70%), validation (15%), and test data (15%) according to time. This simple time series split evaluation is however limited as it evaluates the same period each time. Therefore, to analyze the performance of a predictor over different periods of time (t_eval), cross validated time series split is used. With cross validated time series split the predictor is evaluated on multiple sections of data, using the predictor trained on only the data prior. To avoid predictors learning from outdated data, a limit can be set on the maximum age of the data (t_train). Due to the multiple cycles of training and evaluation, cross validated time series split require significantly more processing power. It has therefore only been used during the evaluation of the final predictors. A visualization of the two different evaluation methods can be seen in figure 2.

Finally from the consultations with air traffic control experts it has also been found that the most important time of prediction is 30 minutes before departure. Hence the test design in this article focuses on developing predictors for a 30

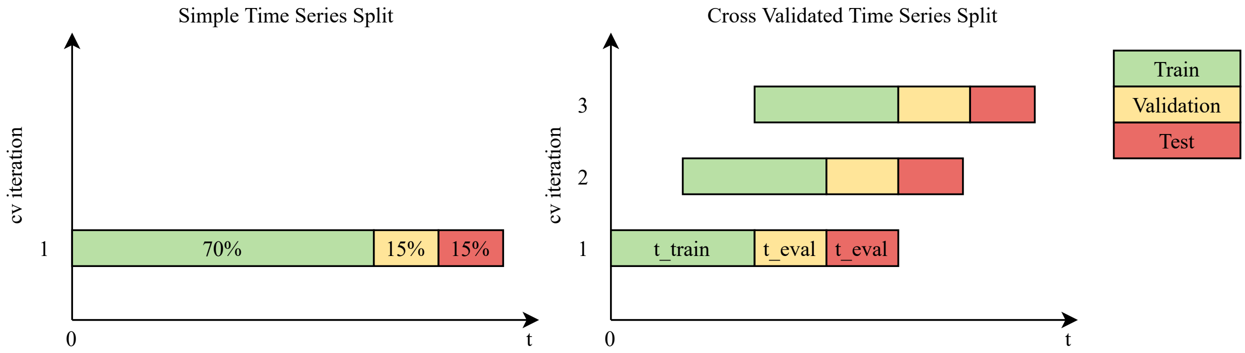

Fig. 2 Visualization of the two different evaluation methods.

minute prediction horizon (h). If the prediction horizon is not specified, the default 30 min horizon is used.

C. Develop Predictors

In total this article will be building and comparing seven predictors: the mean taxi time, a manual decision tree, a linear predictor, a tree-based ensemble predictor, and three AutoML predictors from three different AutoML tools (Google AutoML Tables, TPOT, and AutoKeras). Additionally a performance feedback correction is built and applied to the linear predictor. During the predictor development process the test data set is not used. Instead the validation data set is used for the tuning such that the test data can provide an unbiased evaluation of the final predictors.

1. Mean Taxi Time Predictor

The mean taxi time predictor simply returns the mean taxi time of the training and validation data, and returns this value as the prediction for the test data.

2. Manual Decision Tree Predictor

The manual decision tree returns the mean taxi time of training and validation data for each unique combination of departure gate number, extended take-off runway, and wake turbulence category. More information on each parameter can be found in the appendix. When a parameter is unknown the predictor falls back to the mean of the combination with the known parameters. This predictor is currently operational at Schiphol airport.

3. Linear Predictor

To build the linear predictor, first different scikit-learn linear predictor variations are compared on the whole data set. Secondly, the predictor variation with the best predictive performance but adequately low computation time is used for feature selection. Thirdly, the best linear predictor variation is evaluated with and without feature selection. Finally, this predictor in combination with the best feature selection is tuned to form the optimal linear predictor.

For the comparison of each linear variation, predictors with the default parameters from scikit-learn are used. An

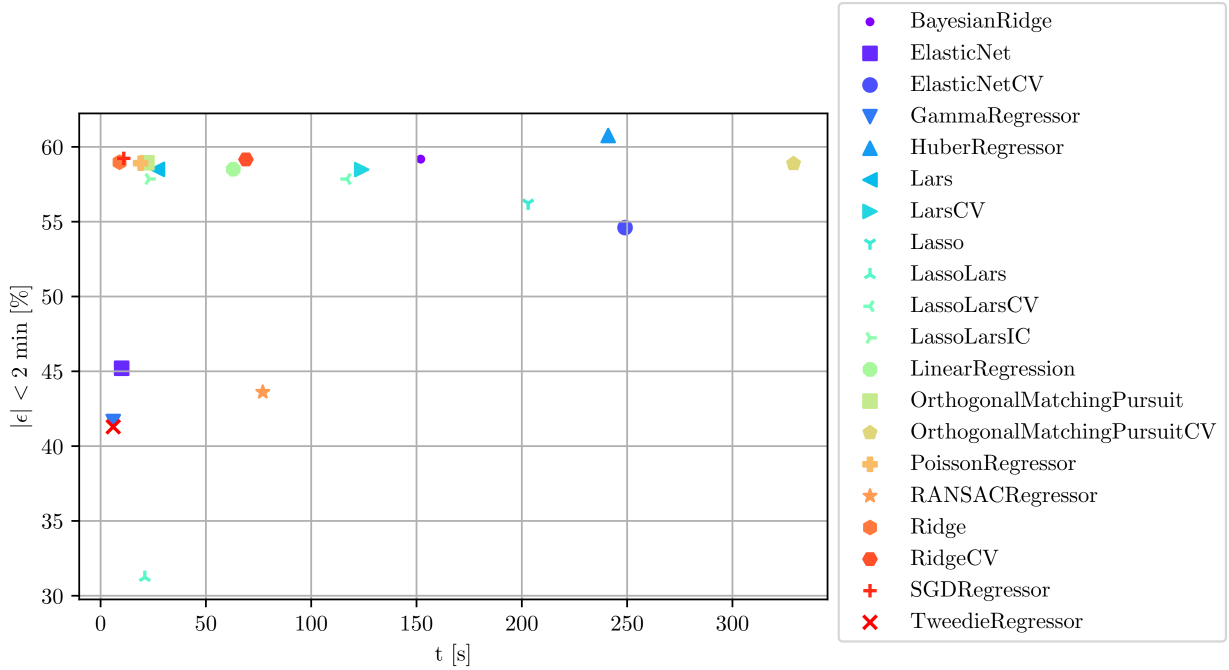

Fig. 3 Comparison of linear predictor variations $|\epsilon| < 2$ min versus t for models with $t < 1000s$.

exception is made for predictors which suggest a parameter change through a warning. In this case the change is applied and compared to the default parameters. Additionally, when different loss functions are present, each loss function is compared and the best is selected. The $|\epsilon| < 2$ min versus computation time (t) is plotted in figure 3, detailed values of the different perfromance metrics can be found in the appendix, table 9.

From the comparison two variations stand out. Firstly the Ridge linear predictor, also known as Ridge Regression or Tikhonov regularization, performs well on the <2 min metric while having a low compute time and is therefore ideal to use for feature selection (FS). Ridge regression differs from ordinary least squares linear regression by penalizing the size of coefficients ($\alpha||w||_2^2$) as seen in equation (2) where X corresponds to the feature matrix, w the coefficients or weights, y the label, and α the scaling factor.

$$\min_w ||Xw - y||_2^2 + \alpha||w||_2^2 \tag{2}$$

Next, the HuberRegression variation performs the best and has a reasonable computation time. In scikit-learn this variation of linear regression uses the loss function found in equation (3) and (4) [†]. Like the Ridge variation it also penalizes large coefficients and optimizes the square error for samples with a low normalized value [33]. However samples with a normalized value above a threshold (ε) are given a linear loss instead. This increases the robustness of the predictor to outliers, without completely ignoring these samples.

[†] https://scikit-learn.org/stable/modules/linear_model.html#huber-regression Accessed: April 15, 2021

Table 1 Performance comparison of HuberRegressor and Ridge predictor for different feature selections.

Predictor	FS	$\|\epsilon\| < 2$ min [%]	MAE [s]	RMSE [s]	$\|\epsilon\| < 5$ min [%]	$\|\epsilon\| < 7$ min [%]	t [s]
Ridge	None	58.97	126.04	181.09	93.38	97.60	7.40
	GA	60.08	125.04	181.15	93.25	97.44	1.91
	Google	59.29	126.33	182.05	93.15	97.44	1.89
Huber-	None	60.76	124.42	181.98	93.02	97.40	225.15
Regressor	GA	61.23	124.31	182.66	92.91	97.26	60.42
	Google	60.91	125.18	183.81	92.67	97.16	55.92

Table 2 Performance comparison of feature selected HuberRegressor predictor with default and tuned parameters on the validation data

α	ε	$\|\epsilon\| < 2$ min [%]	MAE [s]	RMSE [s]	$\|\epsilon\| < 5$ min [%]	$\|\epsilon\| < 7$ min [%]
0.0001	1.35	61.23	124.31	182.66	92.91	97.26
0.4	1	61.57	124.38	183.61	92.67	97.17

$$\min_{w,\sigma} \sum_{i=1}^{n} \left(\sigma + H_{\varepsilon}\left(\frac{X_i w - y_i}{\sigma}\right)\sigma\right) + \alpha ||w||_2{}^2 \tag{3}$$

$$H_{\varepsilon}(z) = \begin{cases} z^2, & \text{if } |z| < \varepsilon, \\ 2\varepsilon|z| - \varepsilon^2, & \text{otherwise} \end{cases} \tag{4}$$

Using the Ridge predictor and a genetic algorithm (GA) an optimal subset of the features is selected. Each subset of features is evaluated based on the $|\epsilon| < 2$ min metric with a penalty of -0.1% for each included column. The penalty ensures that non contributing columns are ignored. An additional subset of features is derived based on the top 10 columns according to the Google AutoML Tables predictor discussed below. The features selected by GA and the relative importance of different features according to Google AutoML Tables is visualized in figure 4. The type, source, and description of each feature can be found in the appendix, table 11. Finally, the performance of no feature selection and the two different feature selections are compared for the Ridge and HuberRegression predictors in table 1. From the comparison it can be found that the GA feature selection improves performance the most while significantly reducing computation time.

Finally, the two parameters (α, ϵ) of the GA feature selected HuberRegressor predictor are tuned. The tuning is performed using a simple grid search. From the grid search an optimum is found for α equal to 0.40 and ε equal to 1. This results in an increased penalty to large coefficients, and increases the robustness to outliers. In table 2 the performance metrics of the default and tuned predictor are listed. The tuned HuberRegressor with GA feature selection is the final linear regression predictor that will be used in subsequent comparisons with different predictor types.

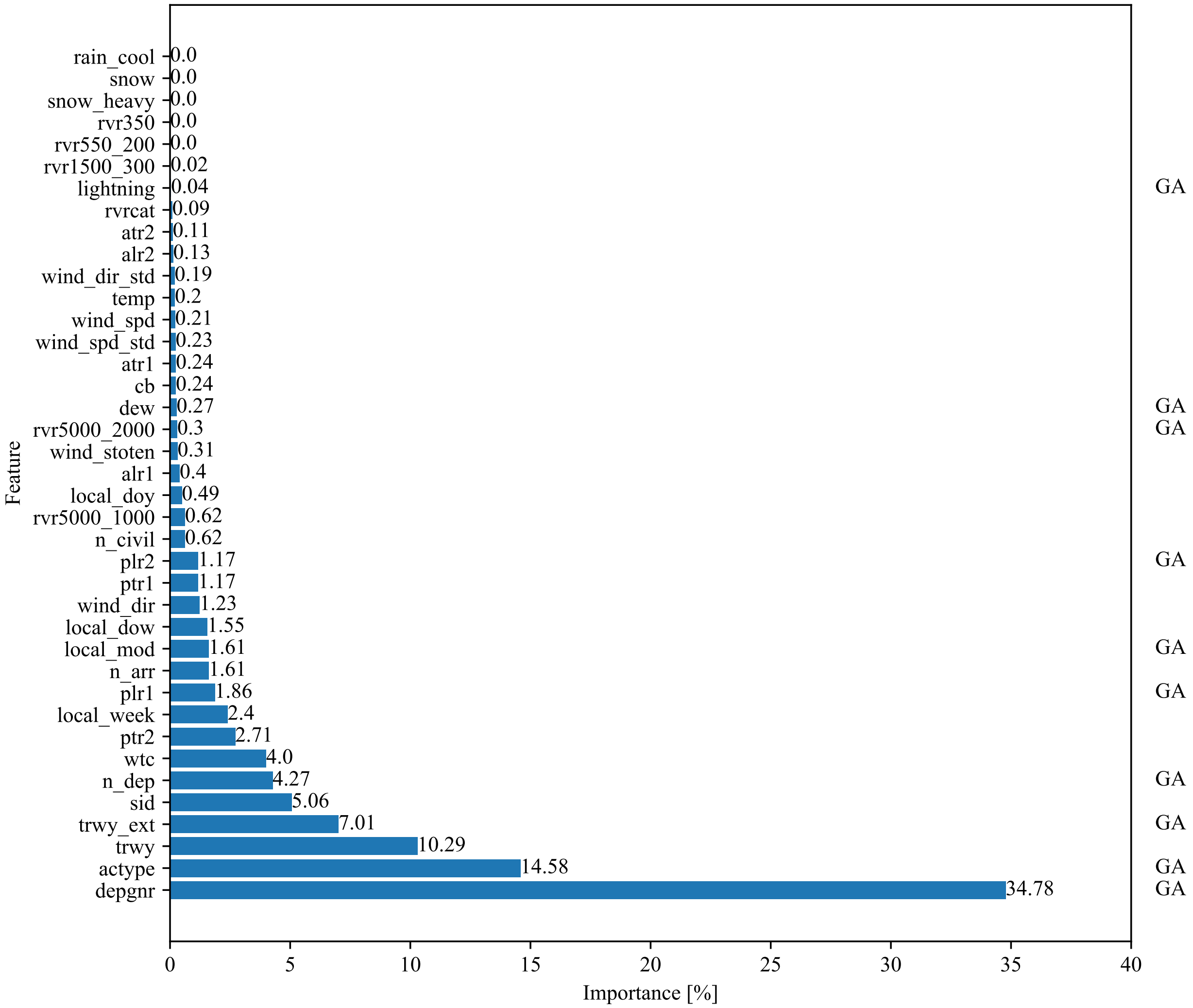

Fig. 4 The features selected by GA and the relative importance of different features according to Google AutoML Tables (Google).

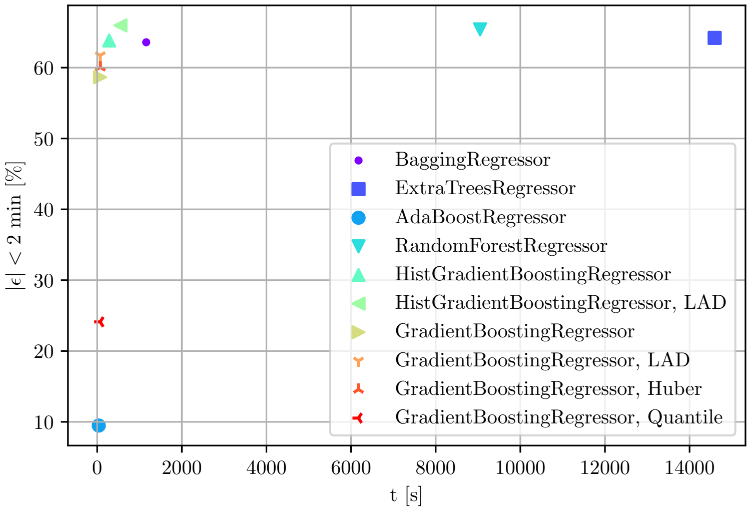

Fig. 5 Comparison of tree-based ensemble predictor variations $|\epsilon| < 2$ versus t.

4. Tree-based Ensemble Predictor

The building process for the tree-based ensemble predictors is similar to the process for linear regression. Firstly the different tree-based ensemble predictor variations are compared, using the same methodology as for the linear predictors, see previous section. In figure 5 the $|\epsilon| < 2$ min versus t is plotted and in the appendix, table 10, detailed performance metrics for each variation can be found.

From the figure it can be seen that the HistGradientBoostingRegressor (HGBR) predictor variation, using a least absolute deviation (LAD) loss, performs the best. This predictor variation is an improved version of the more traditional Gradient Boosted Decision Trees [34]. The principle behind this predictor variation is to iteratively add decision trees to estimate the error of the previous set of trees. The new tree is therefore optimized for the weaknesses of the previous trees. The implementation of HGBR (version 0.24.1) used in this article is still experimental and subject to changes in future versions [‡].

Additionally it can be seen that tree-based ensemble predictor variations require significantly more computing time compared to linear regression. Due to limited computational resources, GA feature selection could therefore not be performed using a tree-based ensemble method. Hence, the same results of the linear regression GA feature selection is used instead. The results from this comparison can be found in table 3. From the table it is concluded that feature selection reduces computation times significantly, but performance is also reduced. Since computation time is of little concern in the operations, no feature selection will be performed on the final tree-based ensemble predictor.

Finally to tune the HGBR predictor grid search is performed on three parameters: l2 regularization (l2) factor, learning rate (λ), and max leaf nodes. The l2 regularization term is similar to the α term in equation 3 and helps to

[‡] `https://scikit-learn.org/stable/modules/ensemble.html#histogram-based-gradient-boosting` Accessed: April 20, 2021

Table 3 Performance comparison of the HistGradientBoostingRegressor LAD loss predictor with different feature selections on the validation data

FS	$\|\epsilon\| < 2$ min [%]	MAE [s]	RMSE [s]	$\|\epsilon\| < 5$ min [%]	$\|\epsilon\| < 7$ min [%]	t [s]
None	65.96	116.03	176.77	93.17	97.36	107.48
GA	65.70	116.35	177.14	93.32	97.37	67.37
Google	65.74	117.29	178.56	92.96	97.26	20.65

Table 4 Performance comparison of HistGradientBoostingRegressor with default and tuned parameters on the validation data

l2	λ	leaf$_{max}$	$\|\epsilon\| < 2$ min [%]	MAE [s]	RMSE [s]	$\|\epsilon\| < 5$ min [%]	$\|\epsilon\| < 7$ min [%]	t [s]
0	0.1	31	65.96	116.03	176.77	93.17	97.36	113.66
0.375	0.1	700	68.05	111.14	171.50	93.91	97.63	885.97

smooth the final results to avoid over-fitting [35]. Next the learning rate determines the step length of the gradient descent procedure used for each additional tree. The max leaf nodes determines the maximum amount of leaves that each decision tree can have. In addition to tuning these parameters also the number of trees has been increased until the performance gains became insignificant. While both increasing the max leaf nodes and the number of trees dramatically improved performance, it has come at significant increase in training time. In table 4 the performance improvements from default to tuned and corresponding training times can be found.

5. AutoML

Lastly the AutoML predictors are built on the same data set, but reformatted to fit each tool. There are significant differences between the three tools. Firstly, Google AutoML Tables is a closed source tool generating an ensemble of the 25 best predictors found during the training period[§]. Next TPOT is an open source tool that builds the best scikit-learn pipeline it can find [36]. Finally, AutoKeras is another open source tool that focuses on building the best neural network using tensorflow and keras [37].

D. Performance Feedback

During consultations with air traffic controllers about the current taxi time predictor (the manual decision tree described above) one issue is raised consistently. Namely, during certain periods of anomalous operations the predictor is incapable of dynamically adjusting its predictions. Manual adjustments are therefore needed on each flight increasing the workload significantly. To solve this problem a method for performance feedback has been developed based on the research by Futur et al [30].

The method adds a fraction of the recent average error of the predictor to the output of the predictor. Therefore, when the predictor is consistently predicting shorter or longer taxi times over a certain period, the predictions are corrected.

[§] `https://cloud.google.com/automl-tables` Accessed: April 15, 2021

Table 5 Performance comparison of the linear predictor with and without performance feedback.

