Predicting Arrival and Departure Runway Assignments with Machine Learning

Andrew Churchill*
Mosaic ATM, Inc., Leesburg, VA, U.S.A.

William J. Coupe† and Yoon C. Jung‡
NASA Ames Research Center, Moffett Field, CA, 94035, U.S.A.

Runway assignments at major airports are made by air traffic controllers subject to various constraints, and to achieve various objectives. In this research, we describe our efforts training machine learning (ML) models to predict both departure and arrival runway assignments using an entirely data-driven approach. This approach is compared to existing rule-based approaches developed in previous research using input from Subject Matter Experts. The models have features derived from various FAA data feeds, and leverage multiple machine learning algorithms. Results for models trained for nine major U.S. airports are described and compared to one another across various important dimensions. Particular attention was paid to developing a repeatable framework for training these models so the approach could be scaled to other airports, and to developing models that are useful in a real-time environment. In addition, the models were designed to be functional in a real-time environment to support NASA's ATD-2 project, as part of an ML-powered shadow system to compare against the performance of the fielded system.

I. Introduction

Assigning flights to runways at an airport is a critical function that influences all aspects of airport operations and performance. These assignments, made by air traffic controllers, indicate the runway on which a flight must land or take off from. In this paper, we describe our research developing data-driven machine learning (ML) models to predict runway assignments for arriving and departing flights for various airports in the U.S.

While the precise roles and responsibilities vary for arriving and departing flights, air traffic controllers in general have tremendous expertise in making these runway assignments. They apply a heuristic based on a multitude of factors including at least direction of flight, airport configuration (i.e., which runways are available for arrivals and departures), configuration of nearby airports (e.g., in the same metroplex region), aircraft size and engine type, noise and other environmental regulations, and traffic volumes. They also consider requests from flight crews for specific runways due to preference or operational necessity. Furthermore, the rules applying to each of these factors may evolve over time as new procedures or routes are developed. This heuristic is clearly complex and learned by controllers over years of site-specific training and experience. Some prior research such as Isaacson et al. [1] has attempted to use observational studies on Subject Matter Experts (SMEs) to capture the decision heuristics that controllers employ to assign runways. Likewise, the development of a similar knowledge base has taken place on the ATD-2 project [2] (of which this research is part), and against which the results of the ML-based approach described here will be compared. The ATD-2 expert-driven approach consists of decision trees. For example, the decision tree may have a path such that all flights with a jet engine, headed for the BLECO fix, while the airport is operating in "South Normal" configuration, will be assigned to depart from runway 18L. The challenge with expert-driven approaches like these is that they are expensive and difficult to construct, given the requirement for access to personnel with specialized and timely knowledge, and the highly-nuanced differences in the heuristics that each individual might apply. Our approach in this research is to leverage the vast amounts of data on airspace and airport operations, in conjunction with modern ML techniques, to predict these runway assignments with performance nearly equivalent to that which is achievable through the expert-driven approach.

*Principal Analyst, Mosaic ATM and Principal Data Scientist, Mosaic Data Science, achurchill@mosaicatm.com, AIAA Member
†Aerospace Engineer, NASA Ames Research Center, Mail Stop 210-6, Moffett Field, CA 94035, william.j.coupe@nasa.gov, AIAA Member
‡Aerospace Engineer, NASA Ames Research Center, Mail Stop 210-6, Moffett Field, CA 94035, yoon.c.jung@nasa.gov, AIAA Senior Member

Whereas research has been conducted to apply machine learning to various prediction problems relating to aircraft and airport operations (e.g., predicting airport configurations, landing times, taxi out times), relatively little attention has been paid to predicting runway assignments. However, two recent works do cover the problem in an international context. In [3], the author developed a predictive model for runway assignments using support vector machines (SVM) for Amsterdam Schiphol Airport. Likewise, in [4], the authors developed a neural network based ML approach to predict the runway to be used by arrivals into Tokyo International Airport. Our work builds on this prior research by employing a more generic framework that can be easily applied to multiple airports, by focusing on deployment to a real-time system, and by exploring different ML approaches and metrics to understand model performance.

