Predicting Time to Pushback of Flights in U.S. Airports

Daniil Filienko*
Yudong Lin*
Kyler Robison*
Trevor Tomlin*
Martine De Cock†
daniilf@uw.edu
ydlin@uw.edu
kylerr10@uw.edu
ttomlin@uw.edu
mdecock@uw.edu
School of Engineering and Technology, University of Washington Tacoma
Tacoma, Washington, USA

ABSTRACT

Air traffic management systems need to predict many details about flights as accurately as possible. Of particular interest is the pushback time, i.e. the moment at which an aircraft is pushed backwards, away from its parking position at the gate. Accurate pushback time predictions can in turn yield more accurate predictions of takeoff time. In this paper we propose a gradient boosting decision tree model for pushback time prediction, trained on a rich feature set encompassing data about weather, airport activity, airline, and aircraft characteristics. In evaluating our approach on a large dataset with data from 10 U.S. airports, we found that training one local model for each airport is more memory efficient, while yielding a mean absolute error at par with a global model trained over the data of all airports combined. Our approach was among the winners of the 2023 "Pushback to the Future" competition hosted by NASA.

CCS CONCEPTS

• **Computing methodologies** → **Ensemble methods**; **Feature selection**; • **Applied computing** → **Transportation**.

KEYWORDS

flight time, gradient boosting decision trees, feature engineering

ACM Reference Format:
Daniil Filienko, Yudong Lin, Kyler Robison, Trevor Tomlin, and Martine De Cock. 2023. Predicting Time to Pushback of Flights in U.S. Airports. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX)*. ACM, New York, NY, USA, 6 pages. https://doi.org/XXXXXXX.XXXXXXX

*Contributed equally to the paper
†Guest Professor at Ghent University

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

To efficiently use the limited capacity of airports and runways, air traffic management systems rely on predictions made by machine learning (ML) models about many correlated aspects of each flight, such as the estimated time that a departing flight will leave the gate, the taxi time, and the estimated time of departure [4–6]. In this paper we describe a model that we designed and trained in response to the 2023 "Pushback to the Future" competition hosted by NASA,[1] in which participants were challenged to develop models for accurate prediction of the pushback time of a flight, i.e. the moment at which a departing aircraft is pushed back from its parking position at the gate. Accurate predictions of pushback time can in turn yield more accurate predictions of flight takeoff time.

The pushback time of flights is influenced by a myriad of factors, including the weather, the activity level at the airport, the airport configuration, and characteristics of the aircraft and the airline. We describe in this paper how we derived a feature set that captures this information from available weather and air traffic data, and how we successfully trained gradient boosting decision tree models over this feature set to minimize cumulative MAE (mean absolute error). When evaluating our approach on a large set of data from 10 U.S. airports, we found that the same low MAE can be obtained whether one trains a global model on the data of all airports combined or whether one trains a local model for each airport, the latter strategy being computationally more efficient.

After describing the data and the problem in more detail in Sec. 2, in Sec. 3 we provide an overview of our ML pipeline and a detailed overview of the feature set. In Sec. 4 we present the results of our method when evaluated on a held-out validation set and compared with a baseline strategy. In Sec. 5 we conclude with our main findings and directions for future work.

2 DATA AND PROBLEM DESCRIPTION

Throughout this paper, by "pushback time" we mean the moment at which an aircraft departs from the gate. We address the problem of *predicting the number of minutes until pushback time*. For a given flight, it is useful to estimate this time gap repeatedly at various

[1] https://www.drivendata.org/competitions/182/competition-nasa-airport-pushback-prescreened/