Performance Feedback	$\|\epsilon\| < 2$ min [%]	MAE [s]	RMSE [s]	$\|\epsilon\| < 5$ min [%]	$\|\epsilon\| < 7$ min [%]
None	61.57	124.39	183.62	92.66	97.17
1 hour & 7 days	61.69	124.86	184.78	92.42	97.05

Table 6 Performance comparison of the final predictors on the test data at 30 min prediction horizon.

Predictor	$\|\epsilon\| < 2$ min [%]	MAE [s]	RMSE [s]	$\|\epsilon\| < 5$ min [%]	$\|\epsilon\| < 7$ min [%]
Mean Taxi Time	34.55	205.84	262.16	77.39	91.58
Manual Decision Tree	64.41	115.46	161.77	94.33	97.84
Linear Predictor	60.91	123.55	172.43	92.78	97.45
Tree-based Ensemble Predictor	66.84	110.78	158.52	94.46	98.00
Google AutoML Tables	67.91	107.62	152.89	95.28	98.29
TPOT	61.01	121.33	166.52	93.90	97.82
AutoKeras	46.33	157.29	203.75	88.22	96.16

This correction has two parameters that need to be tuned: the period over which the average error is taken, and the fraction of the average error that is added.

Due to limited computational resources performance feedback has only been applied to the linear predictor. Using the linear predictor, simple time series split evaluation, and a grid search on the validation data set revealed two optimal periods: the average error of the past hour, and the average error of the past seven days. This indicates that there could be benefits to adding both a short and long term correction. The performance of the developed performance feedback predictor on the validation data can be found in table 5.

VII. Results

This section presents the performance of the predictors on the test data, using both the original training and the validation data set as training data. By default the simple time series split evaluation technique is used except for subsection D.

A. Predictor Comparison

In this subsection the results of the comparison between the seven final predictors are presented. In table 6 the performance metrics of the final predictors on the test data at 30 min prediction horizon are compared. Figure 6 visualizes the $|\epsilon| < 2$ min of the final predictors, trained on the 30 min prediction horizon data, but evaluated on the test data at different prediction horizons.

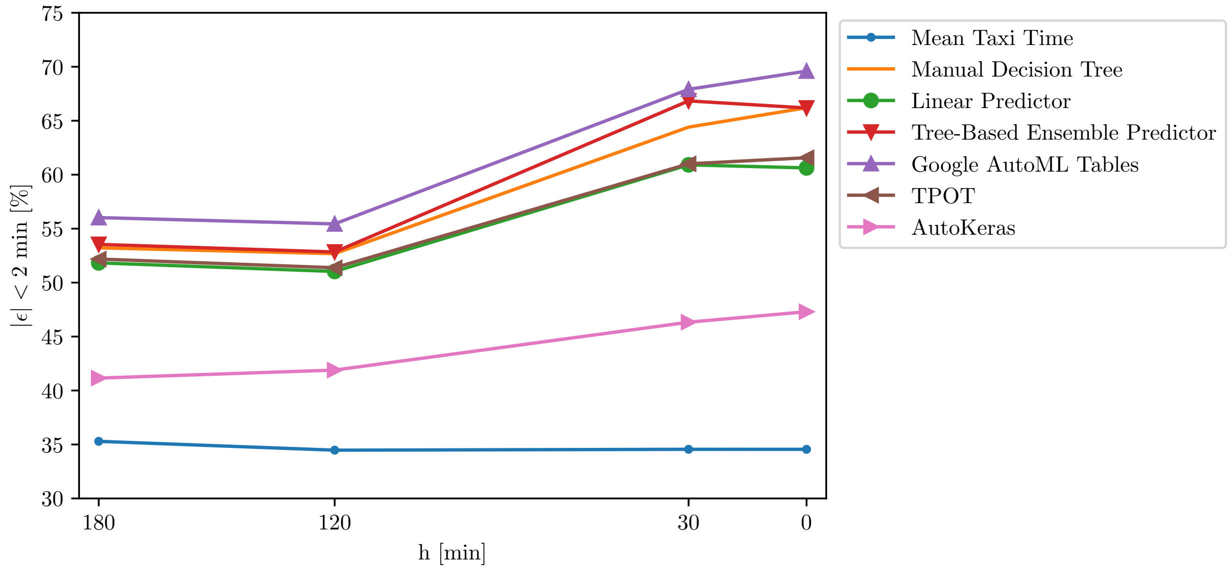

Fig. 6 **$|\epsilon| < 2$ min of the final predictors, trained on the 30 min prediction horizon data, but evaluated on the test data at different prediction horizons.**

Table 7 **Performance comparison of the linear predictor on the test data with and without performance feedback at 30 min prediction horizon.**

Performance Feedback	$\lvert\epsilon\rvert < 2$ min [%]	MAE [s]	RMSE [s]	$\lvert\epsilon\rvert < 5$ min [%]	$\lvert\epsilon\rvert < 7$ min [%]
None	60.91	123.55	172.43	92.78	97.45
1 hour & 7 days	60.93	123.99	173.56	92.60	97.34

B. Performance Feedback

In table 7 the results of applying the linear predictor performance feedback correction to the test data at 30 min prediction horizon can be found.

C. Training Set

An initial hypothesis was made that training the predictors on the most accurate data rather than on the data that is available at the time of prediction would improve performance at the time of prediction. To test this hypothesis separate linear predictors are fitted on the data set at each prediction horizon. Subsequently the performance of the predictors are evaluated at the different prediction horizons. The results of this comparison can be found in figure 7.

D. Cross-Validated Performance

In figure 8 the $|\epsilon| < 2$ min of the linear predictor over the entire year 2019 is shown, using the cross-validation time series split with t_eval equal to 1 month and an infinite t_train. Next in table 8 the average value of the different performance metrics over the twelve months for the different t_train are given.

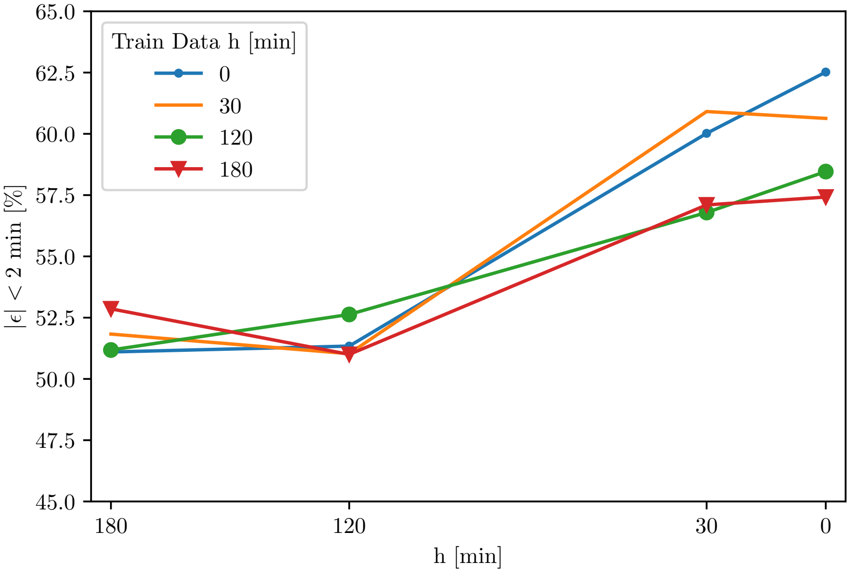

Fig. 7 Performance of Linear Regression for different test and train data set prediction horizons.

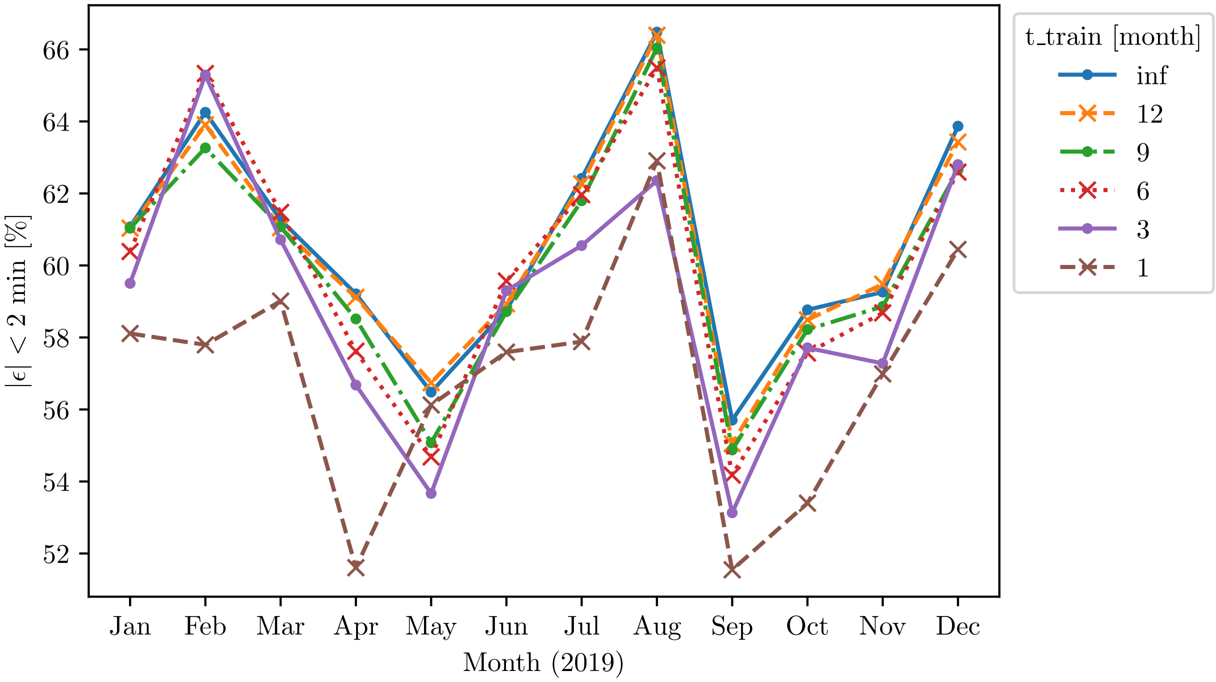

Fig. 8 $|\epsilon| < 2$ min of the linear predictor over the year 2019 for different t_train, trained and tested at 30 min prediction horizon.

Table 8 Performance metrics of the linear predictor over the year 2019 for different t_train, trained and tested at 30 min prediction horizon.

t_train [month]	$\|\epsilon\| < 2$ min [%]	MAE [s]	RMSE [s]	$\|\epsilon\| < 5$ min [%]	$\|\epsilon\| < 7$ min [%]
inf	60.51	176.16	124.45	92.96	97.37
12	60.38	175.98	124.49	93.01	97.38
9	59.89	176.83	125.40	92.94	97.38
6	59.80	177.21	125.78	92.92	97.38
3	58.84	179.22	127.76	92.65	97.33
1	56.86	186.42	133.61	91.62	96.93

VIII. Discussion

This section discusses the results of the predictor comparison, performance feedback, training set, and cross-validated performance evaluations. The results and discussion intentionally do not contain any computation times, because the relative computation times of predictors varied significantly between different hardware and software.

A. Predictor Comparison

Firstly it is logical to find the mean taxi time predictor performing the worst. Any predictor performing worst would be an indication of incorrect implementation.

The second worst performer, AutoKeras, also performs well below the other predictors. This is believed to be due to the low number of iterations (100) with low number of epochs (20) that were run during the training. The performance could therefore be improved by using more computational resources. However the current results are already derived using one computer for nine hours with an eight core AMD Ryzen 7 1700X, GPU acceleration (NVIDIA GP107), and 32 GB of RAM.

Next up, TPOT performs nearly identically to the linear predictor. For TPOT computational resources are again a significant limiting factor. For the TPOT predictor only 10 generations with a population of 10 could be trialed over a six hour period using the same computer as mentioned in the previous paragraph.

To generate the results of both AutoKeras and TPOT, significant tuning is required to use the computational resources most effectively. Furthermore different preprocessing steps are required for each tool to work. These steps are contrary to the main benefit expected from AutoML, lowering the development time.

While linear predictors are simple to implement with low training time and complexity, also the performance is rather low. It performs worse on every metric compared to the manual decision tree predictor currently operational at Schiphol airport.

From all the predictors, except for the mean taxi time predictor, the manual decision tree is the least complex due to the use of just one data set (tower data). Yet, it outperforms the TPOT, AutoKeras, and Linear predictors. It also matches performance with the tree-based ensemble predictor for the prediction horizons that have not directly been trained for.

Finally the tree-based ensemble predictor performs very well with a $|\epsilon| < 2$ min score of 66.87% for the 30 min prediction horizon. The best predictor is however the Google AutoML Tables predictor with a $|\epsilon| < 2$ min score of 67.91% at that prediction horizon. It is believed that this performance is achieved by harnessing the power of the cloud allowing for the development of many predictors with different parameters. The Google AutoML Tables predictor uses an ensemble of 25 predictors including neural networks and Gradient Boosting Decision Trees, which is an ensemble of tree-based predictors in itself.

Comparing these results to literature is difficult as input data and airports differ significantly. The results do at least fall within the range of results in literature. For example, Ravizza et al. achieved 85.30%, and 86.12% $|\epsilon| < 2$ min at the Stockholm Arlanda Airport and Zurich Airport using a linear predictor [26]. However both airports are significantly smaller than Schiphol. Additionally predictions are made on historical data potentially not available at the time of prediction. In contrast, the linear predictors developed by Lee et al. at Charlotte Airport and Lian et al. at Beijing International Airport appear to perform worse than the mean taxi time predictor at Schiphol airport [27, 29].

B. Performance Feedback

In table 7 an insignificant gain in $|\epsilon| < 2$ is found at a small cost to the other performance metrics by using the performance feedback correction. These results do however not show how performance feedback performs during anomalous operations, as performance feedback is developed to lower the average error during such operations. Better evaluation techniques are therefore required to analyze the true impact of performance feedback. Additionally the performance feedback correction could be improved by using a secondary predictor with the recent average error as an extra feature, rather than adding a fraction of this value to the original prediction. The results are therefore inconclusive on the value of performance feedback.

C. Training Set

Figure 7 shows that the performance of the linear predictor at a certain prediction horizon is highest when the predictor has been trained on the training data at that prediction horizon. The hypothesis that training a predictor on the most accurate data rather than on the available data at the time of prediction would improve performance at the time of prediction is therefore false. Instead training on the less accurate, but realistic, data allow the predictor to determine which features to rely on for its prediction at that prediction horizon.

D. Cross-Validated Performance

The results of the cross-validated performance of the linear predictor, found in figure 8, provide multiple insights into predictor and the data. Firstly it shows that $|\epsilon| < 2$ min increases as the t_train increases, with 'infinite' t_train performing the best. It should however be noted that 'infinite' t_train signifies that all training data available is used, but the training data only starts in 2018. It is therefore possible that a maximum t_train exist after which performance degrades due to the training data becoming more and more outdated. Additionally it follows that the results of infinite t_train and 12 months t_train are equal for January 2019, as only twelve months are available at that time. Hence from this plot it can be derived that the optimal t_train is at least larger than 12 months.

Secondly the figure shows that despite having a fixed amount of training data, the performance of the linear predictor can differ significantly. For example for t_train equal to 12 the $|\epsilon| < 2$ min differs 11% between the months of August and September.

Thirdly the figure shows in conjunction with tables 8 and 6 that the test period of the simple evaluation method, around the final 3.6 months of 2019, is representative of an average month. For instance, the $|\epsilon| < 2$ min of the linear predictor on the simple test data is equal 60.91% while the average over the twelve months with infinite t_train (effectively on average 18 months) is equal to 60.51%, a difference of only 0.4%. This difference can additionally in part be attributed to the on average larger t_train (~20 months) used by simple time series split evaluation. The results gathered from the simple time series split evaluation are therefore believed to be representative of the expected performance.

Finally, the monthly $|\epsilon| < 2$ has been compared to the monthly average queuing time, described in section V. However no correlation between both parameters are found. Further investigation is therefore needed to determine the cause of the monthly differences in $|\epsilon| < 2$.

IX. Conclusion

In conclusion the Google AutoML Tables predictor outperforms the other predictors on the simple time series split test data. The predictor is able to use the power of the cloud to trial a large number of predictor variations in quick succession. The resulting predictor is able achieve $|\epsilon| < 2$ min 67.91% on the simple time series split test data at a 30 min prediction horizon.

The downside of this method is a lack of transparency in the construction of the final model due to the process not being open source. Additionally each training session requires financial resources. For this reason cross-validated time series split evaluation, requiring multiple training sessions, could not be performed. However applying cross-validated performance evaluation of the linear predictor revealed that the simple time series split evaluation is representative for the performance of an average month. Additional analysis on the linear predictor showed that training a model for a specific prediction horizon can enable a predictor to perform better at that prediction horizon.

For the other models, the manually designed tree-based ensemble predictor has a $|\epsilon| < 2$ min of 66.84% on the simple time series split test data at a 30 min prediction horizon, just 1.07% lower than the Google AutoML Tables predictor while being fully open source. Similarly the manual decision tree predictor has a $|\epsilon| < 2$ min value of 64.41%, using only three features of a single data source. The benefits of both predictors could provide a reason to chose them over the better performing Google AutoML predictor.

Next, the predictors proposed in this article rely on features which are partially predictions predictions by themselves, i.e. weather forecasts. Further investigation could be performed to determine and reduce the error in these features, therefore improving the performance of the predictors.

Additionally in the future, it should be researched whether taxi time predictors can reduce air traffic controller workload during anomalous operations. In this article a performance feedback correction has specifically been developed to solve this problem. However more extensive evaluations are required to determine its effect during anomalous operations.

Finally, as stated in the introduction, the purpose of improving taxi time prediction is to reduce the uncertainties in the departure process, allowing further optimization of the ground process. However, more research is required to determine the impact of the improved predictors on the operations. Based on this trade-off stakeholders can make an informed decision on the taxi time predictor to be implemented.

Appendix

Table 9 Comparison of linear predictor variations performance on all features of the validation data from simple time series split

Variation	$\|\epsilon\| < 2$ min [%]	MAE [s]	RMSE [s]	$\|\epsilon\| < 5$ min [%]	$\|\epsilon\| < 7$ min [%]	t [s]
ARDRegression	59.14	125.89	181.11	93.3	97.58	5301
BayesianRidge	59.18	125.69	180.77	93.4	97.59	152
ElasticNet	45.2	164.31	224.6	87.14	94.94	10
ElasticNetCV	54.6	137.23	194.15	91.72	96.93	249
GammaRegressor	41.64	178.62	242.07	84.45	93.37	6
HuberRegressor	60.76	124.41	181.97	93.02	97.4	241
Lars	58.5	127.21	182.18	93.33	97.45	27
LarsCV	58.48	127.36	182.42	93.31	97.43	124
Lasso	56.24	133.02	189.52	92.3	97.14	203
LassoCV	59.05	126.74	182.45	93.2	97.44	1212
LassoLars	31.23	225.57	292.29	73.48	88.6	21
LassoLarsCV	57.85	128.62	184.07	93.12	97.34	117
LassoLarsIC	57.85	128.62	184.07	93.12	97.34	23
LinearRegression	58.5	127.4	182.72	93.13	97.54	63
OrthogonalMatchingPursuit	58.94	127.04	182.68	93.15	97.44	22
OrthogonalMatchingPursuitCV	58.88	127.09	182.75	93.15	97.44	329
PoissonRegressor	58.91	127.38	182.94	92.94	97.44	19
RANSACRegressor	43.61	175.84	236.79	83.34	93.59	77
Ridge	58.97	125.98	181.08	93.36	97.58	9
RidgeCV	59.16	125.79	180.9	93.39	97.6	69
SGDRegressor	59.23	125.63	180.81	93.36	97.58	11
TweedieRegressor	41.29	178.13	239.81	84.71	93.74	6

Table 10 Comparison of tree-based ensemble predictor variations performance on all features of the validation data from simple time series split

Predictor	$\|\epsilon\| < 2$ min [%]	MAE [s]	RMSE [s]	$\|\epsilon\| < 5$ min [%]	$\|\epsilon\| < 7$ min [%]	t [s]
BaggingRegressor	63.58	120.44	181.32	92.98	97.23	1159.0
ExtraTreesRegressor	64.19	120.60	184.64	92.55	97.01	14593.0
AdaBoostRegressor	9.48	419.73	476.10	29.25	49.39	37.0
RandomForestRegressor	65.40	116.12	176.15	93.40	97.49	9049.0
HistGradientBoosting-Regressor	63.84	117.36	173.90	93.91	97.73	284.0
HistGradientBoosting-Regressor LAD	65.96	116.03	176.77	93.17	97.36	549.0
GradientBoosting-Regressor	58.65	127.10	183.19	93.23	97.31	69.0
GradientBoosting-Regressor LAD	61.72	125.11	185.25	92.43	97.00	72.0
GradientBoosting-Regressor Huber	60.05	126.02	183.76	92.93	97.16	75.0
GradientBoosting-Regressor Quantile	24.11	220.51	262.42	75.23	94.22	72.0

Table 11 Name, type, source, and description of the features used by the predictors.