Rather than focusing on the problem of *predicting* runway assignments, considerable previous research has addressed on the problem of *optimizing* runway assignments (and sequence of runway use) to achieve various objectives. For example, Berge et al [5] jointly optimize the runway assignment and sequencing decisions. Likewise, see Lohr et al [6] for work that jointly optimizes airport configurations and runway assignments. An example with a longer time scale is [7], in which the authors planned a longer-term runway assignment strategy to achieve various delay, environmental, and safety related objectives.

This distinction between prescription (i.e., optimization) and prediction is critical in justifying the approach that we have developed. Although previous research on optimizing runway assignments and utilization has demonstrated potential benefits (e.g., greater runway throughput, shorter taxi distances), these approaches in general make significant assumptions about the ease with which such changes could be implemented. Our approach assumes that air traffic controllers will continue to use their considerable expertise and available automation tools, and we seek to capture (through machine learning) the runway assignment heuristics that exist within this structure. This ML model will enable us to make detailed forecasts of runway utilization several hours in advance of the actual operation. This distinction is essential to understand the broader context of the research as part of an ML-powered shadow system to help evaluate the its ability to match the performance of the the expert-driven approaches currently used in the ATD-2 system. In the remainder of this paper, we describe our modeling approach, the data used for our study, results and discussion, and finally conclusions and directions for continued investigation.

II. Modeling Approach

In this section, we describe the approach developed to train ML models that imitate the decision heuristics used by air traffic controllers outlined in the previous section. The modeling approach developed here encompasses several elements, each of which is described in greater detail in the following subsections:

- *Requirements*: Several requirements drove our decision-making process in developing these ML models.
- *Target*: The target value is the runway identifier actually used by the flight in the historical data.
- *Features*: Based on literature review, brainstorming, and consultation with available SMEs, identify, explore, and compute relevant input features. This section also describes imputation and encoding strategies used for each feature.
- *Building the Dataset*: Several steps are described in this section that are used to translate from a list of features to a rectangular dataset that can be used by one of the machine learning algorithms we have evaluated.
- *Machine learning algorithms*: This section describes applying machine learning algorithms to the full prepared training dataset to create models.

A. Modeling Requirements

Important requirements informed the development of these models. First, it was of crucial importance that the models developed as part of this research be suitable for deployment in a system processing live data. Thus, they must use features that can be readily computed at runtime. Further, they must have a full suite of imputers to handle inevitably missing data, or alternatively, must have a well-defined filtering approach to ensure that non-imputed features are never passed to the model as nulls. Finally, their query time must be reasonable to support processing large batches of flights at regular intervals. This real-time support was necessary because these runway prediction models are part of a suite of ML models to be assembled into a shadow system to evaluate against the performance of the legacy ATD-2 systems.

Second, an additional objective in training these models is that the process used is highly *data-driven*, in that it does not require the maintenance of significant adaptation data (i.e., site-specific geometric or procedural information). Achieving this objective allows the models to be generated and updated rapidly to cover dozens of airports across the U.S., provided that standard data formats are employed.

Finally, the models needed to be trained (and served in real-time) using data from the ATD-2 Fuser [8], a data

processing and fusion system that is designed to handle many different data feeds and formats simultaneously, generating a standard relational format.

B. Target

The target value for these models is the actual runway on which each flight operated. For this effort, we derived these values from surveillance data (Airport Surface Detection Equipment, Model X [ASDE-X] or Traffic Flow Management System [TFMS]) and runway adaptation from the National Flight Data Center (NFDC) data [9] to ensure consistency and reliability of these crucial data elements. When surveillance located a flight within a runway polygon and with physics consistent with aircraft landing, it was straightforward to determine that the flight operated on that runway at a certain time. When this is not the case, which is relatively rare at many large airports, we used the airborne surveillance data from TFMS to infer which runway was used and at what time. This was accomplished with an approach developed by Robert Kille [10], under funding by NASA.

C. Features

The set of features used in training the arrival and departure runway assignment models to date is relatively simple, with considerable overlap between the two models. Table 1 summarizes which features are included in each model. Descriptions of each feature follow the table. As with any ML modeling workflow, this is an area of iterative exploration, and we believe that additional features may improve the predictive power of each model.