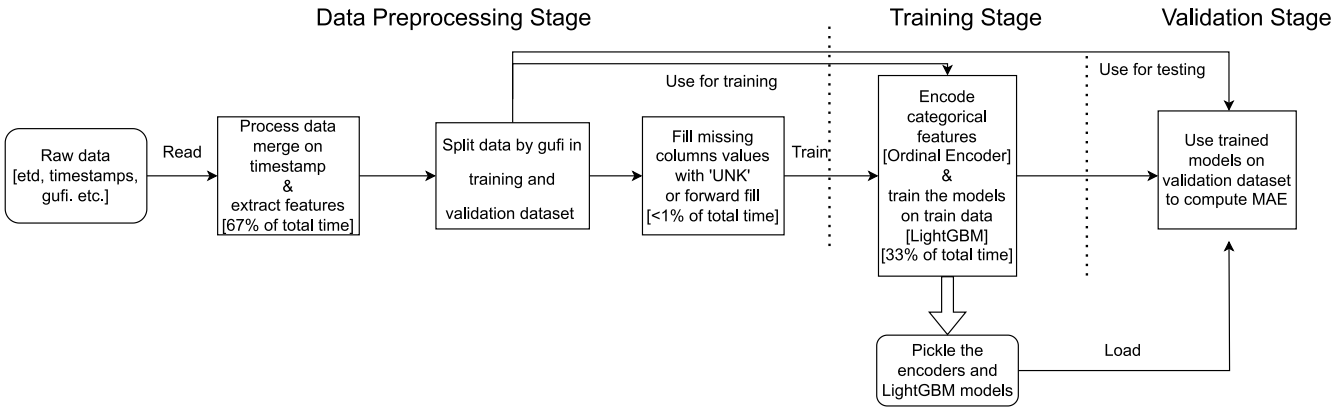

Figure 1: Machine learning pipeline with data preprocessing (including feature extraction), training, and validation steps

Table 1: Sample instances showing minutes_until_pushback beginning at 04:00:00 for a flight at Chicago O'Hare (KORD)

gufi		
SKW5143.ORD.EAU.201031.0059.0006.TFM		
timestamp	**airport**	**minutes_until_pushback**
2020-11-15 04:00:00	KORD	85
2020-11-15 04:15:00		70
2020-11-15 04:30:00		55
2020-11-15 04:45:00		40
2020-11-15 05:00:00		25
2020-11-15 05:15:00		10

Table 2: Flight dataset statistics

	train data		validation data	
airport	#gufis	#instances	#gufis	#instances
KATL	540,273	3,321,960	50,165	305,397
KCLT	381,045	2,218,680	35,708	199,808
KDEN	451,366	2,963,510	42,522	284,439
KDFW	498,426	3,210,397	46,279	298,854
KJFK	203,538	1,278,767	16,609	99,993
KMEM	151,768	1,236,371	13,860	121,167
KMIA	232,846	1,456,633	20,923	126,651
KORD	487,075	3,052,963	46,708	297,823
KPHX	279,203	1,757,201	26,024	163,252
KSEA	286,416	3,511,956	26,140	158,018
total	3,511,956	24,008,438	324,938	2,055,402

points before the plane's actual departure. As illustrated in Tab. 1, each instance for which a prediction needs to be made is characterized by the flight's Globally Unique Flight Identifier (gufi) [1], a timestamp, and the departure airport. The last column in Tab. 1 corresponds to the target variable.

We trained and evaluated our models on data made available in the prescreened arena of the "Pushback to the Future" competition.[1] The data encompasses 3,836,894 flights across 10 airports of departure in the U.S.. As described in Sec. 3.1, we split the data into train and validation data without overlap between gufis in train and validation data (see Tab. 2). Each instance in the dataset corresponds to a row of the form shown in Tab. 1. The number of instances in Tab. 2 is much larger than the number of gufis, because there are on average (more than) 6 instances per gufi, as illustrated in Tab. 1.

The data contains auxiliary information and measurements for each airport that can be leveraged to improve the accuracy of pushback time prediction: (1) estimated departure (takeoff) times of the plane from the runway into the air; (2) actual runways and runway arrival and departure times; (3) active runway configuration at different times; (4) metadata about the flight and aircraft (engine class, aircraft type, etc.); (5) actual gate arrival and departure times; (6) time when flights started to be tracked by the National Airspace System (NAS); and (7) weather predictions for the airport.

The *weather data* originates from the Localized Aviation MOS (Model Output Statistics) Program (LAMP), a weather forecast service operated by the National Weather Service.[2] The rest is *air traffic data* that is collected through Fuser, a NASA data processing platform that processes the Federal Aviation Administration (FAA)'s raw data stream and distributes cleaned, real-time data on the status of individual flights nationwide [2].