Name	Type	Source	Description
actype	cat	flt (constructed)	Aircraft type (icao designator)
alr1	cat	cfs	Alternative landing runway 1
alr2	cat	cfs	Alternative landing runway 2
atr1	cat	cfs	Alternative take-off runway 1
atr2	cat	cfs	Alternative take-off runway 2
cb	num	skv	Probability of CB
depgnr	cat	flt (constructed)	Departure gate number
dew	num	skv	Dew point
lightning	num	skv	Probability of lightning
local_dow	cat	flt (constructed)	Local day of week (0-6)
local_doy	cat	flt (constructed)	Local day of year (0-365) (qcut into 24 bins)
local_mod	cat	flt (constructed)	Local minute of the day (0-86400) (qcut into 48 bins)
local_week	cat	flt (constructed)	Local week of year (0-53)
n_arr	num	flt (constructed)	Number of arrival of non civil aircraft with arrival time within 10 minutes of own off-block time
n_civil	num	flt (constructed)	Number of civil aircraft with off-block time or arrival time within 10 minutes of own off-block time
n_dep	num	flt (constructed)	Number of departure of non civil aircraft with off-block time within 10 minutes of own off-block time
plr1	cat	cfs	Primary landing runway 1
plr2	cat	cfs	Primary landing runway 2
ptr1	cat	cfs	Primary take-off runway 1
ptr2	cat	cfs	Primary take-off runway 2
rain_cool	num	skv	Supercooled precipitation
rvr1500_300	num	skv	Probability of runway visual range < 1500 m and/or cloud base <300 ft
rvr350	num	skv	Probability of runway visual range < 350 m
rvr5000_1000	num	skv	Probability of < 5 km vision and/or cloud base < 1000 ft
rvr5000_2000	num	skv	Probability of < 5 km vision and/or cloud base ≤ 200 ft
rvr550_200	num	skv	Probability of runway visual range < 550 m and/or cloud base < 200 ft

Continued on next page

Name	Type	Source	Description
rvrcat	cat	skv	Combined vision and cloud base category: Good (Vision >= 5 km and cloud base ≤ 1000 ft), Marginal, fase A, fase B, fase C"
sid	cat	flt (constructed)	Standard instrument departure
snow	num	skv	Probability of snow
snow_heavy	num	skv	Probability of medium/heavy snow
temp	num	skv	Temperature
trwy	cat	flt (constructed)	Take-off runway
trwy_ext	cat	flt (constructed)	Extended take-off runway with taxi way for 36L (W, Y, Z)
wind_dir	cat	skv	Wind direction
wind_dir_std	num	skv	Standard deviation of wind direction
wind_spd	num	skv	Wind speed
wind_spd_std	num	skv	Standard deviation of wind speed
wind_stoten	num	skv	Wind gusts
wtc	cat	icao	Wake turbulence category
t_taxi	y	astra (constructed)	Time of last entry of a runway polygon while not in flight - Time of last red zone polygon exit

Code

The code for this article is published in a public repository accessible through the following url:

`https://github.com/EKPyqh40/Taxi-Time-Prediction-Schiphol-Airport`

References

[1] Simaiakis, I., Khadilkar, H., Balakrishnan, H., Reynolds, T. G., and Hansman, R. J., "Demonstration of reduced airport congestion through pushback rate control," *Transportation Research Part A: Policy and Practice*, Vol. 66, 2014, pp. 251 – 267. https://doi.org/https://doi.org/10.1016/j.tra.2014.05.014, URL http://www.sciencedirect.com/science/article/pii/S0965856414001384.

[2] Hao, L., Ryerson, M. S., Kang, L., and Hansen, M., "Estimating fuel burn impacts of taxi-out delay with implications for gate-hold benefits," *Transportation Research Part C: Emerging Technologies*, Vol. 80, 2017, pp. 454 – 466. https://doi.org/https://doi.org/10.1016/j.trc.2016.05.015, URL http://www.sciencedirect.com/science/article/pii/S0968090X16300523.

[3] Liu, Y., Hansen, M., Gupta, G., Malik, W., and Jung, Y., "Predictability impacts of airport surface automation," *Transportation Research Part C: Emerging Technologies*, Vol. 44, 2014, pp. 128 – 145. https://doi.org/https://doi.org/10.1016/j.trc.2014.03.010, URL http://www.sciencedirect.com/science/article/pii/S0968090X14000825.

[4] Bouras, A., Ghaleb, M. A., Suryahatmaja, U. S., and Salem, A. M., "The airport gate assignment problem: a survey," *The scientific world journal*, Vol. 2014, 2014.

[5] Günther, T., Hildebr, M., Fricke, H., and Strasser, M., "Contributions of Advanced Taxi Time Calculation to Airport Operations Efficiency," *Air Transport and Operations : Proceedings of the First International Air Transport and Operations Symposium 2010*, 2010, pp. 95–106.

[6] Atkin, J. A., Burke, E. K., and Ravizza, S., "The airport ground movement problem: Past and current research and future directions," *Proceedings o f the 4th International Conference on Research in Air Transportation (ICRAT), Budapest, Hungary*, 2010, pp. 131–138.

[7] Atkin, J. A. D., Uyar, A. S., Ozcan, E., and Urquhart, N., *Airport Airside Optimisation Problems*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, Chap. 1, pp. 1–37. https://doi.org/10.1007/978-3-642-39304-4_1, URL https://doi.org/10.1007/978-3-642-39304-4_1.

[8] Bennell, J. A., Mesgarpour, M., and Potts, C. N., "Airport runway scheduling," *4OR*, Vol. 9, No. 2, 2011, p. 115. https://doi.org/10.1007/s10288-011-0172-x, URL https://doi.org/10.1007/s10288-011-0172-x.

[9] Atkin, J. A. D., Burke, E. K., Greenwood, J. S., and Reeson, D., "On-line decision support for take-off runway scheduling with uncertain taxi times at London Heathrow airport," *Journal of Scheduling*, Vol. 11, No. 5, 2008, p. 323.

[10] Herrema, F., Curran, R., Visser, H., Huet, D., and Lacote, R., "Taxi-out time prediction model at Charles de Gaulle Airport," *Journal of Aerospace Information Systems*, Vol. 15, No. 3, 2018, pp. 120–130.

[11] Vakaet, C., *Taxi Time Prediction at Schiphol Airport Midterm Report*, Delft University of Technology, 2020.

[12] Pina, P., and De Pablo, J. M., "Benefits obtained from the estimation and distribution of realistic taxi times," *ATM R&D Seminar*, 2005, pp. 693–723.

[13] Chatterji, G. B., and Zheng, Y., "Wheels-Off Time Estimation at Non-ASDE-X Equipped Airports," *2013 Aviation Technology, Integration, and Operations Conference*, 2013. https://doi.org/10.2514/6.2013-4274, URL https://arc.aiaa.org/doi/abs/10.2514/6.2013-4274.

[14] Zelinski, S., and Windhorst, R., "Modelling and simulating airport surface operations with gate conflicts," *The Aeronautical Journal*, Vol. 123, No. 1259, 2019, pp. 1–19.

[15] Lee, H., Malik, W., Zhang, B., Nagarajan, B., and Jung, Y. C., "Taxi Time Prediction at Charlotte Airport Using Fast-Time Simulation and Machine Learning Techniques," *15th AIAA Aviation Technology, Integration, and Operations Conference*, 2015. https://doi.org/10.2514/6.2015-2272, URL https://arc.aiaa.org/doi/abs/10.2514/6.2015-2272.

[16] Khadilkar, H., and Balakrishnan, H., "Network congestion control of airport surface operations," *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 3, 2014, pp. 933–940.

[17] Shumsky, R. A., “Dynamic statistical models for the prediction of aircraft take-off times,” Ph.D. thesis, Massachusetts Institute of Technology, 1995.

[18] Idris, H., Clarke, J.-P., Bhuva, R., and Kang, L., “Queuing model for taxi-out time estimation,” *Air Traffic Control Quarterly*, Vol. 10, No. 1, 2002, pp. 1–22.

[19] Signor, D. B., and Levy, B. S., “Accurate OOOI Data: Implications for Efficient Resource Utilization,” *2006 ieee/aiaa 25TH Digital Avionics Systems Conference*, 2006, pp. 1–12. https://doi.org/10.1109/DASC.2006.313676.

[20] Laskey, K. B., Xu, N., and Chen, C.-H., “Propagation of delays in the national airspace system,” *UAI’06: Proceedings of the Twenty-Second Conference on Uncertainty in Artificial Intelligence*, 2006.

[21] Balakrishna, P., “Scalable approximate dynamic programming models with applications in air transportation,” Ph.D. thesis, George Mason University, 2009.

[22] Balakrishna, P., Ganesan, R., and Sherry, L., “Accuracy of reinforcement learning algorithms for predicting aircraft taxi-out times: A case-study of Tampa Bay departures,” *Transportation Research Part C: Emerging Technologies*, Vol. 18, No. 6, 2010, pp. 950 – 962. https://doi.org/https://doi.org/10.1016/j.trc.2010.03.003, URL http://www.sciencedirect.com/science/article/pii/S0968090X1000029X, special issue on Transportation Simulation Advances in Air Transportation Research.

[23] Kistler, M., and Gupta, G., “Relationship Between Airport Efficiency and Surface Traffic,” *9th AIAA Aviation Technology, Integration, and Operations Conference (ATIO)*, 2009. https://doi.org/10.2514/6.2009-7078, URL https://arc.aiaa.org/doi/abs/10.2514/6.2009-7078.

[24] Chatterji, G., and Zheng, Y., “Wheels-Off Time Prediction Using Surface Traffic Metrics,” *12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference and 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2012. https://doi.org/10.2514/6.2012-5699, URL https://arc.aiaa.org/doi/abs/10.2514/6.2012-5699.

[25] Ravizza, S., Atkin, J. A. D., Maathuis, M. H., and Burke, E. K., “A combined statistical approach and ground movement model for improving taxi time estimations at airports,” *Journal of the Operational Research Society*, Vol. 64, No. 9, 2013, pp. 1347–1360. https://doi.org/10.1057/jors.2012.123, URL https://doi.org/10.1057/jors.2012.123.

[26] Ravizza, S., Chen, J., Atkin, J. A., Stewart, P., and Burke, E. K., “Aircraft taxi time prediction: Comparisons and insights,” *Applied Soft Computing*, Vol. 14, 2014, pp. 397 – 406. https://doi.org/https://doi.org/10.1016/j.asoc.2013.10.004, URL http://www.sciencedirect.com/science/article/pii/S1568494613003384.

[27] Lee, H., Malik, W., and Jung, Y. C., “Taxi-out time prediction for departures at Charlotte airport using machine learning techniques,” *16th AIAA Aviation Technology, Integration, and Operations Conference*, 2016, p. 3910.

[28] Lee, H., Coupe, J., and Jung, Y. C., “Prediction of Pushback Times and Ramp Taxi Times for Departures at Charlotte Airport,” *AIAA Aviation 2019 Forum*, 2019, p. 2933.

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

[30] Futer, A., "Improving Etms' Ground Time Predictions," *2006 ieee/aiaa 25TH Digital Avionics Systems Conference*, 2006, pp. 1–12. https://doi.org/10.1109/DASC.2006.313692.

[31] *Traffic Review*, Schiphol Airport, 2018.

[32] *Traffic Review*, Schiphol Airport, 2019.

[33] Huber, P. J., *Robust statistics*, Vol. 523, John Wiley & Sons, 2004.

[34] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T.-Y., "LightGBM: A Highly Efficient Gradient Boosting Decision Tree," *Advances in Neural Information Processing Systems*, Vol. 30, edited by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf.

[35] Chen, T., and Guestrin, C., "XGBoost," *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016. https://doi.org/10.1145/2939672.2939785, URL http://dx.doi.org/10.1145/2939672.2939785.

[36] Olson, R. S., Urbanowicz, R. J., Andrews, P. C., Lavender, N. A., Kidd, L. C., and Moore, J. H., *Applications of Evolutionary Computation: 19th European Conference, EvoApplications 2016, Porto, Portugal, March 30 – April 1, 2016, Proceedings, Part I*, Springer International Publishing, 2016, Chaps. Automating Biomedical Data Science Through Tree-Based Pipeline Optimization, pp. 123–137. https://doi.org/10.1007/978-3-319-31204-0_9, URL http://dx.doi.org/10.1007/978-3-319-31204-0_9.

[37] Jin, H., Song, Q., and Hu, X., "Auto-Keras: An Efficient Neural Architecture Search System," *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Association for Computing Machinery, New York, NY, USA, 2019, p. 1946–1956. https://doi.org/10.1145/3292500.3330648, URL https://doi.org/10.1145/3292500.3330648.

Midterm Report

Taxi Time Prediction at Schiphol Airport

Midterm Report

C. Vakaet

Unchanged Image by Atamari CC BY-SA 3.0

Taxi Time Prediction at Schiphol Airport

Midterm Report

by

C. Vakaet

to obtain the degree of Master of Science
at the Delft University of Technology,

Student number:	4353099	
Project duration:	September 16, 2019 – October 28, 2020	
Thesis Supervisor:	Prof. dr. ir. J. M. Hoekstra	TU Delft
	Dr. ir. J. Ellerbroek	TU Delft
	F. Dijkstra	Knowledge Development Center

Contents

List of Symbols **v**

Abbreviations & Acronyms **vii**

1 Introduction **1**

2 Background **3**

2.1 Project Setup and Planning 3

2.2 Taxi Time Prediction within Air Traffic Control 3

2.2.1 Taxi Time Prediction and Surface Congestion Management 3

2.2.2 Airport Collaborative Decision Making. 4

2.3 Schiphol Airport 5

2.3.1 Airport Lay-out and Runway Usage 5

2.3.2 Current Taxi Time Predictions 7

3 Literature Review **9**

3.1 Simulation-Based Taxi Time Prediction Models 9

3.1.1 Pertinent Papers 9

3.1.2 Discussion & Research Gap 10

3.2 Data-Driven Taxi Time Prediction Models. 10

3.2.1 Pertinent Papers 10

3.2.2 Discussion & Research Gap 13

3.3 Cost-Benefit Analysis of Taxi Time Prediction 17

3.3.1 Pertinent Papers 17

3.3.2 Discussion & Research Gap 18

3.4 Industry Goals 20

4 Research Objectives & Questions **23**

4.1 Research Objectives. 23

4.2 Research Questions 24

5 Methodology **25**

5.1 Data Understanding 25

5.2 Data Preparation 26

5.2.1 Select Data 27

5.2.2 Clean Data 27

5.2.3 Construct Data. 27

5.2.4 Integrate Data 28

5.2.5 Format Data 28

5.3 Modelling 29

5.3.1 Select Modelling Type 29

5.3.2 Generate Test Design 30

5.3.3 Build Model 30

Bibliography **33**

List of Symbols

Greek

σ	Standard Deviation
μ	Mean

Abbreviations & Acronyms

AAS	Amsterdam Airport Schiphol
A-CDM	Airport Collaborative Decision Making
CTOT	Calculated Take-Off Time (from Central Flow Management)
DSP	Departure Sequence Planner
GNr	Gate Number
HIL	Human-in-the-loop
LVNL	Air Traffic Control the Netherlands
MAE	Mean Absolute Error
RMSE	Root Mean Square Error
SARDA	Spot And Runway Departure Advisor
SID	Standard Instrument Departure
STAR	Standard Terminal Arrival Route
TOBT	Target Off-Block Time
TSAT	Target Start-up Approval Time
TTOT	Target Take-Off Time
TTOT’	Earliest possible Target Take-Off Time
TU Delft	Delft University of Technology
WTC	Wake Turbulence Category

1 Introduction

Air traffic control attempt to plan aircraft arrivals and departures such that capacity is maximized, while maintaining safety and minimizing the environmental impact. One way to minimize the environmental impact of air travel is to reduce the time an aircraft spends queuing at the runway [57]. It has been estimated that a major U.S.-based airline with an extensive domestic network could reduce its fuel consumption by 1%, if taxi-delay were eliminated based on flight data from 2012 to 2013 [25]. To eliminate or reduce this taxi delay, air traffic controllers can tactically hold an aircraft at a gate or parking spot engines off rather than have the airplanes wait in line with engines on. An additional benefit of holding aircraft at the gate is an increase in passenger connection rate by giving passengers more time to catch the flight [39]. However it is important to note that gate availability can often be more critical thus giving the benefit to holding at parking stands [8, 23].

Although the concept of holding might appear simple at first glance, it is difficult to execute efficiently within the airport operations. Care should be taken that the runway capacity is maintained, the workload for air traffic control is manageable, and no unnecessary delays at the gate are introduced. To manage these problems different ground control systems have been developed and implemented [3, 4]. Many of these systems rely on and are hindered by inaccuracies of the estimated taxi-time to plan the different aircraft [2, 7]. Current operational systems rely on simple and often inaccurate taxi-time estimates leading to efficiency losses and added workload for controllers. In addition, improved taxi-time predictions would improve passenger information and allow for further optimization of different ground processes.

The goal of this project is therefore to improve taxi-time estimations at airports by analyzing the practical performance of different prediction methodologies and input parameters for different prediction horizons. While different methodologies and input parameters have already been developed and compared in literature [27], this project intends to focus on different prediction horizons and the ability of models to perform in difficult to predict scenarios. It is the aspiration of this project to aid decision makers to implement advanced taxi-time predictors in the ground control system. This document is a midterm report and includes all elements of the project except for the results and evaluation as these are currently being generated.

This report has been subdivided into five chapters. Chapter 2 describes the project setup, taxi time prediction within air traffic control, the particularities of Schiphol airport, industry goals, and the project planning. Next in Chapter 3 a literature review of the taxi-time prediction subject is presented. Subsequently in Chapter 4 research objectives and questions are detailed. Finally in Chapter 5 the methodologies behind the taxi-time prediction models that will be used in this project are described.

2
Background

This chapter provides the broader context of the project and details where taxi time prediction fits within the broader area of air traffic control. The chapter is divided into three sections. Section 2.1 describes the different project stakeholders and the project plan. Next, section 2.2 describes taxi time prediction within the larger field of ground control. Additionally the section describes how taxi time prediction fits within Eurocontrols Airport Collaborative Decision Making (A-CDM) framework. Finally in section 2.3 the particularities of Schiphol Airport with respect to taxi operations, such as the airport lay-out and runway usage, and the current taxi time prediction method is described.