Table 1 List of modeling features

Feature	Arrivals	Departures
Aircraft engine class	x	x
Wake turbulence category	x	x
Planned fix	x	x
Flight plan filed	x	x
Airport configuration	x	x
Time since airport configuration changed	x	x
Time until estimated operation	x	x
TBFM-assigned runway	x	

1. Aircraft Engine Class

This categorical value (handled via one-hot encoding) indicates whether a flight has a jet, turboprop, or propeller engine. Discussions with SMEs and insight from previous ATD-2 work indicated that this was an important discriminator in runway assignments. For example, some runways are reserved exclusively for propeller-driven aircraft, due to the significant differences in their takeoff and landing performance. Missing values for this feature are imputed, by using the most frequently observed value in the training dataset. For most training datasets, this will be jet engine, as jets comprise the vast majority of operations at airports for which this model is relevant.

2. Wake Turbulence Category

This categorical value (handled via one-hot encoding) indicates the impact of the wake vortex induced by the flight. This roughly correlates with the size of the aircraft, and implies the separation required between operations on the same runway. As with engine class, the wake vortex category was identified during discussions with SMEs and in previous ATD-2 work. For example, at some airports, the largest aircraft may be restricted to using certain runways due to available length for takeoffs or geometric constraints. For all analysis, FAA RECAT wake vortex categories [11] are used. Missing values for this feature are imputed, by using the most frequently observed value in the training dataset to replace missing values. Typically there is only one dominant weight class, making this imputation insignificant.

3. Planned Fix

As part of filing a flight plan or through communication with air traffic controllers, a flight operator provides some indication about their path through the terminal area and which fix they plan to use to transition into / out of the terminal area. The fix name is generally available through the data feeds used to build training datasets, and is handled in the model through one-hot encoding. Observations with missing values are not used in training models. According to discussions with SMEs, for both arrivals and departures, this planned fix, in conjunction with the current airport configuration, provides significant information about which runway will be used.

4. Flight Plan Filed

This boolean indicates whether a flight plan has been filed by the flight operator. The flight plan provides the planned fix, as described in the previous section. However, before the flight plan is actually filed, the FAA automation systems will provide a 'default' fix value for most flights. Missing values are replaced with a false value. By including this indicator, the model is able to differentiate between these filed and default values, and learn the circumstances under which each provides valuable information.

5. Airport Configuration

Airport configuration lists which runways are being used for arrivals and for departures at a specific instant in time. Runways may be in both lists; in this case, they are known as dual-use runways. The distinct combinations of these lists form the set of configurations available for the runway assignment models. Neither in reality, nor in the runway assignment models, do the airport configurations totally constrain which runways may be assigned, however they strongly influence this. This value could be provided by either a data feed that provides live updates (e.g., Digital Automatic Terminal Information Service (D-ATIS)), or a predictive model for airport configurations (e.g., [12]). For model training, the D-ATIS data indicating the actual configuration at the time of the operation was used. This ensures that the model is not biased by errors from another model. In future work, this assumption should be revisited.

Regardless of the source, a custom encoder (in place of one-hotting) is used to translate the airport configuration to values usable by the model. The custom encoder, for an airport with N runways, creates $2N$ columns, one for each combination of either 'arrival' or 'departure' and runway name. We hypothesize that this encoding strategy should help the model learn from similar configurations (e.g., add / remove a single runway) in a way that considering the name alone would not. Missing values are extremely unlikely for this data source, as we assume that the previous configuration continues until a new one is explicitly provided by the data feed. However, should a missing value be encountered (e.g., at the beginning of a time period), those observations are not used to train models.

6. Time since Airport Configuration Changed

As there may also be a relationship between runway assignments and the amount of time that the airport has been operating in the current airport configuration, we include this duration in seconds as a feature in the model. The logic for this feature is that the relationship between configuration and runway assignment policies may be more flexible when a configuration is newly in place, e.g., as a result of aircraft already lined up for specific runways.

7. Time until Estimated Operation

Another feature that may influence the runway assignment policy is the time expected until a flight operates on the runway. For example, there may be less certainty about the policy to be applied when this lookahead is very long. This lookahead is computed as the difference between the time at which the prediction is being made, and an estimated operation time. For departures, this estimated operation time is the Earliest Off Block Time (EOBT) value provided by the airline (if unavailable, other estimates from FAA systems). For arrivals, this value is one of several landing time estimates provided by FAA systems. Which of these estimates to use at each instant is a research question unto itself, and an approach for selecting this 'best' landing time is described in [13].