Except for the aircraft metadata, all data points are associated with timestamps. When making a prediction for a given flight at a given timestamp t, only auxiliary data with a timestamp t' such that $t' \leq t$ can be leveraged. In particular, the actual runway and the actual gate and runway departure time of a flight are not known yet at the time of prediction. Such information about flights that have already departed can however be used to estimate the current level of activity at an airport. For more details about the data, we refer to the competition webpage.[3]

3 METHOD

3.1 System Overview

We follow a strategy similar to studies that have been done on the related subject of flight delay classification [4], which tend

[2] https://vlab.noaa.gov/web/mdl/lamp
[3] https://www.drivendata.org/competitions/182/competition-nasa-airport-pushback-prescreened/

Table 3: Overview of features

Airport features	Agg	Description
airport		Name of the departure airport associated with the particular gufi.
1hr_ETDP	✓	Average departure delay, time between estimated and true departure time in the past hour.
deps_nhr	✓	Number of flights that have departed from the prediction airport in the past n hours.
arrs_nhr	✓	Number of flights that have arrived at the prediction airport within the past n hours.
deps_taxiing	✓	Number of flights that departed from the runway in the past 30 hours.
exp_deps_nmin	✓	Number of flights expected to depart from the prediction airport within the next n minutes.
delay_nhr	✓	Average difference between the estimated and true departure time of all flights at the prediction airport within the past n hours. [4]
standtime_nhr	✓	Average difference between the first position (timestamp of gufi creation) and pushback time at the origin airport for all arrival flights at the prediction airport in the past n hours.
dep_taxi_nhr	✓	Average time that all departing flights in the past n hours spent taxiing.
minutes_until_etd	✓	Number of minutes between the prediction timestamp and the most recent departure prediction for the flight, i.e. the number of minutes until the estimated time of departure (etd). The etd as predicted by the Fuser system is readily available in the data; it varies substantially in accuracy.
gufi_lifespan	✓	Number of minutes between the prediction timestamp and when the flight gufi has originated.
derived_destination		Last 3 characters of the ICAO code for the flight's destination airport, extracted from the gufi string. [4]
departure_runways		Collection of runway names currently utilized at the airport for departures.
arrival_runways		Collection of runway names currently utilized at the airport for arrivals. [4]
Time features	Agg	Description
minute		Minute of the prediction timestamp.
hour		Hour of the prediction timestamp. [4]
day		Day of the month of the prediction timestamp.[4]
weekday		Weekday of the prediction timestamp. [4]
month		Month of the prediction timestamp. [4]
year		Year of the prediction timestamp.
Aircraft and airline features	Agg	Description
derived_carrier		The code of the airline operating the flight, extracted from the gufi string. It is valuable because certain companies may have unique pushback tendencies, such as FedEx (FDX), which is a large cargo carrier. FDX is a great example of a carrier that makes this feature so useful because the nature of how they operate directly affects the pushback time. Their flights park in different parts of the airport, load only cargo, and often operate at volume throughout the nighttime; times when passenger flights often see a drop in frequency. Apart from this major example, many passenger airlines operate differently from one another, a difference which was also successfully learned by the model.
major_carrier		Code of the airline operating the flight. For small, regional flights that major airlines have operated by smaller airlines, this code will be that of the larger airline that is doing the contracting. [4]
aircraft_type		Model of the aircraft in a short string. "Boeing 737-800" would be represented as "B738".
flight_type		Type of flight, "scheduled air transport" or not (most flights are the former).
Weather features	Agg	Description
temperature		Temperature reading around the destination airport.
wind_direction		Wind direction in compass heading divided by 10 and rounded to the nearest integer (to match runway codes).
wind_speed		Wind speed in knots.
wind_gust		Wind gust speed in knots.
cloud_ceiling		Cloud ceiling height in feet encoded categorically.
visibility		Visibility in miles encoded categorically.
cloud		Cloud cover encoded categorically.
lightning_prob		Probability of lightning encoded categorically.

to process raw data obtained from various sources, merging and aggregating the data based on timestamps, extracting features, and encoding categorical features to be able to utilize them in model training and validation. We follow that approach, as presented in Fig. 1, with the additional steps of saving the models and separating the resulting dataset into training and validation dataset by gufi, so that the same flight does not occur in both training and validation datasets, and both datasets have flights that have varying departure times, in order to best emulate a real deployment scenario.