2.1. Project Setup and Planning

This project is a collaboration between Delft University of Technology (TU Delft) and the Knowledge Development Centre mainport Schiphol through the Centre of Excellence. The Centre of Excellence is a program by the Knowledge Development Centre which allows bachelor and master students to complete their thesis together with the industry partners involved in the Knowledge Development Centre: Air Traffic Control the Netherlands (LVNL), Royal Dutch Airlines, Amsterdam Airport Schiphol [1]. In this cooperation the university provides the academic support to the project, while the Knowledge Development Centre provides financial and technical support. Through this collaboration the project has access to the resources, including the relevant data, of the organizations involved and implies a focus on Schiphol airport.

2.2. Taxi Time Prediction within Air Traffic Control

This section describes where taxi time fits within air traffic control (2.2.1), and the operational concept A-CDM (2.2.2).

2.2.1. Taxi Time Prediction and Surface Congestion Management

Taxi time prediction is used by airports to optimize ground movement through surface congestion management. Surface congestion management strategies differ significantly between airports. In small airports, aircraft may simply be routed the shortest path without any strategic planning, hence no taxi time prediction is needed. For larger airports scheduling and routing is needed to avoid conflicts, minimize taxi time, or arrive at the runway at the expected take-off time.

Surface congestion management is strongly linked with arrival-, departure-, and gate-stand management. These systems rely on the ability for aircraft to be at the runway or gate-stand at a particular time. Hence it is necessary to determine the time it takes for an aircraft to reach a particular point through taxi time prediction. This project solely focuses on the aspect of taxi time prediction, and does not cover the wider area of ground control strategies. More information and literature review on different ground control strategies can be found in the 2010 article and 2013 book by Atkins et al. [3, 4]. Additionally, Sandberg et al. provides an overview of

[1] `https://kdc-mainport.nl/centre-of-excellence/` Accessed: 13-01-2020

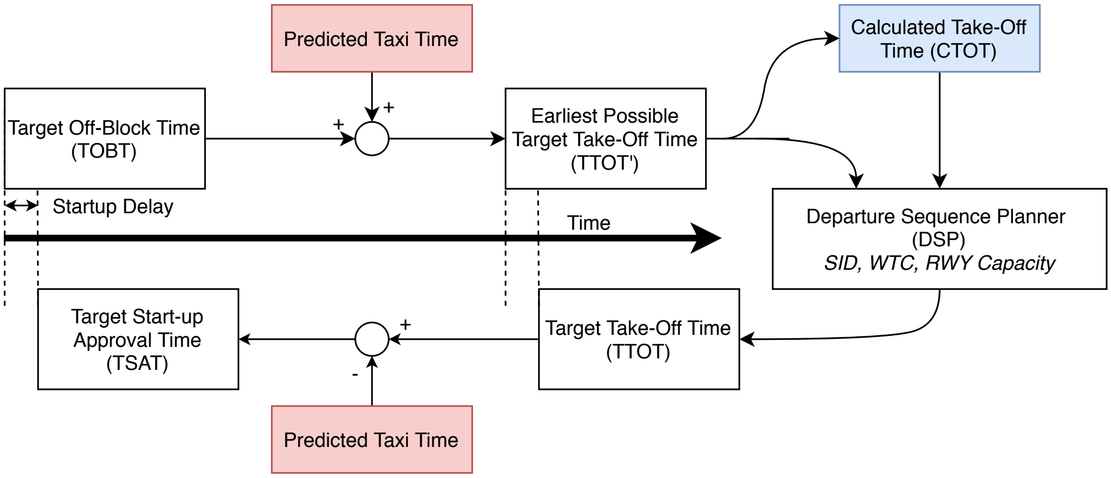
Predicted Taxi Time
Calculated Take-Off Time (CTOT)
Target Off-Block Time (TOBT)
+
+
Earliest Possible Target Take-Off Time (TTOT')
Startup Delay
Time
Departure Sequence Planner (DSP)
SID, WTC, RWY Capacity
Target Start-up Approval Time (TSAT)
+
-
Target Take-Off Time (TTOT)
Predicted Taxi Time

Figure 2.1: Visualization of the departure management process

different surface congestion management approaches [49].

2.2.2. Airport Collaborative Decision Making

A-CDM is an operational concept by Eurocontrol which enables the air traffic management network and airport stakeholders to improve the operational efficiency, predictability and punctuality [17]. The A-CDM concept is currently increasingly being adopted growing from 16 airports in 2016 [16] to 27 as of 2019[2], additional Schiphol airport is a A-CDM airport. The A-CDM consists of five main elements plus the element "Information Sharing" which is essential for all other elements to be implemented.

A-CDM is of interest to this project since one of the five main elements is the "Variable Taxi Time". This element describes how taxi times should be calculated and used within A-CDM and the departure manager. In short, the taxi time should be used to calculate and provide Estimated In-Block Times, Target Take-Off Times (TTOT) and Target Start-up Approval Times (TSAT) to the airport partners.

Note: Eurocontrol uses the term Variable Taxi Time for both the concept as well as the estimated taxi time within the concept. To avoid confusion, this project will not use Variable Taxi Time for the estimated taxi time and instead call it simply the taxi time.

The process to calculate TTOT and TSAT using the estimated taxi time has been visualized in figure 2.1. From this figure it can be seen that the departure manager sums the taxi time with the Target Off-Block Time (TOBT) where TOBT is the estimated time when the aircraft is ready to start up or push back immediately upon reception of clearance from the tower. TOBT should be provided by the operator or handling agent. This summation results in the earliest possible Target Take-Off Time (TTOT'), which is used by the Central Flow Management unit to potentially allocate a Calculated Take-Off Time (CTOT). When a CTOT is returned by the Central Flow Management the flight is expected to depart within the time window five minutes before or ten minutes after CTOT.

Next the Departure Sequence Planner (DSP), a set of algorithms often specific to each airport, determines TTOT. The DSP is another one of the five elements of the A-CDM concept where it is called Collaborative Pre-departure Sequence. the DSP uses the TTOT' and CTOT together with the aircraft's Standard Instrument Departure (SID), Standard Terminal Arrival Route (STAR), Wake Turbulence Category (WTC), and the runway capacity to determine TTOT. Subsequently the estimated taxi time is substracted from the TTOT to determine the TSAT. Finally this TSAT is sent back to the pilot and indicates the time window five minutes before or after in which the pilot is expected to call ready and receive start-up approval by the tower.

[2] `https://www.eurocontrol.int/concept/airport-collaborative-decision-making` Accessed 29-01-2020

By using taxi time estimations in this manner, inaccuracies in the predictions result in potential misses of the CTOT window and additional workload for Air Traffic Management due to the required corrections. Inaccuracies in take-off times also reduce passenger information, predictability, and impede further operational optimization.

2.3. Schiphol Airport

As discussed in the previous section this project focuses on Schiphol airport. Schiphol airport is a large hub airport located in the Netherlands. In 2018 Schiphol airport was the second largest airport in Europe in terms of air transport movements [50]. While it is believed that this research can be generalized to other large airports with similar available data this section discusses the particularities of Schiphol airport. This section has been subdivided into two sections with the first subsection, subsection 2.3.1, discussing the airport lay-out and runway usage. Subsection 2.3.2 discusses the current method for calculating taxi times at Schiphol airport.

2.3.1. Airport Lay-out and Runway Usage

In this section the main characteristics of the lay-out and taxi operations at Schiphol airport are discussed. In figure 2.2 the six available runways at Schiphol airport can be seen. Additionally the dark blue and light green airplane symbols indicate the possible landing and take-off directions respectively for each runway. It is also important to note that runway 04/22 is significantly shorter. It is generally only used for general aviation, but can also be used in case of a south-westerly storm [3].

On figure 2.2 it can be seen that the runways are spread out significantly. Runway 18R/36L is particularly far removed from the central hub. Additionally, when runway 18C/36C is in use, aircraft taxiing towards or away from 18R/36L may need to use taxiway Yankee (Y) or Zulu (Z) rather than Whisky five (W5) increasing the taxi distance significantly. This indicates that the runway configuration in use will have a significant impact on the taxi time at Schiphol airport.

Also shown on the chart are taxiways Victor (V) and Quebec (Q). Both these taxiways are one way with no alternative, hence constrain the taxi operations significantly, especially during changes of the runway configuration. Taxiway Q is also of interest since closure of the taxiway requires aircraft to make significant detours. The status of taxiway Q could therefore be an important parameter to determine the taxi time.

Lastly taxiways S2 and S8 both cross runway 06/24 and are the only exit for ramp Sierra (S). It has therefore been found that the taxi time, for aircraft departing/arriving at S, is highly dependent on the ability to cross these two taxiways. Hence the landing/take-off capacity of runway 06/24 used might determine the taxi time for ramp S.

Figure 2.3 shows the parking and docking chart of Schiphol centre. The figure shows Schiphol airport' seven piers, the Sierra ramp and their gate layout. Additionally one can see the two main taxiways surrounding the centre Alpha (A) and Bravo (B). As indicated in the figure, it is customary for traffic to travel clockwise on A and anti-clockwise on B. Also on this figure it is clear that taxiway Q constrains airport operations as it is the single one way section circling the central hub.

During normal daytime operations Schiphol airport has three active runways. The number of take-off runways tend to alternate between one and two depending on whether there is an inbound or outbound peak respectively. In some circumstances, mainly during runway configuration transitions, also a forth runway is active. In table 2.1 the main runway configurations and their preferences are given. Table 2.2 provide the expected usage of each preference level for runway configurations in 2020 during daytime (06:00-23:00). During nighttime operations (23:00-06:00) generally two runways are active, one for take-off and one for landing.

[3] https://www.schiphol.nl/en/schiphol-as-a-neighbour/page/is-my-house-underneath-a-flight-path/ Accessed 13-01-2020

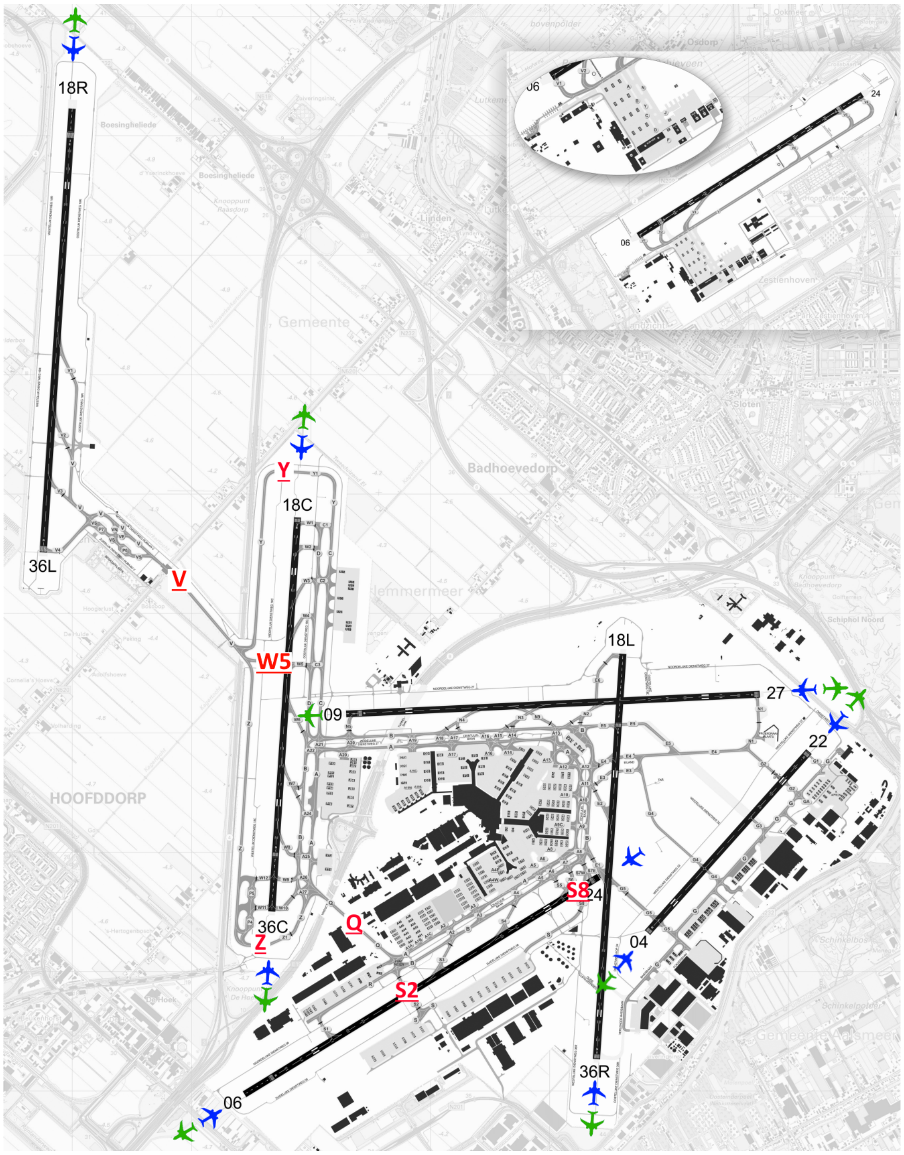

Figure 2.2: Aerodrome ground movement chart

Table 2.1: Main landing (L) and take-off (S) runway configurations and their preference during daytime (06:00-23:00) [15]

Preference	L1	L2	S1	S2
1	06	(36R)	36L	(36C)
2	18R	(18C)	24	(18L)
3	06	(36R)	09	(36L)
4	27	(18R)	24	(18L)
5a	36R	(36C)	36L	(36C)
5b	18R	(18C)	18L	(18C)
6a	36R	(36C)	36L	(09)
6b	18R	(18C)	18L	(24)

Table 2.2: Usage prognosis of each preference level for runway configurations in 2020 during daytime (06:00-23:00) [15]

Preference	Movements	Fraction
1	127,600	26.7%
2	182,400	38.1%
3	5,800	1.2%
4	22,600	4.7%
5	71,600	15.0%
6	1,400	0.3%
Other	67,400	14.1%

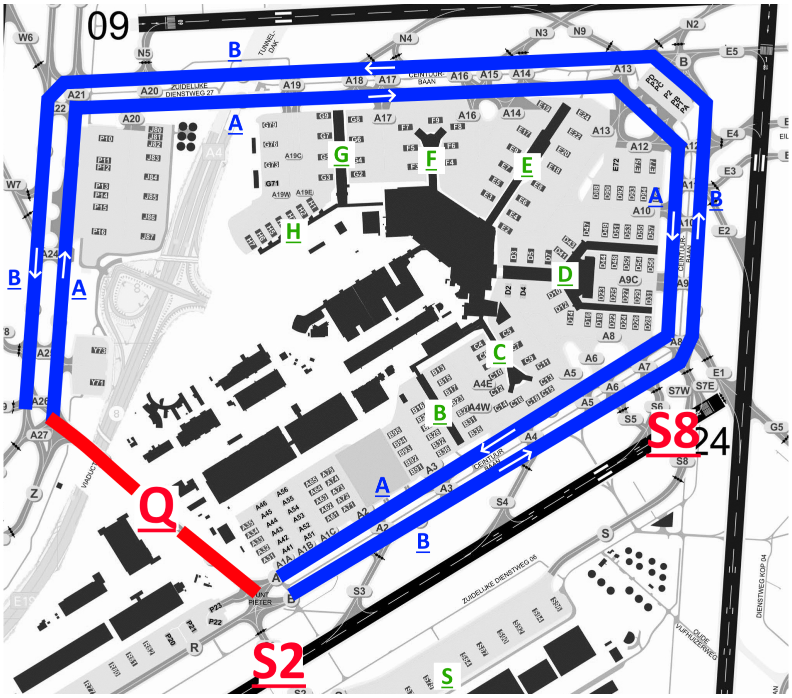

Figure 2.3: Aircraft parking and docking chart Schiphol centre

Table 2.3: Snapshot of Schiphol airports taxi time look-up table

GNr	Cond.	04	06	09	18C	18L	22	24	27	36L(W5)	36L(Y)	36L(Z)	36C
A31	01	0	780	900	960	1020	0	840	1380	1200	1320	1200	840
A31	02	720	960	780	720	720	900	540	780	840	1140	900	480
A31	04	0	1680	960	1260	1800	0	1320	1800	1500	1800	1440	1080
A31	05	0	1500	1020	900	1320	0	1140	1800	1080	1620	1080	780
A32	01	0	780	1140	960	1080	0	840	1380	1260	1260	1260	960
A32	02	720	960	780	720	720	900	540	780	840	1140	900	480
A32	05	0	1500	1020	900	1380	0	1200	1800	1080	1620	1080	780
A33	01	0	780	1140	960	1080	0	840	1380	1260	1260	1260	960

2.3.2. Current Taxi Time Predictions

Schiphol airport is am A-CDM airport and therefore uses the process described in section 2.2.2. Currently Schiphol airport derives the estimated taxi time through a look-up table. A snapshot of the look-up table is given in table 2.3. In the look-up table taxi times are given from each Gate Number (GNr) to each runway for different conditions. The conditions have been divided into twelve categories based on different combinations of WTC, deicing location, and S-ramp/S8/S2 (see subsection 2.3.1) considerations. Lastly it is important to note that for runway 36L three different taxi times are given based on the taxiway used (W5, Y, Z) due to the lay-out considerations again described earlier in section 2.3.1. Schiphol Airports look-up table has been made by averaging historic taxi times. The table does not update automatically but relies on manual updates. As can be seen from the snapshot, the estimations have been rounded to the minute, the reasoning behind this is discussed in section 3.4.

3

Literature Review

For this project an extensive literature review has been performed. This chapter describes and discusses the main literature on the topic of taxi time prediction and surface congestion management. Within literature a large number of articles and conference papers have been written on the topic. This chapter will highlight a subset of these papers and articles revealing the development from pioneering models to the current state-of-the-art. While recent developments may perform better, older articles help to understand the challenges and broader context of the problem. In this literature review a focus is put on highlighting the value that each article brought to the project, rather than describing the content of each article.

This chapter has been subdivided into four sections. In the first section articles related to simulation-based taxi time prediction models are reviewed. The second section focuses on data-driven taxi time prediction models. Subsequently section three surveys articles on cost-benefit analysis of surface congestion management and taxi time prediction. Finally section four contains an analysis of the goals set by the industry.

Each section starts with a subsection listing the papers pertinent to the topic of the section in a generally chronological order. In the subsequent subsection of each section the knowledge gained from these papers is discussed and research gaps are identified.

3.1. Simulation-Based Taxi Time Prediction Models

This project defines simulation-based taxi time prediction models as models which predict taxi times through determining the individual tracks (position and speed) of the aircraft. These models tend to generate a network of the ground infrastructure and analyze the dynamics and interactions of the different vehicles and objects. This section is subdivided into two sections. Firstly the pertinent papers are listed and described. Subsequently take-aways and research gaps are discussed in the second subsection.

3.1.1. Pertinent Papers

One of the first articles that uses simulation-based taxi time prediction is a conference paper by Pina et al. from 2005 [45]. Based on a network of nodes and links, the model provides each aircraft an optimal taxi route for the given operational configuration, as well as the current and the predicted traffic. This model was subsequently compared to a statistical model based on the average taxi time for a given runway configuration and group of gates. From the comparison it is concluded that the value of the simulation over the data driven model cannot be substantiated.

Next up in 2013 Chatterji et al. applied a fast-time simulation of airport surface operations developed by the National Aeronautics and Space Administration called Surface Operations Simulator and Scheduler to the taxi time prediction problem [12]. The simulation contains kinematic models of aircraft and uses node-link graphs to simulate surface traffic. In the article by Chatterji et al. the simulator is only briefly described, but more information was found in an article by Zelinski et al. [62] or on the ational Aeronautics and Space Administrations website [1]. To predict taxi times using the simulator, additional runway crossing and separation

[1] https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20160004947.pdf Accessed: 28-01-2019

rules were added. The predictions by the simulator were compared to historical average taxi times based on gate, queue-area and runway. Although the simulator is extensive, the simple data driven model produced better estimates than the simulation.