Data preprocessing and feature extraction. Our approach relies heavily on feature engineering. Sec. 3.2 contains an overview of the features that we extract from the raw data by performing various operations, such as calculating future expected values, recording average differences, or counting occurrences of flights satisfying certain relevant conditions. Unlike Kiliç and Sallan [4] we do not remove rows with missing attributes. Instead we use various data imputation approaches, including (1) forward fill, if the data updates were rare across timestamps and did not change in between, (2) replacing the missing values with a mean value of the feature, or, more frequently, (3) treating the absence of a value as a separate feature value, labeling it as 'UNK'. We encode all categorical features with ordinal encoders, mapping categorical feature values to integers and creating a separate category for null data points to account for the absence of data.

Model training and validation. We train gradient boosting decision tree models [3] over the extracted features. We train a global model $\mathcal{M}$ over the data of all airports combined, as well as 10 local models $\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_{10}$ trained separately on the data of each airport. After model training, we evaluate the models on the validation data. To compute the cumulative MAE, we take the weighted average of the individual airport results with the weights representing the relative size of the validation dataset for each airport.

Table 4: Results in terms of MAE

	baseline	MLP		LightGBM	
airport	ETD-15	global	local	global	local
KATL	11.6135	8.0039	7.9851	7.8549	7.8682
KCLT	12.4652	9.4824	9.5111	9.3135	9.3352
KDEN	15.1364	12.0029	12.1098	11.9466	11.9446
KDFW	15.5376	12.3230	12.2801	12.0823	12.1768
KJFK	14.7204	11.5790	12.0592	11.1881	11.1143
KMEM	25.9130	18.7407	18.8432	17.7555	17.8176
KMIA	14.6529	12.4397	12.9948	11.9421	12.0881
KORD	14.5247	11.3000	11.4879	11.1351	11.2458
KPHX	12.5830	9.7497	9.6736	9.3849	9.4106
KSEA	10.1884	7.5612	9.6736	7.2195	7.2300
All	14.3250	10.9883	11.0500	10.7232	10.7686

3.2 Feature Overview

Our models consume more than 30 features that are either readily available or that are extracted by aggregating relevant information. The features capture weather conditions around the airport, and air traffic data to gauge the busyness at the given airport. Features of the latter kind are included based on the common-sense assumption that busier airports tend to be less efficient and therefore could have slightly different average pushback time. Tab. 3 contains an overview of all features, separated in four feature categories described below. The column "Agg" in Tab. 3 indicates whether the values of the feature are obtained by aggregating information from two or more rows of data as opposed to extracted directly from one data point. The terms "prediction airport" and "prediction timestamp" systematically refer to the airport and the timestamp in the instances for which predictions need to be made, or, during training time, the airport and the timestamp in the labeled instances such as those in Tab. 1.

- **Airport busyness features**: included with the common-sense assumption that flights are more likely to be delayed or rescheduled, increasing their pushback time, when there are many other flights arriving or waiting to depart around the same time.
- **Time features**: used as an indirect measure of the busyness of an airport, reflecting the assumption that airports do not operate with the same efficiency/capacity at different times of the day.
- **Aircraft features**: correlating with total passenger/cargo carried by the flight, its speed, length of preparation needed to fly, and other characteristics.
- **Weather features**: includes the probability of lightning, type of clouds, and other information, which can potentially be strongly correlated to the current capacity and measure of congestion in the airport, since in severe conditions, flights can be canceled or delayed, affecting average flight pushback time.

We arrived at this feature set through a trial-and-error feature engineering process in which we repeatedly applied the model training and validation steps (see Fig. 1) for a growing feature set, inspecting the resulting feature importance graphs (such as the one presented in Fig. 2) to analyze model performance and to remove features with low importance. Knowing that expanding the feature set could increase the risk of overfitting, we built out the feature set in a greedy bottom-up manner, adding one feature at a time and only maintaining features that yielded a consistent significant improvement on the training and the validation data. As we report in Sec. 4, our final models were evaluated on a separate "hidden" test set. Unlike the training and validation data, we did not use this test data during training, feature selection, or hyperparameter tuning at all.