Lastly, in 2015, Lee et al. presented a similar paper based on a fast-time simulation tool developed by US Airways (now American Airlines) called LINear Optimized Sequencing [35]. A business rule engine is used by the simulator to resolve taxi and gate conflicts in addition to resolving ground congestion. Like the other simulations, the simulator employs a node-link network. The network changes dynamically depending on the runway configuration, operational conditions, and aircraft capabilities. This time the simulation is compared to a swathe of data-driven methods including linear regression, support vector regression, k-nearest-neighbours, and random forest based methods. This time the simulation performs better than the linear regression model and equally well as the support vector regression model. However, the simulation does perform worse compared to models based on k-nearest-neighbours or random forest.

While there are other simulation-based models that predict taxi-out times such as the 2014 article by Khadilkar et al. [32], no examples were found where simulation convincingly outperformed a state-of-the-art data driven model.

3.1.2. Discussion & Research Gap

When analyzing the model types of the articles reviewed together, a couple clues can be gathered. First and foremost from the pertinent papers, it is clear that extensive simulations don't seem to be able to outperform data driven methods. Literature in general agrees that this lack of performance is caused by the highly stochastic nature of the taxi time problem. Because it is nearly impossible to simulate every possible aspect, it is believed that simulation will never be able to exactly predict the taxi time.

Simulation-based models do however have some benefits. Firstly these models can be used to analyze operational conditions for which no historical data is available. For example, these models can be used to derive the taxi times during maintenance or for a new airport lay-out. A further benefit of simulation based methods are their ability to update in real time. For example the model could easily be adopted to improve the taxi time prediction the closer the aircraft is to the runway.

Finally, one may think that a combination of a data driven and a simulation based method could provide a solution. However, the simulation-based models considered in this literature review are already a hybrid. Namely, the simulation models are calibrated with historical training data.

3.2. Data-Driven Taxi Time Prediction Models

Data-driven taxi time prediction models are defined in this project as taxi time prediction models that use a set of dependent variables to calculate the taxi time without directly simulating aircraft dynamics. As with the other sections in this chapter, this section is subdivided into two subsections: pertinent papers, and discussion & research gap.

3.2.1. Pertinent Papers

One of the first studies on aircraft taxi times is the phd thesis "Dynamic Statistical Models for the Prediction of Aircraft Take-off Times" by R. A. Shumsky (1995) [51]. In the study different linear models, significantly simpler than the recent models, are compared. Shumsky also analyzes dynamic models which adjust their prediction based on recent taxi times, not historic data. While the models are simple, the analysis is extensive and raises interesting questions. For example, it is one of the few studies that consider different prediction horizons. With the prediction horizon being the time between the prediction being made and the start of taxiing. Additionally the study decomposes the taxi time between pushback and rolling.

Next Idris et al. (2002) has developed a queuing model based on quadratic regression between taxi time and the number of aircraft on the airport surface [28]. In the article Idris et al. also performs parameter selection using linear regression and the R-squared value, showing the importance of queue size, departure demand, and airline/terminal on the taxi time. Additional analysis was done on the effect of runway configuration, weather and downstream restrictions.

In 2006 A. Futer published a conference paper where a working taxi time prediction model is improved by using rolling averages of the error to enhance the prediction [19]. This article shows that this can help to improve taxi time predictions during anomalous periods where taxi times of all aircraft are increased.

In the same year D.B. Signor and B. S. Levy presented a similar model using a bivariate quadratic polynomial regression [52]. This model has increased complexity, but was only developed and specifically optimized for predictions for a single runway configuration from one ramp to a single runway. This paper shows that complicated, manually constructed, regression models can improve taxi times, but at the cost of scalability.

A way to solve this problems is to allow the computer itself to determine the rules and relationships based on a set of training data. An early example of this concept, applied to the taxi time problem, is a model presented by K. B. Laskey again in 2006 [34]. For the model a Bayesian Network is manually constructed, but the tuning of the parameters is done using dirichlet-multinominal learning on training data.

Further research to enhance the scalability of models was performed by P. Balakrishna. Balakrishna's dissertation research (2009) provides a generic methodology for sequential decision making for large scale complex applications applied to the taxi time problem [5]. The resulting model based on a Markov Decision Process and reinforcement learning has also been applied in a case study at Tampa Bay [6].

Additionally in 2009 a paper was presented by Kistler et al. on taxi time prediction using a linear and log-linear model [33]. In the paper it is found that linear regression performs better than log-linear in their test set-up. The article also studies several independent variables such as the number and total time of stoppages during taxi. While these variables are unknown prior to take-off, it is possible to derive the importance of these variables on the taxi time through the linear model. Hence, one can analyse the significance of these variables.

From around 2012, it was found that the taxi time literature tends to converge towards seven types of models: linear regression, support vector regression, decision tree, random forest, k-nearest neighbours, and artificial neural network. This trend corresponds to the developments in the wider field of data science, and the availability and accessibility of tools such as scikit-learn [2].

One of the first examples of this trend is the 2012 conference paper by Chatterji on the taxi time prediction using a linear and artificial neural network model [11]. In the paper the models were trained on seven input parameters. The artificial neural network consists of three layers with 7 input nodes, 20 nodes in the hidden layer, and 1 output node each activated through the sigmoid function. The paper concludes that both models generated comparable results. In 2013 Chatterji published another article comparing a simulation-based and data driven model, more information about this article can be found in section 3.1.

Also in 2013 Ravizza et al. presented his first paper, improving taxi time estimations through linear regression for both arrivals and departures [46]. Additionally Ravizza highlights the difference in taxi time estimates for Europe and North America. In North America, aircraft are generally not held at the gate resulting in more queuing time. Hence, the papers focusing on queuing, such as the one by Idris described above, tend to work less well in Europe. Lastly, the paper concludes that comparison between models of different authors is hard due to the vastly different scenario's. According to Ravizza realistic benchmark scenario's that are close to reality could provide a solution.

In 2014, another paper was published by Ravizza which compared linear regression, support vector regression, regression tree, and two fuzzy rule-based models [47]. From the comparison it is concluded that the the TSK fuzzy rule-based model outperforms the other approaches on the same data set. It is however important to note that the results can be considered close. For example, the fraction of predictions within +/- 2 minutes of the actual taxi time differ by less than a percent between the fuzzy rule-based, the regression tree, and support vector regression models.

Lastly, in 2018 a paper by Brownlee et al., coauthored by Ravizza, was published which creates an improved fuzzy rule-based model. But the paper does not clearly provide or compares the customary perfor-

[2] `https://scikit-learn.org/` Accessed: 27/01/2020

mance values for taxi time prediction, due to its focus on ground control systems in general.

Going a little back in time, another researcher, Hanbong Lee, has written three conference papers comparing different taxi time models. The first paper from 2015, already described in the previous section on simulation-based methods, compares linear regression, support vector regression, k-nearest-neighbours, random forest and a simulation-based method [35]. The paper concludes that support vector regression and simulation perform equally well, with linear regression performing worse, but k-nearest-neighbours and random forest outperforming the others. See section 3.1 for a more detailed discussion on the paper.

A subsequent paper, released in 2016, again compares linear regression, support vector regression, random forest, and k-nearest-neighbours models but now a artificial neural network model is considered instead of the simulation-based model [36]. The comparison results in linear regression and random forest outperforming the other models. This is surprising, since in 2014 the linear regression model performed the worst. Sadly no discussion on the difference in results between both papers was found. In the discussion section below (3.2.2), this discrepancy will be further explored.

A third conference paper by Lee on taxi time predictions was published in 2019 which investigates specific periods of the taxi time: the pushback time, and ramp taxi time [37]. To predict these periods, Lee developed six models using linear regression, support vector regression, k-nearest-neighbours, random forest, artificial neural network, and a manually developed tree. From the analysis it is found that all of the models performed similarly and that predicting the specific taxi time components is difficult due to the large uncertainties contained within.

During the period of Lee's article, two other articles of note for this project were published. Firstly Herrema et al. published an article comparing linear regression, regression tree, Reinforcement Learning, and artificial neural network [27]. In the article the author also uses the RReliefF, and SequentialFs algorithm to select the best dependent variables from the ones available to him. Next, to compare the models, the author not only considers the performance, but also the computation time and data required. In Herrema's analysis it is concluded that the regression tree model provides the best results.

Lastly in 2018 another paper was published comparing linear regression, artificial neural network, and two modified support vector regression models by Lian et al. [38]. A unique addition to the models is the dependent variable delay. By including this variable, the models may be able to react to anomalous events by increasing the taxi time when aircraft have been taking off later than expected in the past hour. Finally the comparison concludes that the two modified support vector regression methods perform the best.

From this subsection it can be seen that a large number of taxi time prediction models have been created. By no means does this literature review contain every model described in literature. Notable other taxi time models include the 2017 cell transmission and fundamental diagram based model by Yang et al. [60], two articles and the phd thesis by Simaiakis et al. on analytical models [53, 55, 56], the 2010 analytical model by Robinson et al. [48], the phd thesis by Tu on genetic algorithms and ground movement control [58], the traffic and queueing rule-based model by Carr et al. [9], and lastly the 2000 article by Andersson et al. [1]. These models have not been considered for this project because they have not yet been evaluated in a way that allows for them to be compared to other models.

Finally two articles that compare data driven methodologies on different problems have also been considered. The first article is the technical report by Eurocontrol on predicting flight routes with deep neural networks [43]. This article is analysed for its succesfull use of artificial neural networks within the field of air traffic management. The article compares five different methodologies, and continues with artificial neural networks with the following argumentation:

> *Several machine learning methods have been evaluated, including Decision Trees, Random Forests, Kernel SVMs, K-Nearest Neighbours and Neural Networks. A random forest with adequate pruning offered the best results out of the box. However, after research and careful tuning, a deep neural network could surpass the results by a small margin. The decision to continue with a artificial neural network is mainly driven by:*
>
> - *Random forests do not scale as well if more training data or more predictors are used.*
> - *The serialised model is much smaller for a artificial neural network.*

- *Off-the-shelf libraries offer a high degree of customisability, e.g. to try custom cost functions or experiment with alternative topologies, and allow integration with existing application code.*

While this type of argumentation for a particular model is common place in the industry, some questions can be raised on the scientific aspects of the argumentation. First, the article indicates a small marginal improvement without any numbers to back up this claim. These improved results are also not mentioned in the three main decision drivers mentioned for artificial neural networks.

Instead the article claims that random forest performs worse when more predictors are used. However no sources for this claim are given and further literature review by this project could not back this up either. Furthermore, the remaining two arguments given for artificial neural networks, model size and available libraries, are not considered relevant for the taxi time prediction problem.

Another concern with the argumentation is the lack of clarity with respect to the methodology. For example it is unclear whether each methodology has been given the same development time. This could severely impact the results. Additionally, the article describes overfitting problems with the artificial neural network, but it is ambiguous as to when those problems were fixed. Hence, the tuning that made artificial neural networks perform better than random forest could have been due to the model overfitting instead.

Finally, the paper mentions twice that they have the goal of creating *"a path to gradually increase the role of machine learning"*. This indicates a bias towards new and prominent solutions, rather than the best one. Together with the other arguments, it is therefore believed that this article cannot be used as an argument in favor of any model.

Next, the 2014 article with over a thousand citations on scopus by Fernández-Delgado comparing 179 classifiers on 121 different data sets was reviewed [18]. The article concludes that the most likely family of classifiers to be the best is random forest. Additionally the article could not show that other families of classifiers performed better for increasing instances or dependent variables. The article does however focus on classification problems rather than regressions problems like taxi time prediction.

3.2.2. Discussion & Research Gap

As seen in the previous subsection, there is a plethora of literature concerning data-driven taxi time prediction models. Each model differs in its type and dependent variables, also called parameters or features. Additionally each model is evaluated with a different technique at a different airport. This causes inconsistencies in the results and conclusions between papers. This subsection discusses these difference and is further subdivided into four parts analyzing the different model types, dependent variables, test designs, and results.

Model Types

This section describes factors related to different model types such as the scalability, and variability of model types. Additionally the benefit of providing the taxi time prediction probability distribution and the research gap with respect to model types is discussed. Comparison of prediction performance of different model types is not found in this section. Instead this can be found in the section below called 'Results & Conclusions'.

Firstly, different model types have different scalability. Scalability can make the model more easy to implement in a wider variety of operating conditions and locations. Generally these models have the ability to predict taxi time by learning from data, rather than by the architecture of the model itself[3]. For these models to perform well it may however be important to include dependent variables specific to operating condition and location.

When limited to these types of models, the main types encountered in literature are: linear regression, support vector regression, regression tree, random forest, k-nearest-neighbours, and artificial neural network. It is however important to be aware that within each type large variations are found. Linear regression for example can refer to a model with a single or multiple relations between the input parameters and the output. Additionally Lee et al. use a least absolute shrinkage and selection operator linear regression model, a significantly different implementation than the standard linear regression.

[3] These models are often grouped together as Machine Learning. Due to the lack of a clear definition and boundary for machine learning models this term is not used in this project.

Lastly, while not required for taxi time prediction, the benefit of some models to provide a probability distribution of the taxi time should not be ignored. For example, an airline might be interested to know the probability of the aircraft to be at its gate by a certain time to optimize its operations.

With the already large number of model types applied to taxi time prediction, developing a new type of model is unlikely to have a large impact within the field. Instead it is believed that enhancing existing model types by adding a correction to the model based on average prediction error of recent predictions could have a larger impact.

This correction could aid taxi time predictions during periods where all aircraft have an increased taxi time. As will be discussed in section 3.4, these kind of periods are currently believed to cause the largest problem in terms of loss of efficiency and increased workload.

Lastly as discussed in the previous section, Futer et al. has already successfully applied such a correction to an older model in 2006 [19]. It is believed that the same approach could be taken to improve current state-of-the-art models.

Dependent Variables

Next to the type of model, nearly each article analyses and uses different dependent variables, also called (input) parameters, features, or attributes. These dependent variables include variables related to weather, traffic (ground vehicles and aircraft), airport configuration (active runway, taxiways, etc.), local time (hour, day, month, year, etc.), unimpeded taxi time, taxi distance, total turning degrees, and recent airport delay. In addition parameters specific to the aircraft in question are used such as its gate, assigned runway, SID, STAR, WTC, type, and airline.

One major reason for the different dependent variables, is the available data. For example, some articles such as the one by Chatterji et al. from 2013 [12], specifically focus on models which do not rely on a certain set of data. In case of the article by Chatterji et al. the models analyzed does not rely on the availability of airport surface detection equipment.

A second cause of variation in dependent variables are differences in the way the conditions are captured by a parameter. For example, one may quantify traffic based on the number of arrivals and departures, or by the number of aircraft taxiing to or from a particular runway instead.

Thirdly, many model types perform better when only faced with those variables useful for prediction. Benefits include easier visualization and understanding of the data, reducing the required measurements, storage, training times, and overfitting [22]. Hence for this reason it is important to understand how the dependent variables are chosen, a process called feature selection.

Especially early articles spend little effort on feature selection, due to model or data availability limiting the number of dependent variables. In these cases feature selection occurs in a subjective manner based on what is available, intuition, trial-and-error, or basic plots of the variable in relation to the taxi time.

Other articles calculate the significance or p-value, and coefficient of determination (R^2) to rank and determine the most important parameters. The p-value indicates the probability that no relation between the dependent and independent variable exists. R^2 indicates predictability of the independent variable by the dependent variable. This analysis can also be extended by analyzing the covariance matrix and the cross-correlation coefficients. The goal of this analysis is to determine coefficients that capture the same dynamics. Hence, when there is a large cross-correlation between two parameter, using only one can be beneficial. Lastly Herrema et al. (2018) uses two algorithmic feature selection techniques called ReliefF and SequentialFs [27].

Additionally, articles have been written to specifically determine the most valuable variables to determine taxi times. In 1999 Idris et al. published a detailed analysis of different potential variables that influence taxi time [29], a more elaborate and extensive analysis can also be found in Idris' PhD thesis [30]. More recently, Yin et al. (2019) proposed a set of 'taxi situation indices' through analysis of the correlation and significance of variables related to taxi time prediction and their cross-correlation [61].

Finally for a taxi time prediction model that can be used within the operation, care should be taken that the dependent variables are available or predictable in the time period prior to the start of taxiing. Hence, depending on the evaluated prediction horizons for the model, the selection of the dependent variables may be different.

Overall no gap has been found in literature with respect to forgotten variables. Instead, models have often had to select a subset of the possible variables due to the constraints mentioned above. This feature selection could however be done in a more organized manner as described in general data science literature [22]. Herrema et al. has already applied such techniques, but more could be tested [27]. Moreover, additional data and computing power could also reduce the need for feature selection. Lastly, no consideration of the prediction horizon for feature selection has been found in literature.

Test Design

Another way in which literature differs from one another is the method used to assess the performance of the model. The assessments differ both in experimental set-up as well as in the performance metrics used. The most common performance metrics are Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Standard Deviation (σ) and Coefficient of Determination (R^2). In addition to these fixed numerical performance metrics, the fraction of predictions with an error within a certain time window or a percentile of the absolute error are also frequently used. Other numerical performance metrics used are median error, mean error, mean absolute percentage error, median absolute difference, root relative squared error, relative absolute error. Lastly, a single study used the Mahalanobis distance metric for evaluation, while another study also considered computation time and the amount of data required.

Comparative studies, such as Ravizza et al. 2014 [47], tend to provide a combination of these numeric metrics for different model types. Different models subsequently perform relatively well for some metrics and worse for others. Finally, the best method is selected based on which model performs good for most of the metrics, without elaborating on what value each metric provides.

In addition to the numerical performance metrics, plots are regularly used to visually compare different models. The plots used in studies include histograms, boxplots, and cumulative frequency graphs.

In general there are two types of experimental set-ups: simulation, and human-in-the-loop (HIL) experiments. To evaluate taxi time prediction models it is important to consider different prediction horizons as some models may perform better or have specifically been designed for a certain prediction horizon. Specifically for simulation it is important to ensure that the model only uses data that is available at the simulated time of prediction.

Compared to simulation, HIL experiments allows for additional evaluation of the effects of the model on the whole airport operations. As will be discussed in section 3.3, the improved taxi time estimates can have an impact on a variety of airport stakeholders such as the air traffic controller, airline, ground handlers, pilots, and passengers through the A-CDM system.

From reviewing literature it was found that significant progress can be made by improving test design. Most articles, especially in recent times, do not consider a prediction horizon. Additionally, minimal research has been done on model performance during anomalous airport operations. Finally, the focus of many evaluations has been on reducing the average error, rather than variability. This is contrary to how the performance of a model is experienced in the working environment.

In a working environment predictions are made at different times prior to the start of taxiing. Additionally, from talking to air traffic controllers mainly larger taxi time prediction errors form a problem for the operation, especially during anomalous operations. Currently air traffic controllers manually adjust taxi time during such events causing extra workload.

Results & Conclusions

It is not relevant to discuss the research gap of results, since the missing results are caused by gaps in the analysed model types, dependent variables, and test design. This section therefore aims to summarize and determine the value of the results and conclusions found in literature.

The most significant problem with the results and conclusions presented in literature are the inconsistencies between them. A single author within the span of a year published two articles comparing the same four model types. But the authors conclude in the first article that linear regression performed the worse, while in the second linear regression performed best together with random forest [35, 36]. It is believed that these kind of discrepancies are caused by a variety of reasons which are described below. Additional information on the difficulties of comparing models can also be found in the 2014 article by Férnandez-Delgado et al. that compares classifiers [18] and the 2014 article by Macia et al. on mindful repository design [41].

Firstly it shows that the performance of different model types is similar. Other factors such as the manner of implementation of a model can therefore make a larger difference in performance. It is common for the main model type in an article to have been under development for years. Subsequently it is compared to other model types which have been implemented without the same development time. This creates a bias towards the models that authors have developed themselves.