4 RESULTS

4.1 Experimental setup

Tab. 4 shows the mean absolute error (MAE) on the validation data (see Tab. 2) of predictions obtained with five different approaches:[4]

(1) Baseline method, based on the rough assumption that pushback starts around 15 minutes before departure time. To make a prediction with this baseline method, we compute the estimated number of minutes until pushback as **minutes_until_etd** −15. See Tab. 3 for a description of the feature **minutes_until_etd**.
(2) Global MLP model, which is a neural network $\mathcal{M}$ trained over the features in Tab. 3 and across the data of all airports combined. We use the same global model $\mathcal{M}$ to make predictions for each airport.
(3) Local MLP models, which are neural networks $\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_{10}$ trained separately on the data of each airport, using the same feature set as the global MLP model. We make predictions for each airport using the appropriate corresponding local MLP model that was trained on the data from that airport.
(4) Global tree model, which is a LightGBM model trained over the features in Tab. 3 and across the data of all airports combined.
(5) Local tree models, which are LightGBM models trained separately on the data of each airport, similar as approach (3).

The cumulative MAE in the bottom row of Tab. 4 is a weighted average of the MAEs of the individual airports, with the weights representing the relative size of the validation dataset for each airport.

To train the gradient boosted decision tree models, we used the LightGBM library[5] with the default hyperparameter settings unless noted otherwise. We used `LGBMRegressor` with objective function `regression_l1` which means that the model was trained to minimize the MAE of the predictions. The `num_leaves` hyperparameter controls how many leaves each weak learner has. A higher number of leaves can increase the model's performance, but it can also lead to overfitting on the data. In this case, the value of 4096 was chosen because we found that during hyperparameter tuning using Optuna[6] it achieved the lowest MAE on the validation dataset. Finally, the `n_estimators` hyperparameter determines the number of trees in the LightGBM model. Through the use of Optuna and manual tuning, we selected value 128 for the number of estimators (trees).

To train the MLPs, we utilized TensorFlow.[7] Each MLP consists of a normalization layer, followed by three dense ReLU layers with 32, 64, and 64 neurons respectively, and a linear layer with 1 output

[4]We will make the code repository public when Phase 2 of the "Pushback to the Future" competition ends. tinyurl.com/flhuskies
[5]https://lightgbm.readthedocs.io/
[6]https://optuna.org/
[7]https://www.tensorflow.org/api_docs/python/tf