Secondly discrepancies between results are created by differences in dependent variable, test designs, and applied airports. As already discussed, airports in Europe have different dynamics than those in North America where queuing is much more common. The relevance of comparing model types between articles where these aspect vary significantly is therefore low. As stated by Ravizza et al., a solution for this problem could be benchmark scenarios, but none have been developed yet [46]. Generalized results derived from a variety of airports are scarce.

When considering the development time bias and the importance of implementation, it is however still feasible to derive trends in articles which compare model types internally using the same dependent variables, test design, and airport. The following model types can be compared in this manner: linear regression, support vector regression, regression tree, random forest, and artificial neural network. Sadly insufficient comparisons have included k-nearest-neighbours models, hence this type could not be compared.

Analysing the models that can be compared, performance of linear regression models tends to vary from best to worst. This indicates that the performance is highly dependent on implementation and experimental setup. The method is however simple, and often used as a baseline to compare with other models.

Lian et al. concluded that their specific implementation of support vector regression performs best, however in general support vector regression performs badly [38]. The results from Lian et al. result could be explained by development time bias, since the onset of the research was to develop and determine the feasability of support vector regression.

Next, regression tree and the similar random forest perform in a remarkably consistent manner. It performs often as well or close to the best models compared. It is therefore a promising type. This corresponds to the findings by Férnandez-Delgado et al. that found that random forest classifiers are most likely to be the best option from a comparative study of 179 classifiers [18].

Lastly, it is necessary to discuss the artificial neural network model type. Neural networks are the default method to perform some of the hardest computer problems such as image classification and natural language processing. On these types of problems artificial neural networks perform orders of magnitude better than any other type of model and sometimes even beats human intelligence [13].

As discussed in the previous subsection, six articles have been analysed where artificial neural networks are used within air traffic management. Firstly Eurocontrol successfully used a deep neural network for trajectory prediction, but it is unclear whether an artificial neural network was the best option. Furthermore, the five studies that compared artificial neural networks for taxi time prediction with other models have all concluded that a different model type performs better [11, 27, 36–38].

It is often said that artificial neural networks tend to perform better with more data. However the model trained on 250,000 flights and 42 input parameters by Herrema et al. did not perform better. Reducing the number of input parameters to ten even reduced computation time by a factor of three and increased the robustness of the model while having a negligible effect on performance. Hence, large amount of data does not appear to favor artificial neural networks in this problem.

Additionally artificial neural networks have the drawback of being a black box. This means that it is hard to determine why the system made a certain prediction. While this may not be necessary, this may cause implementation difficulties within a conservative environment.

3.3. Cost-Benefit Analysis of Taxi Time Prediction

This section reviews the literature related to the costs and benefits of surface congestion management. Surface congestion management is the system that controls airport traffic to reduce congestion. Taxi time predictions are often an important part of this system. Similar to the other sections in this chapter, this section is divided in the subsections pertinent papers (3.3.1) and discussion & research gap (3.3.2).

3.3.1. Pertinent Papers

For the initial development of the A-CDM concept, Pina et al. performed a study in 2005 to determine the performance of A-CDM and variable taxi time calculations. Additionally, the paper analysed the qualitative benefits for the airlines, the airport, and ATC. The study was performed by running a shadow mode trail, where the existing and improved calculation methods were directly compared in an operational setting. From the trials it was found that airlines and the airport could effectively use the taxi time calculations to improve their resource allocation. Furthermore, ATC and the airlines praised the predictions for aiding the decision-making process and slot compliance.

Next in 2010, Simaiakis et al. published an article which attempts to determine the impact of congestion on taxi times, fuel burn, and emissions at major U.S. airports through data analysis [54]. If the unimpeded taxi-out times could be achieved, the article found that fuel burn and emissions are reduced by nearly 50% during the taxi process compared to normal operations at major U.S. airport in 2010.

In 2014 a subsequent peer reviewed article was published by Simaiakis et al. [57]. In this study the results of field tests are presented instead. From the field tests it was estimated that over a 37h period over several different days at Boston Logan International Airport between 12,250-14,500 kg of fuel was saved through pushback rate control. In the article no extrapolation was made to determine the fuel savings on a yearly basis or daily average for the airport. Additionally it was found that pushback rate control resulted in 3 minutes of unused runway capacity during the field test, increasing total delay. Hence, the paper suggest to increase queuing buffers at the runway and to consider the cost due to potential losses of runway capacity.

In 2010 Günter et al. presented the effect of reducing taxi time prediction error on the efficiency of pushback rate control [23], additionally an article was published on the topic in 2011 [21]. In the articles Günter et al. compares two different taxi time predictions at Frankfurt airport in combination with the A-CDM concept. The first model uses look-up tables for stand-runway combinations, like the current LVNL model at Schiphol airport. The second model takes the current traffic situation into account and calculates taxi times dynamically. The second model is a commercial model developed by ATRiCS which uses simulation. Sadly little information is given to the public on the inner workings of the model. Finally both methods implemented within the A-CDM concept are compared to a scenario without A-CDM or TSAT allocation.

The papers estimate that the fraction of flights with a taxi time prediction within a two-minute window rise from 42% to 66% by using dynamic taxi time calculations compared to the static table. These results have been derived through simulation of the airport surface. However since a simulation is used to evaluate the performance of another simulation (to calculate the dynamic taxi time) limitations present in both may provide overly optimistic results. Additionally the dynamic taxi time simulation is commercial software unavailable to the public and the methodology is only minimally explained.

Despite these limitations the research provides some valuable insight. Firstly the research concludes that improving taxi time results in lesser fuel burn, emissions, costs and delays due to capacity losses. The research also stresses the importance of strategic TSAT calculation from the taxi time estimate. If calculations are made such that aircraft are expected to directly roll onto the runway with zero buffer, unexpected delays create capacity losses causing an increase in costs. It is therefore important to balance the probability of losing capacity, with the benefits of gate holding.

To determine the optimal strategy and the financial benefit of different strategies careful selection of cost factors related to flight delay and fuel consumption is required. Günter et al. compare different strategies for different flight delay costs. From the comparison it is found that depending on the accuracy of the taxi time prediction, and the TSAT allocation strategy the potential benefits can vary significantly and even be negative compared to the scenario without TSAT allocation.

A study analyzing the potential benefit of surface congestion management on the entire U.S. air system was performed in 2013 by Nakahara et al. [44]. The study analyzes and combines the benefits of surface congestion management at each U.S. airport. A high fidelity model applied at two airports based on field trials is used to validate a second lower fidelity model applied at more airports, which is again used to validate an even lower fidelity model. This lowest fidelity model is subsequently applied to the top 35 U.S. airports to calculate the aggregate benefit for the U.S. air system. These calculations estimate that between 2.2 and 3.9 billion gallons of fuel could be saved across the 35 top U.S. airports, which would correspond to a reduction in costs of 5.5 to 9.5 billion USD.

Next in 2014 Yoon et al. presented a paper on the benefits of a surface congestion management approach developed by NASA, called Spot And Runway Departure Advisor (SARDA), at Dallas/Fort Worth airport [31]. SARDA is analysed through a human in the loop experiment in a simulator. It found that a reduction of 45% in taxi delay in a medium traffic scenario, and a 65% reduction in taxi delay in a heavy traffic scenario is possible. This corresponds to a reduction in fuel and emissions of 23% and 33% respectively. Additionally it was found that runway capacity was unaffected, and gate-holds were generally less than 15 minutes. Finally, the simulation does not take into account pushback uncertainties. According to Hayashi et al. this could reduce the real world benefits of the tool [26].

Additionally in the 2015 conference paper by Hayashi et al. an evaluation of a different variation of SARDA is made. The variation of SARDA was developed to analyse the possibility for a hub airline who operates a significant portion of the flights at an airport to implement surface congestion management by itself, without communication with ATC. The study found that American Airlines at Charlotte Douglas International Airport could reduce its fuel for departure with 10-12% by applying this variation of SARDA. Additionally it was found that controller workload is reduced, and further reduction of queue times are possible.

Finally there is the 2017 study by Hao et al. who calculated the fraction of fuel for a whole flight that could be saved by a large U.S.-based network carrier through surface congestion management [25]. The study concludes with statistical analysis of airline data that 1% of the fuel consumed by the study airline could be reduced by eliminating taxi-out delay through gate-holding strategies.

Additionally it is found that the potential fuel savings per airport vary significantly. For some airports fuel consumption by outbound flights could be reduced by as much as 2% compared to the average 1%.

Next, the research found that the fuel consumption from a minute of taxi-out delay is less than the impact of a minute of unimpeded taxi time. The difference can be as much as 50%, hence it is inaccurate to assume equal fuel consumption rate for taxi delay and unimpeded taxiing.

Lastly it is hard to know how the results of this study could be generalized. Namely the results are based on a single U.S. airline, hence the results are affected by the visited airports and operating procedures of the airline. Furthermore, in the study taxi delay is totally eliminated. This is not a realistic scenario as this would lead to capacity losses and extra total delay.

3.3.2. Discussion & Research Gap

In this subsection the pertinent papers concerning the cost-benefit analysis of surface congestion management and taxi time prediction will be discussed and research gaps will be identified. This subsection has been further divided into four parts: potential costs & benefits, the method of analysis, the variation of results, and research gaps.

Potential Costs & Benefits

Within the cost-benefit analyses a variety of costs and benefits have been identified. Firstly, the main benefit of surface congestion management is reduced fuel consumption and cost by shutting down the main engines at the gate rather than idle during taxi. The reduction in fuel consumption additionally reduces the environmental impact of a flight. Next, by increasing the time at the gate passenger & bag connection rates could be improved. Finally by controlling surface traffic taxi time variation can be reduced. Combined with improved taxi time calculation methods this reduced uncertainty allows for more efficient resource allocation.

The costs of surface congestion management is often ignored. These costs include the potential for reduced capacity due to a smaller queue which adds flight delay. Additionally it is not always possible to perform optimal surface congestion management due to gate availability. Finally the fuel costs at the gate generated by the auxiliary power unit should not be ignored.

In addition literature identifies two aspects that could be positively or negatively impacted by surface congestion management: controller workload and slot compliance.

Method of Analysis

The method employed to determine the costs and benefits of surface congestion management vary wildly between research papers. Firstly in the research paper by Simaiakis et al. a field trial with and without surface congestion management was used [57]. This method however requires significant testing time and resources to negate differences between the tested scenario's. Additionally the scenario's should ideally be representative of the different circumstances encountered throughout operations.

To compare identical scenarios a human in the loop simulation with tower controllers and pseudo pilots could be used instead. Such studies have been performed by Hayashi et al. and Jung et al. [26, 31]. The performance of this type of analysis does however highly depend on the fidelity of the simulator.

Next simulations can also be run without human involvement. This requires significantly more modelling potentially introducing additional errors. It is therefore necessary to perform extensive validation to ensure the model accuracy. For example, Günter et al. uses a simulator to SequentialFs the performance of another simulation [21, 23]. Since both simulators may contain similar errors performance may be overly optimistic reducing the validity of the results.

Lastly, the results of several papers are based on data analysis. In these analyses an assumption is made for the effect of surface congestion management. Next, based on current operations the data analysis determines the potential costs and benefits with the assumed effect.

For example, the potential fuel savings for zero taxi delay are calculated by Hao et al. [25]. The assumption that surface congestion management eliminates taxi delay is however overly optimistic. Surface congestion management cannot eliminate taxi delay from aircraft interacting on the taxiway. Hence, the validity of models based on data analysis is limited.

Variety of Results

This section explores how and why the results between different research papers vary significantly. More specifically this section will look at the different types of results and the limitations of results including generalization problems.

Similar to the method of analysis, the types of results vary significantly from qualitative results to different quantitative results. Firstly research may be limited to qualitative assessments to determining whether there are improvements or not. For example determine whether fuel is saved or the controller workload is increased. More quantitative analysis may determine the amount of fuel saved, or the stress rating of the controllers. Together with stakeholders these results can further be converted into the financial cost and benefits for each stakeholder. This could for example include the financial benefit of lower emissions on the public.

Most research however solely focus on the aspect of fuel cost savings. A notable exception is the research by Pina et al. who includes the qualitative benefits for controllers and airlines [45]. Additionally research by NASA has an extensive quantitative analysis of controller workload [26, 31]. Finally Günther et al. not only considers the fuel savings, but also the cost of added total delay due to capacity losses [21, 23].

Next, the value of the results of several papers are limited due to the assumptions made. Firstly potential fuel burn from the auxiliary power unit at the gate due to surface congestion management is often ignored. Additionally it is often assumed that the fuel burn during delayed taxiing is equal to normal, unimpeded, taxiing. However, Hao et al. found a difference of 50% between the impact of a minute of taxi-out delay versus a minute of unimpeded taxi time [25].

Additionally most analyses ignore gate availability problems. For surface congestion management, aircraft need a place to wait. This is mostly done at the gate, but gates are a limited resource especially at busy airports. While an aircraft could wait at an alternative parking spot, this requires a pusher and or engine power adding to the costs.

Finally there are limitations to the generalization of different research. The results compare different implementations of surface congestion management at a specific airport during a particular scenario. This makes

Table 3.1: Eurocontrol A-CDM taxi time estimation required accuracy

Timelines	Time periode covered	Input	Required Accuracy
Long Term	Off-Block Time -3h to -2h	Predicted static data (current runway use, and planned stand). If this information is not available then a default should be used	+/- 7 minutes
Medium Term	Off-Block Time -2h to -30 min.	Update static data (current runway in use and planned stand)	+/- 5 minutes
Short Term	Off-Block Time -30 min. to Actual Off-Block Time	Current runway in use and actual stand	+/- 2 minutes

the results highly dependent on the airport itself, the scenario's analysed, and the specific implementation of surface congestion management.

This is exemplified in the research by Hao et al. [25] where it was found that within the top 35 U.S. airports the potential savings of surface congestion management vary by a factor of two. For surface congestion management to be effective, there has to be congestion in the first place. Additionally, many European airports already have implemented a basic surface congestion management strategy, hence the potential gains are reduced.

In addition the effectiveness of surface congestion management depends on its implementation. Several papers assume surface congestion can remove all taxi delay, equalizing the taxi time to the unimpeded taxi time. This is overly optimistic as surface congestion management can only eliminate queue delay, not the delay from interaction between different vehicles on the airport surface.

Lastly, this project is specifically interested in the potential surface congestion management with improved taxi time prediction. This has only been analyzed by Günter et al. [21, 23]. The research found that improved taxi time prediction can significantly increase the effectiveness of surface congestion management.

Research Gaps

The primary missing knowledge in cost-benefit analyses of surface congestion management and taxi time prediction is an integrated and representative analysis involving all stakeholders. The study should aim to calculate the financial costs and benefits of fuel savings, emissions savings, controller workload, passenger & bag connectivity, improved resource allocation and slot compliance within the gate availability constraints for the public, passengers, airlines, airports and air navigation service providers. This type of calculation would require the cooperation of each stakeholder to truthfully share its cost factors.

Next, for the study to be representative it is necessary that the method of analysis is thoroughly validated preferably through field trials or human in the loop simulation. The generalisation of this type of research to different airports can only be done when the research is applied to multiple airports and patterns emerge. Additionally research could be done on different surface congestion management strategies.

3.4. Industry Goals

The need for improving operational efficiency and accurate taxi time predictions has been identified by a variety of stakeholders. This section will highlight some of the main goals set by different stakeholders with respect to this topic.

Firstly, in Eurocontrol's A-CDM implementation manual table 3.1 is given with the required taxi time estimation accuracies [17]. While it states that these accuracies are required, within current implementations such accuracies are only achieved for around 60% of the flights. It should therefore be seen as a goal for this project to achieve these goals more consistently at Schiphol airport. Additionally Eurocontrol states that future systems should calculate taxi times 'dynamically' using 'sophisticated data sources' such as ground movement surveillance equipment.

Besides Eurocontrol, also the European Commision has set a similar goal of predicting taxi times using surface traffic monitoring through the pilot common project:

> *Advanced Surface Movement Guidance and Control Systems (A-SMGCS) shall provide optimised taxi-time and improve predictability of take-off times by monitoring of real surface traffic and by considering updated taxi times in departure management [14]*

Finally LVNL has set goals and requirements for the taxi time prediction for the new outbound planner [40]. One of these requirements is that the taxi time tables shall be in minutes. This requirement has been set because the other systems such as the DSP uses this resolution and it is therefore easier to implement. While this is a Schiphol specific requirement, it gives an indication that improving taxi times with seconds might have an immeasurable effect on the actual operations. This means that for operational improvements one should focus on reducing the large deviations from the estimates. A similar conclusion was found when talking to air traffic controllers who indicated that inaccurate taxi times only become a problem above a minute.

Additionally, through conversations with LVNL air traffic controllers, it was found that the main issues with the current taxi time predictions occur during periods where the actual taxi times of all aircraft are increased. These increases are caused by a variety of reasons such as maintenance, construction, lightning or emergencies. During periods with increased taxi times, air traffic controllers currently need to manually adjust the planning. This leads to an increase in workload and efficiency losses. While LVNL already has plans to tackle this problem by enabling air traffic controllers to increase taxi times with a fixed percentage, this project should look at ways to improve taxi time prediction performance during these periods.

In conclusion, it is found that there is significant interest in improving taxi time prediction for surface congestion management within the industry. A focus should be put on improving prediction during periods with increased taxi times and a variety of lookahead times should be analysed.

4 Research Objectives & Questions

Every good research project needs a set of objectives to work towards. Additionally to achieve these objects it is necessary to identify a set of research questions for which the answers will help the project to achieve its objectives. This chapter first describes the main- and sub-objectives of this research project in section 4.1. Subsequently in section 4.2 the research questions are discussed.

4.1. Research Objectives

As stated in the introduction, the main research objective of this project is the following:

> *The objective of this research is to maximally improve taxi-time predictions at airports by analyzing and comparing the operational performance of different prediction methodologies with different input parameters at different prediction horizons.*

To understand the full implications of this objective, it is important to break it down into several elements. Firstly, the objective does not refer to a specific airport, hence it is implied that the research should be generalizable to every airport.

Additionally, operational performance signifies a focus on performance within the larger operational context. Furthermore operational predictions are made at different points in time. The objective therefore includes performance at different prediction horizons. Moreover the performance of the prediction model is expected to drift as the training data becomes less relevant to the current situation. Hence this drift should also be evaluated. Additionally, as discussed in section 3.4, for operational performance improvement one should focus on reducing the large prediction errors and performance during anomalous operations.

Next, it should be noted that not every methodology can be analyzed. It is therefore the objective to identify the most promising methodologies through literature and subsequently only apply and analyse those methods.

Finally the use of different input parameters implies the use of feature selection techniques to determine the optimal input parameters for each model technique and prediction horizon. Furthermore, the optimal amount historical training data should be identified.

To achieve the main-objective, additional sub-objectives have been identified. These sub-objectives are based on the CRISP-DM framework [10]. The CRISP-DM framework provides a general set of phases, tasks, objectives, and deliverables for general data mining projects. A diagram of the phases and objectives of the standard CRISP-DM framework can be found in figure 4.1. In the figure, the color red indicates the elements which have not been considered in this project.

Compared to the standard CRISP-DM framework, this project has replaced the business understanding phase with a literature study as the project is focused on scientific research rather than a product. Similarly the deployment phase is not considered in this project. Furthermore with the goal of comparing different models, multiple different models are build and compared. The crisp-dm assess model sub-objective has therefore been extended into a whole phase called results where the models are assessed and compared. Additionally the evaluation phase has been renamed to the discussion to adhere to the convention used at the university.