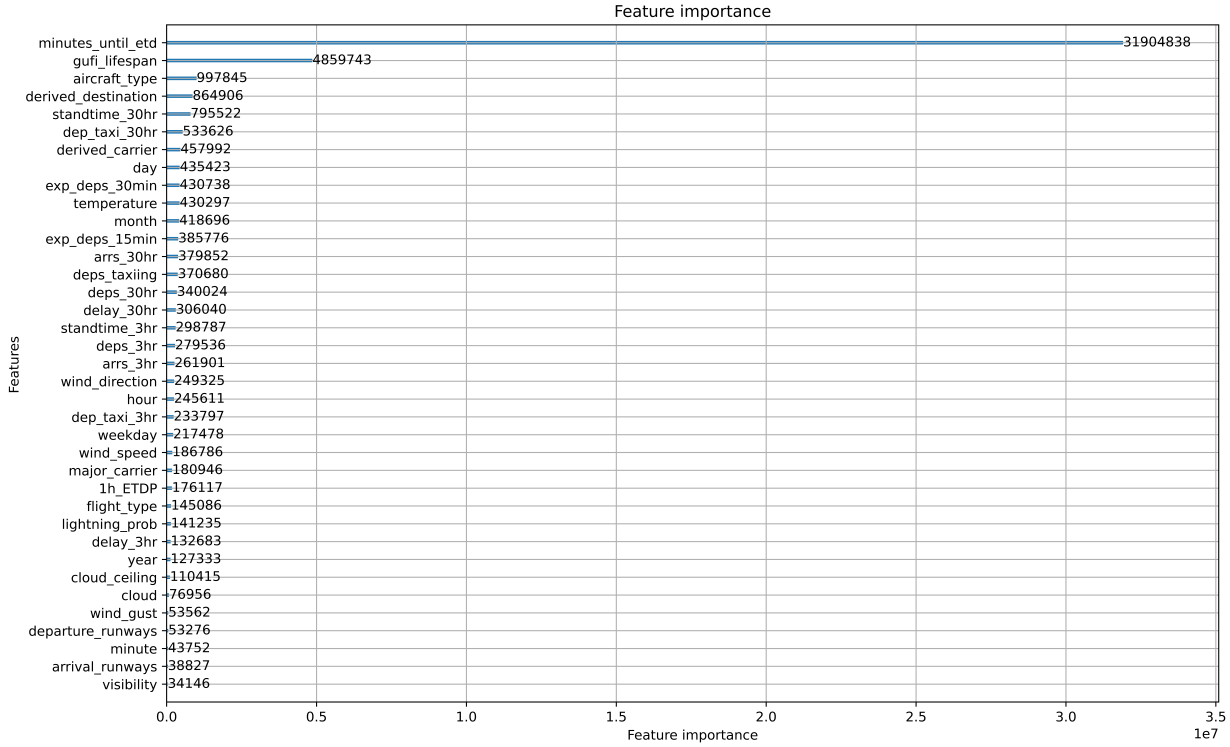

Figure 2: KATL Airport Feature Importance

neuron. We used MAE as the loss functions and the default setting for other hyperparameters.

While the results for MLP in Tab. 4 are worse than those for LightGBM, we acknowledge that the MLP results could potentially be improved through further hyperparameter tuning, at a sizable computational cost. Indeed, training LightGBM models on the extracted feature set requires less than an hour (see Sec. 4.2.3 for hardware specifications), allowing rapid prototyping and tuning, while the global MLP model requires 3 hours of training, and an ensemble of local MLP models requires upwards of 8 hours of training. In the remainder of the paper, we focus on the LightGBM results.

4.2 Utility and Efficiency Analysis

4.2.1 Utility Results. As can be seen in Tab. 4, the baseline model results in a mean absolute error (MAE) of 14.325, meaning that predictions made by the baseline model differ, on average, approximately 14 minutes from the actual pushback time. Predictions made by the baseline model are obtained by subtracting 15 minutes from the **minutes_until_etd** feature in Table 3, which is in itself based on an estimate of the time of departure that is made available through NASA's data processing platform Fuser. Errors made by the baseline model therefore accumulate from two different sources, namely (1) lack of accuracy in estimates of the time of departure as available through the Fuser system; and (2) the interval between the moment of pushback and the actual departure of the aircraft from the runway is not (always) precisely 15 minutes.

As shown in the other columns of Tab. 4, the baseline predictions can substantially be improved upon by incorporating all the other features from Tab. 3. While building out our feature set, as described in Sec. 3, we observed that for smaller subsets of features the "local model" based approach would result in lower MAEs than the "global model" based approach. When training LightGBM models over the entire feature set from Tab. 3 however, the global model and the local model based approaches are at par, yielding an MAE or around 10.7 on the validation data, which is a substantial improvement over the 14.3 baseline result.

Another interesting observation from Tab. 4 is that the MAE varies significantly between the different airports, with the same trend manifesting itself across all approaches. The average prediction errors vary from around 7 minutes for Seattle-Tacoma International Airport (KSEA) to more than 17.5 minutes for Memphis International Airport (KMEM). We conjecture that the latter stems from KMEM's role as the busiest cargo airport in the U.S..

4.2.2 Feature Importance Analysis. Fig. 2 represents the relative importance of each feature used in the LightGBM model for the KATL airport, computed by calculating the total gains of splits which use the feature, as specified in the LightGBM description.[8] Each of the features individually may vary in its relative importance for each airport, but in general, the placement does not change

[8]https://lightgbm.readthedocs.io/en/latest/pythonapi/lightgbm.plot_importance.html

drastically, with an exception of KMEM, potentially because of its unique status as primarily a cargo airport.

As can be expected, the most important feature for predicting true pushback is the estimated time of departure (etd), which provides an "anchor" from which we can approximate the time pushback will happen, since the pushback happens shortly before the departure from the runway. Another very important feature is the gufi lifespan, i.e. the time difference in minutes between the time when the gufi was generated, tracking the flight, and the current time, i.e. the timestamp at which a prediction is made. Interestingly, we found that knowing the destination airport is a very valuable piece of evidence for all airports, possibly because each destination airport has certain specific type of flights that tend to fly there. Many interval/average features regarding departures and arrivals such as "exp_deps_30min", which encodes the total number of flights expected to depart within the next 30 minutes, "dep_taxi_30hr", which represents the average time taken by flights to taxi from the gate to the runway in the past 30 hours, and "standtime_30hr", which is the average time between when flights started to be tracked and their pushback time during the last 30 hours, are also among the most important features. These features capture the busyness of the analyzed airport, which can affect the pushback time.