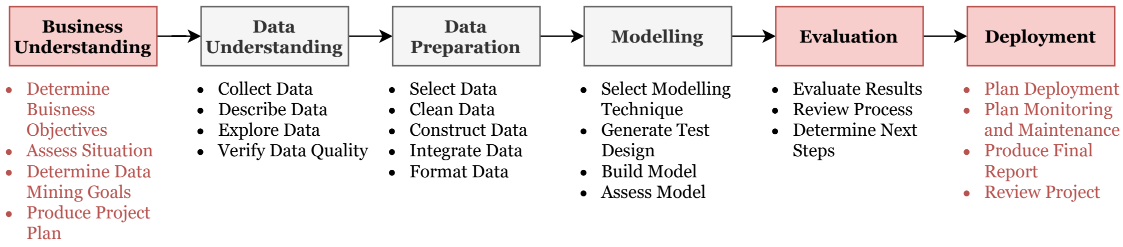

Figure 4.1: Diagram of the standard CRISP-DM phases and objectives

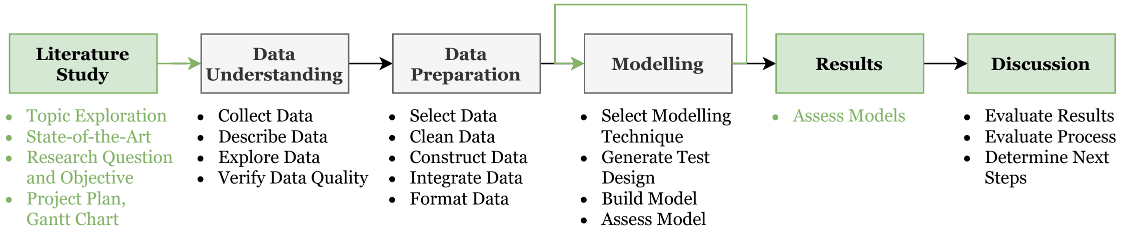

Figure 4.2: Diagram of the adjusted CRISP-DM phases and objectives

Finally, a diagram of the adjusted CRISP-DM phases and objectives for this project can be seen in figure 4.2. The color green indicates the elements which are not present in the standard CRISP-DM framework.

4.2. Research Questions

From the main- and sub-objectives discussed in the previous section, a set of research questions have been identified and listed below:

1. What is the current state-of-the-art in taxi time prediction?
    (a) Which types of taxi time prediction models are used?
    (b) Which dependent variables are used for taxi time prediction?
    (c) What are the issues taxi time prediction models run into?
    (d) Which potential solutions have not yet been researched?
2. What are the potential benefits of surface congestion management at Schiphol airport compared to current operations?
3. Which type of model, selection of data, and type of performance feedback should be used for taxi time prediction at Schiphol airport?
4. What taxi time prediction performance can be achieved at Schiphol Airport for different prediction horizons (Actual Startup Request Time, -3 hours, -2 hours, -30 minutes)?
5. Can a taxi time prediction model be used to predict the time of different elements within the taxi process such as pushback, ramp taxiing, or queuing?

5 Methodology

In this chapter the methods to achieve the data understanding, data preparation and modelling phases are described. The sections in this chapter corresponds to each phase respectively.

5.1. Data Understanding

The first step in understanding the data is to collect the data. For this step the literature study, expert consultation, and brainstorming is used to identify all potential input parameters that can be used for taxi time prediction. Subsequently the airports air navigation service provider, in this case LVNL, is requested to provide all available data on these input parameters. This process resulted in four major data sets: tower flight data, ground surveillance data, weather data, and runway configuration and capacity predictions. The origin, content, use, and issues of each of these data sets is described below. Additional minor data sets are also briefly described in the final paragraphs.

The first major data set contains all records of a flight available to the tower. Each time information about the flight is updated, a new record is made in the data set. It is therefore possible to reconstruct the flight information available to the tower at any given time. The records contain the aircraft identifier and registration, aircraft type, in- and outbound airport, departure or arrival gate number, take-off or landing runway, a variety of estimated or target or actual time of reaching a milestone (such as arrival/departure or in/off-block), CDM milestone status and more.

The main use for the tower data are the flight specific parameters directly impacting the taxi time such as gate number, take-off or landing runway, etc. The flight identifiers such as flight id, and registration are used for cleaning and linking the various data sources. Additionally the data of all the flights together can be used for traffic predictions. With the exact information available at a given time, it is easy to use with a prediction horizon.

The major issues with the data are the lack of a unique identifiers, cancelled flights, and manual overwrite. As flight id and registration repeat and no other flight identifier is present, it can be difficult to link the records for a single flight together. Next, the inclusion of cancelled flights in the data is valuable for the project, but no information is included on the cancellation time. It is therefore impossible to reconstruct when the tower became aware of the cancellation using the data set. Finally parts of the data can be manually overwritten causing inconsistencies.

The next major data set is the aircraft positioning data from the airports multilateration ground surveillance system, also called the ASTRA data set. It contains the location of all vehicles with active automatic dependent surveillance–broadcast systems. The position data is accompanied with ground speed, heading, aircraft identifier, and other parameters.

The positioning data can be used to calculate the taxi time of each flight. Additionally the runway configuration, taxi routes, and queue size can be derived from the data. The inclusion of all airport vehicles equipped with active automatic dependent surveillance–broadcast systems also allows for the determination of aircraft and other vehicle traffic.

Care should be taken when using the data for the presence of jitter, losses of signal, and incomplete vehicle identification. Especially in the presence of buildings near the ramp, the location information of vehicle may be temporarily lost or jitter around a given position. Lastly, incomplete identification data makes it difficult to match the vehicle track with a flight or to categorize the type of traffic.

The succeeding data set is the collection of all weather forecast provided to LVNL by the Royal Netherlands Meteorological Institute. The data contains near-term weather forecasts for the next eight hours which are given every three hours with an hourly resolution. Additionally long term forecasts for the upcoming thirty hours are produced every six hours with a three hour resolution. Each forecast contains the visibility-, wind-, temperatures-, precipitation parameters and additional comments.

The weather data will be used for taxi time prediction. It is expected that especially the visibility could have a major effect on the taxi times. Selection of the exact parameters used from this data set is further described in the next section.

The quality of the weather data is very high. The records are produced according to a strict update schedule and contain well defined fields.

The fourth major data set contains each "capacity forecast Schiphol". These forecast are created four times a day by an assembly of Schiphol airport, LVNL, and airline representatives. They are released at around two and eight o'clock in the morning and at one and seven o'clock in the afternoon. Each forecast contains the expected runway configuration and the corresponding capacity as well as a potential alternative configuration.

The runway configuration and capacity data set will be used as input for the taxi time prediction. From the literature study and expert opinion it is clear that runway configuration is one of the major drivers of taxi times. Compared to other airports, the runway configuration is believed to be even more important at Schiphol due to its complicated layout as discussed in section 2.3.1. While weather, specifically wind, is generally the most important parameter for the runway configuration, the specific environmental regulations and traffic demand affect the applied configuration significantly.

The data set is however rather inaccurate. Depending on the actual traffic the configuration transition times vary significantly from the predictions as well as the predicted configuration. Despite these limitations it is believed that this data set is the best available predictor for the runway configuration.

In addition to the four major data sets, several smaller data sets have been used by the project. This includes maintenance documents of the airports. These will be used to determine the ability for the model to adapt to construction. Since each construction project is different, manual analyses of these documents is needed and cannot easily be used in the model, more on that in section 5.2. Additionally several airport geometry data sets have been given. These can be used to determine gate coordinates, runway locations, taxiway paths, etc.

Finally, LVNL has collected a large amount of data over the years. This section has already made a selection of the best data set for each input parameter. For example, there exist other data sets containing the runway configuration, however these do not contain any predictions and are of lesser quality. The main challenge with the data will be to link them together as the identifiers within data sets repeat, or differ between data sets. In addition data is not always complete, and irregular manual inputs are present. The process to deal with these challenges is discussed in the next section.

5.2. Data Preparation

In this section the methodology used to create a data set containing the taxi time and potential dependent variables for as many flights in 2018 and 2019 as possible is discussed. Since the models generated in the next phase have different model techniques and prediction horizons, the data set contains all dependent variables that may be used by those models. Further selection of the dependent variables is therefore done during the modelling phase.

This section is subdivided into five subsections. Each subsection corresponds to the following sub-objectives respectively: select data, clean data, construct data, integrate data, and format data.

5.2.1. Select Data

With the large quantity of data available it is necessary to select only the most relevant data to reduce the required computational resources and avoid overfitting problems. In literature data selection is often not performed due to the low number dependent variables available. Others rely on intuition to determine the best parameters.

More advanced dependent variable selection techniques use significance and correlation tests between the variable and the output. While this method is more objective, it does not consider cross correlation between the different input parameters. Meaning that two significant parameters which are highly correlated to the output, but contain the same underlying information may be chosen over a less significant or correlated parameter that does provide new information.

For this reason several studies additionally use visualisations of the cross correlation between the different input parameters and output parameter. Subsequently based on these visualisations a set of independent input variables are derived. The limitation of this data selection methods is the need for visual judgement of the plots.

To avoid requiring significant subjective judgement, this project has chosen to pursue dependent variable selection through a genetic algorithm. This method is believed to be new within the field of taxi time prediction. The method allows for automatic variable selection and is able to find the best variables for a specific model. Since this project develops and compares multiple models the main dependent variable selection therefore takes place within the model specific phase, more specifically during model building (section 5.3.3).

The data selection during the preparation phase is therefore limited to subjective elimination of variables which are not believed to be predictive of taxi time. Additionally of those variables which attempt to predict a certain physical state, only the most accurate predictors available at the different prediction horizons are kept. For example, the most recent weather forecast at the different prediction horizons for the flight are kept. Or, flight plan data is used over passenger numbers for traffic predictions. Similarly as for data understanding, this process uses the literature review, expert consultation, and brainstorming to determine these variables.

5.2.2. Clean Data

The data cleaning process in this project consists of three elements: identifying missing data, identify and ignore general aviation taxi times, and remove outliers. For the taxi time prediction model to be used in operations, it has to be capable of performing with missing data. Hence missing data should not be removed. Instead it is identified and used to determine the performance of the model within the operational context where missing data is also present.

Next, general aviation and non-commercial flights tend have different and unique operating procedures for a flight. Additionally these flights are significantly less common. The taxi times of these flights are therefore hard to predict and less useful to the whole operations. For this reason the project decided to ignore the taxi time of these flights. However, dependent variable selection it will be determined whether data from these flight can contribute to the taxi time prediction of the other flights.

The final step of the cleaning process is taxi time outlier detection. For this project outliers are identified using the z-score. The z-score is calculated using equation 5.1 with x the variable, μ the mean of the variable, and $sigma$ the standard deviation of the variable. Once identified the project can assess the performance of the model with and without taxi times with a z-score above a certain value.

$$z = \frac{x-\mu}{\sigma} \tag{5.1}$$

5.2.3. Construct Data

Several dependent variables and labels are not directly available from a data source and therefore have to be constructed. Firstly only the whole taxi time is present in the tower data. The taxi time decomposed in pushback, apron taxiing, taxiway, and runway time have to be constructed from the aircraft position data. Secondly no data source is found containing the exact traffic demand (number of planned take-offs and landings), instead the tower data has to be used to determine the number of flights with an estimated time of departure or estimated time of arrival in a certain time interval.

To calculate the decomposed taxi times, a set of polygons based on the geometry data set were defined for the gate areas, aprons, taxiways and runways. Subsequently the positioning data is used to detect the transition times of each flight between these different polygons. For speed and efficiency these calculations have been performed within the SQL database containing the positioning data.

The number of take-offs and landings for a given period is predicted using estimated take-off and arrival times for flights from the tower data. Tower data may however not be entirely accurate due to late filings of flight plans. Additionally the data does not contain information on when the tower became aware of a cancelled or refiled flight.

For this calculation it is therefore assumed that when the last entry of a flight does not contain an actual time of departure or arrival the flight was cancelled or refiled. The cancellation time is equal to the time of the last entry. Since the time of cancellation may be later than the last entry, some hindsight might be introduced into the prediction. However no other method is found to handle these cancellations. Despite these limitations, this method of quantifying traffic is believed to be the best available option.

5.2.4. Integrate Data

To generate the model data different data sources have to be integrated. For each flight and time of prediction the dependent variables are derived and combined from the corresponding data set. For this process first all relevant flights are selected from the tower data. Subsequently the taxi time data of these flights are extracted from the aircraft positioning data set using flight id and time of arrival or departure and added to the flight data. Next, the best weather forecast at the estimated time of arrival or departure for the different prediction times are appended to each flight. Finally the best available capacity forecast for the flights estimated time of arrival or departure is also integrated with each flight record.

5.2.5. Format Data

With the dependent variables and labels for each prediction integrated, the data set has to be modified to fit the model and perform evaluation. Firstly certain dependent variables contain categorical data. Most models require this type of data to be one-hot encoded. This encoding decomposes the categorical dependent variable into multiple variables corresponding to each category. Subsequently each of the new dependent variables is set to zero except for the one corresponding to the category of the record which is set to one.

Next, the dependent variable estimated time of departure or arrival contains important cyclical information that could improve model prediction. For example, the taxi time might be influenced by the day of the year or hour of the day. This information is extracted using cyclical encoding. For example, the time of day can be represented on a unit circle with a radial where each pass of the radial around the circle corresponds to one day. Next this radial is decomposed in its sine and cosine elements, providing two dependent variables. This method ensures that the dependent variables of the time of day gradually changes over a change of date compared to a representation using 0 to 24 hours. Similarly the day of the year can be encoded.

Thirdly the data set is divided into training, validation, and test data sets. The test data set consists of 15% of the data set and is used to assess the unbiased performance of the model by not using the data until the model is completed. Since taxi operations are expected to develop over time, the training data set contains the chronologically 15% last flights of the whole data set. In this way the degradation of the model over time due to aging training data can be captured.

The second data set, the validation data set, again contains 15% of the data. The data from this set is randomly selected from the remaining data. This data set is used to calibrate the model's hyperparameters, the parameters which are not determined by the training data set.

Finally the remaining 70% of the data consists of the training data. As the name suggest training data is used to train the model itself.

To avoid dependent variables being given larger importance during initial training due to having a larger range of values, dependent variables are normalized [24]. The training data is used to create a transformation such that the range of the different inputs is equalized. This results in improved convergence rate of models

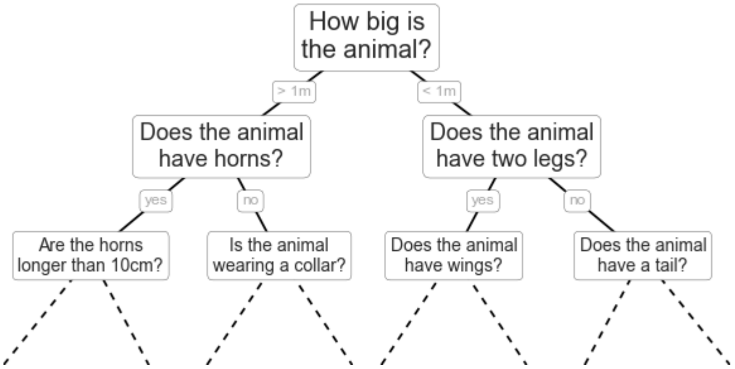
How big is the animal?
> 1m
< 1m
Does the animal have horns?
Does the animal have two legs?
yes
no
yes
no
Are the horns longer than 10cm?
Is the animal wearing a collar?
Does the animal have wings?
Does the animal have a tail?

Figure 5.1: Example of a biological decision tree [59].

which compare dependent variables (tree based models will not benefit as only one parameter is considered at a time). For this project specifically z-score normalization, also called standardization, is used.

To apply standardization the same equation as in section 5.2.2, see equation 5.1, is applied to all dependent variables of the training data. Note, the mean and standard deviation of the dependent variables in the training data is stored and used to normalize the validation and test data to ensure the same transformation is applied.

5.3. Modelling

In the modelling phase, the different models required to answer the research questions and achieve the research goal are created. This section outlines the methodology of creating these models in the following three subsections: select model type (5.3.1), generate test design (5.3.2), build model (5.3.3).

5.3.1. Select Modelling Type

With dozens of potential model types for taxi time prediction, this project uses the literature study to identify the three model types best suited for this project. Subsequently these three model types will be implemented and evaluated. As discussed in section 3.2.2 there are six common model types in literature: linear regression, support vector regression, regression tree, random forest, k-nearest neighbour, and neural network. The difficulties to compare model types, and the benefits and downsides of each model have already been discussed in the literature study. This section therefore focuses on the motivation for the selection of the following three model types: linear regression, random forest, and neural network.

Firstly linear regression has been chosen for its simplicity, adequate performance, and use as a reference. Nearly every comparative study starts its analysis with linear regression. It is one of the oldest, best understood methods and is easy to implement. Performance of linear regression in comparative studies inconsistently varies from best to worst. Due to its simplicity yet potential for high performance it is a great choice for this study. Additionally due to high use of linear regression the performance of linear regression within this project can be compared to a large number of other studies.

The project will specifically use the ordinary least squares multiple linear regression method. This method generates linear relations between the different inputs and the output.

Next random forest has been chosen for its consistent performance near the top in comparative studies even outside the taxi time prediction context. The model type is also well known with many available tools for implementation reducing the work required for implementation.

The random forest method creates a set of independent decision trees. These decision trees are automatically generated by the algorithm and resemble biological determination trees like the one given in figure 5.1. These trees tend to overfit the data, but by combining multiple overfitting estimators this effect is negated [59].

The third model type implemented for this project is the neural network. While neural networks have not performed best in predicting taxi times, neural networks have shown remarkable results on other problems.

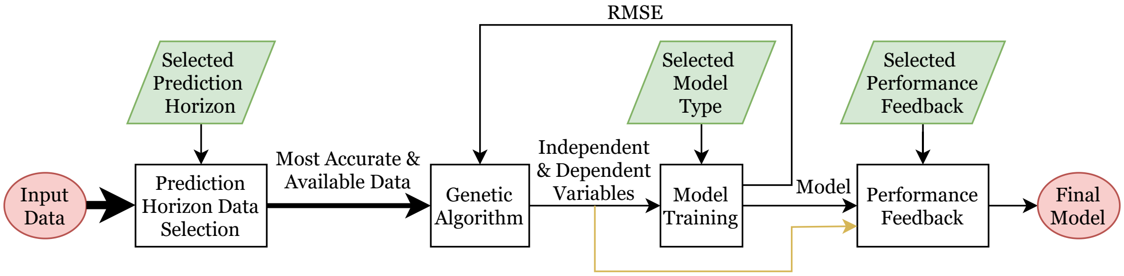
RMSE
Selected Prediction Horizon
Selected Model Type
Selected Performance Feedback
Input Data
Prediction Horizon Data Selection
Most Accurate & Available Data
Genetic Algorithm
Independent & Dependent Variables
Model Training
Model
Performance Feedback
Final Model

Figure 5.2: Diagram of the model building process.

This project will specifically analyse the feed forward multilayer perceptron neural network with back propagation. This is the most standard implementation of a neural network consisting of one input layer, at least one hidden layer, and an output layer [20].

5.3.2. Generate Test Design

With the research questions and objectives defined in chapter 4, the test design is tailored to answer and complete the research questions and objectives as accurately and complete as possible. As the objective of this project does not involve determining the effects of improved taxi time prediction on the operations, a HIL experiment is unnecessary and the problem is scoped to a data mining project. The test design instead focuses on comparing the performance of different models using a set of metrics.

To achieve this goal a large number of models have to be tested. Each models varies in the model type (linear regression, random forest, or neural network), prediction horizon (Off-Block Time -30 minutes, -2 hours, -3 hours), and performance feedback type (none, simple, or advanced) used. The tree model types have been derived in the previous section. Next, the three prediction horizons are chosen based on Eurocontrols performance goals discussed in section 3.4. More information on the three types of performance feedback are provided in the next section.

During the training and optimization of the models only the RMSE metric is used. This optimization includes the selection of dependent variables, and the amount of historical data. RMSE is the default metric used in machine learning. RMSE punishes large errors more than small errors compared to a metric like mean absolute error. It therefore corresponds better with the actual operational performance which is mainly determined by the large errors.

For the comparison of the different models, additional metrics to the RMSE are used. These metrics include the mean absolute error, the standard deviation, and the 95th percentile of the errors. These metrics are easier to interpret and allow better human intuition. Next, to assess the performance of the models during anomalous operations, data from operations during the corona pandemic will be used to assess the ability of models to adapt to new situations. Lastly, to analyse the ability of the models to determine the decomposed taxi times, the same analysis is done for each taxiing phase.

Finally the dependent variables of each optimized model is logged to determine the driving variables. Additionally to determine the model drift a plot is made of the errors over time since the last entry in the test and validation data. Similarly, the optimal amount of historical data is analysed by plotting the error for varying amounts of historical data.

5.3.3. Build Model

With the combinations of model type, prediction horizon, and type of performance feedback provided by the test design, it is possible to build a model. The model building process is visualized in figure 5.2. The parallelograms in the figure represent the inputs provided by the test design. Next, the rectangles represent the four major elements of the building process. The remainder of this section is subdivided along these elements and describe each in more detail.

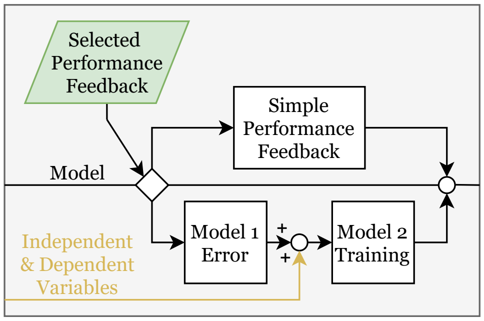
Selected Performance Feedback
Simple Performance Feedback
Model
Model 1 Error
+
+
Model 2 Training
Independent & Dependent Variables

Figure 5.3: Diagram of the performance feedback process

Prediction Horizon Data Selection

With the prediction horizon set, the selection process ensures that only the data available at the time of prediction is fed to the model. Additionally only the most accurate form of each parameter is selected, meaning the most recent prediction or the most accurate data source.

Genetic Algorithm

After the prediction horizon data selection, the genetic algorithm is initialized to further narrow down the dependent variables. The algorithm starts by generating a set (population) of randomly generated data selections (individuals) called the first generation. Subsequently the performance (fitness) of each individual in the generation is assessed by feeding the dependent variables to the model training step (see next section). Next the best performing individuals of the first generation are combined and randomly changed (mutated) to form new generation.

This process is repeated for multiple generations until a certain number of generations is reached or the performance stops improving. Finally the best individual of all generations is considered the best input parameter selection. More information on genetic algorithms can be found in the introductory book by Mitchell [42].

Model Training

To implement each type of model, Python is used in combination with scikit-learn [1] [2]. Python and scikit-learn have been chosen for its prevalent use in literature and familiarity to the researchers of this project. Scikit-learn does not have GPU support for the training of neural networks, hence when computation times become too high the project uses PyTorch and skorch which enable GPU accelerated training [3] [4].

When training the models the training data is initially kept small to improve computation times. When the dependent variable selection is optimized, the amount of training data is increased to find its optimum.

Performance Feedback

As mentioned in the research objectives and questions, see chapter 4, operational performance of a taxi time prediction is highly dependent on its ability to react to anomalous operations. Since these situations may not be encountered in training data, models may not have the correct reaction to the situation. By providing live performance feedback to models, models may be able to react better to such situations. To test this hypothesis each model will be evaluated with three different versions of performance feedback: none, simple, and advanced. A diagram of the performance feedback process, showing the mechanics and differences of each type of performance feedback, can be seen in figure 5.3.

[1] https://www.python.org/ Accessed: 11/09/2020
[2] https://scikit-learn.org/ Accessed: 11/09/2020
[3] https://github.com/skorch-dev/skorch Accessed: 11/09/2020
[4] https://pytorch.org/ Accessed: 11/09/2020

The simple version of performance feedback is inspired by the 2006 paper by Futer et al. [19]. In this paper predictions are improved by adding the average prediction error of the past hour to new predictions. This resulted in significant reduction in the large prediction errors. For the simple version of performance feedback the same system is implemented where a fraction of the average error of a certain period prior to the prediction is added to the prediction when it exceeds a threshold. The period over which the average is taken, the fraction of the error, and the threshold are manually optimized to reduce RMSE.

The advanced version of the performance feedback is similar. But instead of applying the correction with a manually set correction factor and threshold, the model is retrained with the average error over a set period prior to the time of prediction of the original model as an added dependent variable. Multiple iterations are required to find the optimal period from which the errors are taken. The benefit of this system is that the average error is combined with the other input parameters to determine whether a correction is needed. The downside of this system is the need to train and operate two models, one to determine the error without performance feedback, and another that uses the error of that model as an extra input parameter to carry out performance feedback.

Bibliography

[1] Kari Anderson, Francis Carr, Eric Feron, and William Hall. Analysis and modeling of ground operations at hub airports. In *3rd USA/Europe Air Traffic Management R&D Seminar*, 2000.

[2] Jason A. D. Atkin, Edmund K. Burke, John S. Greenwood, and Dale Reeson. On-line decision support for take-off runway scheduling with uncertain taxi times at london heathrow airport. *Journal of Scheduling*, 11(5):323, 2008.

[3] Jason A. D. Atkin, A. Sima Uyar, Ender Ozcan, and Neil Urquhart. *Airport Airside Optimisation Problems*, pages 1–37. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013. ISBN 978-3-642-39304-4. doi: 10.1007/978-3-642-39304-4_1. URL `https://doi.org/10.1007/978-3-642-39304-4_1`.

[4] Jason A.D. Atkin, Edmund K Burke, and Stefan Ravizza. The airport ground movement problem: Past and current research and future directions. In *Proceedings o f the 4th International Conference on Research in Air Transportation (ICRAT), Budapest, Hungary*, pages 131–138, 2010.

[5] Poornima Balakrishna. *Scalable approximate dynamic programming models with applications in air transportation*. PhD thesis, George Mason University, 2009.

[6] 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. ISSN 0968-090X. doi: https://doi.org/10.1016/j.trc.2010.03.003. URL `http://www.sciencedirect.com/science/article/pii/S0968090X1000029X`. Special issue on Transportation Simulation Advances in Air Transportation Research.

[7] Julia A. Bennell, Mohammad Mesgarpour, and Chris N. Potts. Airport runway scheduling. *4OR*, 9(2): 115, May 2011. ISSN 1614-2411. doi: 10.1007/s10288-011-0172-x. URL `https://doi.org/10.1007/s10288-011-0172-x`.

[8] Abdelghani Bouras, Mageed A Ghaleb, Umar S Suryahatmaja, and Ahmed M Salem. The airport gate assignment problem: a survey. *The scientific world journal*, 2014, 2014.

[9] F. Carr, A. Evans, J. Clarke, and E. Feron. Modeling and control of airport queueing dynamics under severe flow restrictions. In *Proceedings of the 2002 American Control Conference (IEEE Cat. No.CH37301)*, volume 2, pages 1314–1319 vol.2, May 2002. doi: 10.1109/ACC.2002.1023202.

[10] P. Chapman, J. Clinton, R. Kerber, T. Khabaza, T. Reinartz, C. Shearer, and R. Wirth. *CRISP-DM 1.0 Step-by-step data mining guide*, 2000.

[11] Gano Chatterji and Yun Zheng. Wheels-off time prediction using surface traffic metrics. In *12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference and 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2012. doi: 10.2514/6.2012-5699. URL `https://arc.aiaa.org/doi/abs/10.2514/6.2012-5699`.

[12] Gano Broto Chatterji and Yun Zheng. Wheels-off time estimation at non-asde-x equipped airports. In *2013 Aviation Technology, Integration, and Operations Conference*, 2013. doi: 10.2514/6.2013-4274. URL `https://arc.aiaa.org/doi/abs/10.2514/6.2013-4274`.

[13] D. Ciregan, U. Meier, and J. Schmidhuber. Multi-column deep neural networks for image classification. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3642–3649, June 2012. doi: 10.1109/CVPR.2012.6248110.

[14] European Commission. Commission implementing regulation (eu) no 716/2014 of 27 june 2014 on the establishment of the pilot common project supporting the implementation of the european air traffic management master plan. *Official Journal of the European Union*, 190, 2014.

[15] W. H. Dalmeijer. Gebruiksprognose 2020. Technical report, Schiphol, 2019.

[16] Eurocontrol. Airport collaborative decision-making (a-cdm) impact assessment. Technical report, Eurocontrol, 2016.

[17] Eurocontrol. Airport cdm implementation. Technical report, Eurocontrol, 2017.

[18] Manuel Fernández-Delgado, Eva Cernadas, Senén Barro, and Dinani Amorim. Do we need hundreds of classifiers to solve real world classification problems? *Journal of Machine Learning Research*, 15:3133–3181, 2014. URL `http://jmlr.org/papers/v15/delgado14a.html`.

[19] A. Futer. Improving etms' ground time predictions. In *2006 ieee/aiaa 25TH Digital Avionics Systems Conference*, pages 1–12, Oct 2006. doi: 10.1109/DASC.2006.313692.

[20] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[21] Thomas Günther, Matthias Hildebrandt, Hartmut Fricke, and Moritz Strasser. Contributions of advanced taxi time calculation to airport operations efficiency. *Journal of Aerospace Operations*, 1(1-2): 95–106, 2011.

[22] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182, 2003.

[23] Thomas Günther, Matthias Hildebr, Hartmut Fricke, and Moritz Strasser. Contributions of advanced taxi time calculation to airport operations efficiency. In *Air Transport and Operations : Proceedings of the First International Air Transport and Operations Symposium 2010*, 2010.

[24] Jiawei Han, Micheline Kamber, and Jian Pei. 3 - data preprocessing. In Jiawei Han, Micheline Kamber, and Jian Pei, editors, *Data Mining (Third Edition)*, The Morgan Kaufmann Series in Data Management Systems, pages 83 – 124. Morgan Kaufmann, Boston, third edition edition, 2012. ISBN 978-0-12-381479-1. doi: https://doi.org/10.1016/B978-0-12-381479-1.00003-4. URL `http://www.sciencedirect.com/science/article/pii/B9780123814791000034`.

[25] Lu Hao, Megan S. Ryerson, Lei Kang, and Mark Hansen. Estimating fuel burn impacts of taxi-out delay with implications for gate-hold benefits. *Transportation Research Part C: Emerging Technologies*, 80: 454 – 466, 2017. ISSN 0968-090X. doi: https://doi.org/10.1016/j.trc.2016.05.015. URL `http://www.sciencedirect.com/science/article/pii/S0968090X16300523`.

[26] Miwa Hayashi, Ty Hoang, Yoon Chul Jung, Waqar Malik, Hanbong Lee, and Victoria Lee Dulchinos. Evaluation of pushback decision-support tool concept for charlotte douglas international airport ramp operations. In *Eleventh USA/Europe Air Traffic Management Research and Development Seminar*, 2015.

[27] Floris Herrema, Richard Curran, Hendrikus Visser, Denis Huet, and Régis Lacote. Taxi-out time prediction model at charles de gaulle airport. *Journal of Aerospace Information Systems*, 15(3):120–130, 2018.

[28] 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.

[29] Husni R. Idris, Ioannis Anagnostakis, Bertrand Delcaire, R. John Hansman, John-Paul Clarke, Eric Feron, and Amedeo R. Odoni. Observations of departure processes at logan airport to support the development of departure planning tools. *Air Traffic Control Quarterly*, 7(4):229–257, 1999. doi: 10.2514/atcq.7.4.229. URL `https://doi.org/10.2514/atcq.7.4.229`.

[30] Husni Rifat Idris. *Observation and Analysis of Departure Operations at Boston Logan International Airport*. PhD thesis, Massachusetts Institute of Technology, 2001.

[31] Yoon Jung, Ty Hoang, Miwa Hayashi, Waqar Malik, Leonard Tobias, and Gautam Gupta. Performance evaluation of sarda: An individual aircraft-based advisory concept for surface management. *Air Traffic Control Quarterly*, 22(3):195–221, 2014. doi: 10.2514/atcq.22.3.195. URL `https://doi.org/10.2514/atcq.22.3.195`.

[32] Harshad Khadilkar and Hamsa Balakrishnan. Network congestion control of airport surface operations. *Journal of Guidance, Control, and Dynamics*, 37(3):933–940, 2014.

[33] Matthew Kistler and Gautam Gupta. Relationship between airport efficiency and surface traffic. In *9th AIAA Aviation Technology, Integration, and Operations Conference (ATIO)*, 2009. doi: 10.2514/6.2009-7078. URL https://arc.aiaa.org/doi/abs/10.2514/6.2009-7078.

[34] Kathryn Blackmond Laskey, Ning Xu, and Chun-Hung Chen. Propagation of delays in the national airspace system. In *UAI'06: Proceedings of the Twenty-Second Conference on Uncertainty in Artificial Intelligence*, 2006.

[35] Hanbong Lee, Waqar Malik, Bo Zhang, Balaji Nagarajan, and Yoon C. Jung. Taxi time prediction at charlotte airport using fast-time simulation and machine learning techniques. In *15th AIAA Aviation Technology, Integration, and Operations Conference*, 2015. doi: 10.2514/6.2015-2272. URL https://arc.aiaa.org/doi/abs/10.2514/6.2015-2272.

[36] Hanbong Lee, Waqar Malik, and Yoon C Jung. Taxi-out time prediction for departures at charlotte airport using machine learning techniques. In *16th AIAA Aviation Technology, Integration, and Operations Conference*, page 3910, 2016.

[37] Hanbong Lee, Jeremy Coupe, and Yoon C Jung. Prediction of pushback times and ramp taxi times for departures at charlotte airport. In *AIAA Aviation 2019 Forum*, page 2933, 2019.

[38] Guan Lian, Yaping Zhang, Jitamitra Desai, Zhiwei Xing, and Xiao Luo. Predicting taxi-out time at congested airports with optimization-based support vector regression methods. *Mathematical Problems in Engineering*, 2018, 2018.

[39] Yi Liu, Mark Hansen, Gautam Gupta, Waqar Malik, and Yoon Jung. Predictability impacts of airport surface automation. *Transportation Research Part C: Emerging Technologies*, 44:128 – 145, 2014. ISSN 0968-090X. doi: https://doi.org/10.1016/j.trc.2014.03.010. URL http://www.sciencedirect.com/science/article/pii/S0968090X14000825.

[40] LVNL. Outbound planning operational concept document. Technical report, LVNL, 2019. Confidential.

[41] Núria Macià and Ester Bernadó-Mansilla. Towards uci+: A mindful repository design. *Information Sciences*, 261:237 – 262, 2014. ISSN 0020-0255. doi: https://doi.org/10.1016/j.ins.2013.08.059. URL http://www.sciencedirect.com/science/article/pii/S0020025513006336.

[42] M. Mitchell. *An Introduction to Genetic Algorithms*. A Bradford book. Bradford Books, 1998. ISBN 9780262631853. URL https://books.google.nl/books?id=0eznlz0TF-IC.

[43] Herbert Naessens, Thomas Philip, Marcin Piatek, Kristof Schippers, and Robert Parys. Predicting flight routes with a deep neural network in the operational air traffic flow and capacity management system. Technical report, EUROCONTROL Maastricht Upper Area Control Centre, 2017.

[44] Alex H. Nakahara and Tom G. Reynolds. Estimating current & future system-wide benefits of airport surface congestion management. In *10th USA/Europe Air Traffic Management Research and Development Seminar*, 2013.

[45] Patricia Pina and Jose Miguel De Pablo. Benefits obtained from the estimation and distribution of realistic taxi times. In *ATM R&D Seminar*, 2005.

[46] S. Ravizza, J. A. D. Atkin, M. H. Maathuis, and E. K. Burke. A combined statistical approach and ground movement model for improving taxi time estimations at airports. *Journal of the Operational Research Society*, 64(9):1347–1360, Sep 2013. ISSN 1476-9360. doi: 10.1057/jors.2012.123. URL https://doi.org/10.1057/jors.2012.123.

[47] Stefan Ravizza, Jun Chen, Jason A.D. Atkin, Paul Stewart, and Edmund K. Burke. Aircraft taxi time prediction: Comparisons and insights. *Applied Soft Computing*, 14:397 – 406, 2014. ISSN 1568-4946. doi: https://doi.org/10.1016/j.asoc.2013.10.004. URL http://www.sciencedirect.com/science/article/pii/S1568494613003384.

[48] Derek Robinson and Daniel Murphy. Aircraft taxi times at u.s. domestic airports. In *10th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference*, 2010. doi: 10.2514/6.2010-9147. URL `https://arc.aiaa.org/doi/abs/10.2514/6.2010-9147`.

[49] Melanie Sandberg, Tom G Reynolds, Harshad Khadilkar, and Hamsa Balakrishnan. Airport characterization for the adaptation of surface congestion management approaches. In *10th USA/Europe Air Traffic Management Research and Development Seminar*, 2013.

[50] Schiphol. Traffic review 2018. Technical report, Schiphol, 2019.

[51] R. A. Shumsky. *Dynamic statistical models for the prediction of aircraft take-off times*. PhD thesis, Massachusetts Institute of Technology, 1995.

[52] D. B. Signor and B. S. Levy. Accurate oooi data: Implications for efficient resource utilization. In *2006 ieee/aiaa 25TH Digital Avionics Systems Conference*, pages 1–12, 2006. doi: 10.1109/DASC.2006.313676.

[53] Ioannis Simaiakis. *Analysis, Modeling and Control of the Airport Departure Process*. PhD thesis, Massachusetts Institute of Technology, 2013.

[54] Ioannis Simaiakis and Hamsa Balakrishnan. Impact of congestion on taxi times, fuel burn, and emissions at major airports. *Transportation research record*, 2184(1):22–30, 2010.

[55] Ioannis Simaiakis and Hamsa Balakrishnan. A queuing model of the airport departure process. *Transportation Science*, 50(1):94–109, 2016. doi: 10.1287/trsc.2015.0603. URL `https://doi.org/10.1287/trsc.2015.0603`.

[56] Ioannis Simaiakis and Nikolas Pyrgiotis. An analytical queuing model of airport departure processes for taxi out time prediction. In *10th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference*, 2010. doi: 10.2514/6.2010-9148. URL `https://arc.aiaa.org/doi/abs/10.2514/6.2010-9148`.

[57] Ioannis Simaiakis, Harshad Khadilkar, Hamsa Balakrishnan, Tom G. Reynolds, and R. John Hansman. Demonstration of reduced airport congestion through pushback rate control. *Transportation Research Part A: Policy and Practice*, 66:251 – 267, 2014. ISSN 0965-8564. doi: https://doi.org/10.1016/j.tra.2014.05.014. URL `http://www.sciencedirect.com/science/article/pii/S0965856414001384`.

[58] Yufeng Tu. *Air Transportation System Performance: Estimationand Comparative Analysis of Departure Delays*. PhD thesis, The Robert H. Smith School of Business, 2006.

[59] J. VanderPlas. *Python Data Science Handbook*. O'Reilly Media, Inc., 2016. ISBN 9781491912058.

[60] Lei Yang, Suwan Yin, Ke Han, Jack Haddad, and Minghua Hu. Fundamental diagrams of airport surface traffic: Models and applications. *Transportation Research Part B: Methodological*, 106:29 – 51, 2017. ISSN 0191-2615. doi: https://doi.org/10.1016/j.trb.2017.10.015. URL `http://www.sciencedirect.com/science/article/pii/S0191261517303843`.

[61] Jianan Yin, Minghua Hu, Yuanyuan Ma, Ke Han, and Dan Chen. Airport taxi situation awareness with a macroscopic distribution network analysis. *Networks and Spatial Economics*, 19(3):669–695, 2019.

[62] Shannon Zelinski and Robert Windhorst. Modelling and simulating airport surface operations with gate conflicts. *The Aeronautical Journal*, 123(1259):1–19, 2019.