Among the aircraft and airline features, we observed that the aircraft type and the carrier are very valuable. The aircraft type indirectly captures the size, style, and other aircraft characteristics, thereby providing useful information to estimate the time required for an airplane to become airborne. Temporal information such as month, hour, and day were also found to be fairly important. The observation that "airport", i.e. the departure airport, does not appear at all in Fig. 2, is consistent with the fact that the model corresponding to the feature graph in Fig. 2 was trained using data from KATL only, i.e. over a dataset in which the airport feature has only a single value, namely KATL. As a result, the total gain of the airport feature is 0 for this and all other local models.

4.2.3 Efficiency Analysis. We utilized Microsoft Azure cloud computing for data preprocessing and model training. The runtimes reported below were achieved with a VM running x64 Ubuntu Server v20.04 on an Intel Xeon Platinum CPU with 44 cores and 352 GB of RAM. Execution of the entire model construction pipeline on the training dataset took roughly 4 hours on average, namely 3 hours for data preprocessing and feature extraction, and roughly 40 minutes for fitting encoders and training LightGBM models. Inference duration is dependent on the number of instances. Classifying 2,042,723 instances takes around 20 minutes. Memory usage during all these operations is quite high. Ideally, a computer with a minimum of 128 gigabytes of RAM should be used; using less could lead to slower execution or possibly a failure to execute.

While they are at par in terms of utility, the "local model" approach is significantly more computationally efficient than the "global model" approach. Indeed, the models in the local model approach can be fit simultaneously in parallel on multiple machines, leading to higher training modality and fault-tolerance. Furthermore, the local model approach has significantly higher memory efficiency, because fitting a single model per airport requires holding in memory only one airport dataset at a time.

4.2.4 Results on Competition Leaderboard. We merged the validation set from Sec. 2 with the training data and retrained the local LightGBM models on this combined data, resulting in an MAE of 11.1046 on the competition leaderboard (4th place). The MAEs obtained by the teams in the 1st, 2nd, and 3rd positions were (respectively): 10.6734, 10.7283, and 11.0543. Based on available high-level descriptions, all these approaches utilize decision tree ensembles similar to our solution.[9]

5 CONCLUSION AND FUTURE WORK

In this paper we presented a gradient boosting decision tree model approach to predict the number of minutes until the pushback of a flight. Our main findings are that (1) more accurate predictions can be made by integrating information from many different sources, including weather, airport, and airline information, (2) the mean absolute errors of predictions made by training a local model per airport are at par with the errors of one global model trained across all airports, and (3) training local models is computationally more efficient. Each of our models is trained on data from many airlines. A roadblock regarding the latter is that some valuable information collected by airlines that is relevant to pushback time, like the number of passengers that have checked in for a flight or the number of bags that have been loaded onto a plane, may in practice be too sensitive to share with other entities for model training. An important next step would be designing a federated learning solution that will enable airlines to contribute this valuable information in a privacy-preserving manner.

6 ACKNOWLEDGMENTS

The authors would like to thank NASA and the team at DrivenData for hosting an interesting competition. They would also like to thank Sikha Pentyala and Anderson Nascimento for their help and advice during model development and evaluation, and Microsoft for the generous donation of cloud computing credits through the UW Azure Cloud Computing Credits for Research program.

REFERENCES

[1] Flight Information Exchange Model (FIXM). 2014. Globally Unique Flight Identifier (GUFI) Form and Content. https://www.fixm.aero/documents/GUFI%20Format%20v2%201_Final.pdf.
[2] Shawn M Gorman. 2019. Fuser and Fuser in the Cloud. In *NASA Airspace Technology Demonstration 2 (ATD-2) Industry Workshop.*
[3] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. LightGBM: A highly efficient gradient boosting decision tree. *Advances in Neural Information Processing Systems* 30 (2017).
[4] Kerim Kiliç and Jose M. Sallan. 2023. Study of Delay Prediction in the US Airport Network. *Aerospace* 10, 4 (2023).
[5] Hanbong Lee, Jeremy Coupe, and Yoon C Jung. 2019. Prediction of pushback times and ramp taxi times for departures at Charlotte airport. In *AIAA Aviation 2019 Forum.*
[6] Hanbong Lee, Waqar Malik, and Yoon C Jung. 2016. Taxi-out time prediction for departures at Charlotte airport using machine learning techniques. In *16th AIAA Aviation Technology, Integration, and Operations Conference.*

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009

[9] https://drivendata.co/blog/airport-pushback-finalists