8. TBFM-assigned Runway

One additional feature included in the arrival runway model is the runway assignment generated by the FAA's Time Based Flow Management (TBFM) system [14]. One of the many features of this decision support system is the capability to predict runway assignments for arriving flights. However, the accuracy of these predictions varies

significantly between TBFM systems used in different terminal areas. Under the proper conditions, this TBFM-assigned runway may be sufficiently accurate to match the performance of an ML model, as it is the product of an expensive and lengthy process of codifying the controllers' runway assignment heuristics into adaptation data. In other situations, this value may simply indicate a flow direction for the airport, without a specific runway identifier. In any case, the value of training runway assignment models for each airport individually allows the ML model to learn the value of this data as a feature and use it accordingly. These data are provided as categories, encoded in the model as one-hot features. Missing values for this feature are imputed by filling with a constant value of UNKN, creating a new category.

D. Building the Dataset

Several steps are required to build a rectangular dataset that an ML algorithm can use to train a model. These steps are primarily mechanical, but are described in the interest of promoting reproducible research. It is critical to recognize that the various features listed in the previous section are available at different instants in time, and are updated at different rates. Through the use of the ATD-2 Fuser, described in the requirements section, the state of each flight is readily available by carrying forward values from previous messages. However, even with this approach, data are only available at the instants at which messages were received from various automation systems. To create a more uniform dataset from which to sample (e.g., to avoid bias induced by certain kinds of flights producing more messages), this approach of carrying values forward was extended further by creating a uniformly-spaced sequence of lookahead (i.e., time until estimated operation) values against which the raw dataset was joined. An example is shown by converting the data in Table 2 to that in Table 3.

Table 2 Sample Data from ATD-2 Fuser

Flight	Lookahead (mins)	Other Features
ABC123	74.3	[set1]
ABC123	55.1	[set2]
ABC123	31.9	[set3]
ABC123	12.8	[set4]

Table 3 Cleaned, Carried-forward Sample Data

Flight	Lookahead (mins)	Other Features
ABC123	74	[set1]
ABC123	73	[set1]
ABC123	...	[set1]
ABC123	55	[set2]
ABC123	54	[set2]
ABC123	...	[set2]
ABC123	31	[set3]
ABC123	30	[set3]
ABC123	...	[set3]
ABC123	12	[set4]
ABC123	11	[set4]
ABC123	...	[set4]
ABC123	0	[set4]

This is clearly a large dataset. Assuming a medium-sized airport with 500 flights/day, a four-hour lookahead period with a one-minute update rate, and six months of training data, there are 21.6 million rows of data available. For some algorithms and computational setups, this volume may introduce difficulties, so lower sampling rates may be necessary than are traditionally employed.

Once a 'uniform' dataset of this nature has been constructed, some rows may be filtered out for having unsuitable data. First, we check that the target values (i.e., runway names) for each flight fall into the set of known runway identifiers for the airport being studied. After exploratory data analysis, we identified several features that seemed unwise to impute. As a result, these features were identified as *core* for training a model and making predictions. Thus, any observation with a missing value for a core feature was not used in model training. Because these rules are logged with the trained model itself, they are applied during real-time operations. Any flight in the real-time environment that fails any of these rules (which would not have been used to train the model in the first place) is assigned a default runway. Based on our analysis and discussion with SMEs, the following features are considered core, and so any rows with a missing value are discarded:

- Planned fix: could be missing if FAA automation systems malfunction, or data is lost

- Airport configuration: could be missing if FAA automation systems malfunction, or data is lost
- Time since airport configuration changed: follows mechanically with airport configuration
- Time until estimated operation: could be missing if all relevant FAA systems are not providing valid data

E. Machine Learning Algorithms

Several ML algorithms have been evaluated thus far in training models for the runway assignment problem. As formulated, this problem is a multinomial classification problem, for which many algorithms currently exist. As will be highlighted in the results, we have thus far used the classic logistic regression [15] available through scikit-learn [16], and the more recently developed and very popular XGBoost [17]. Cross-validation and hyper-parameter tuning for each approach has been evaluated, but results are not included in this paper. Initial analysis of those processes indicated limited improvement in performance metrics as compared to the default parameters available in their implementations.

III. Results and Discussion

Using the features and algorithms described in previous sections, with the Kedro framework [18], we have developed a modeling process that can be easily replicated for any airport for which data are available in the suitable format. A similar series of pipelines for data query, data engineering, and model training / evaluation have been developed for both the arrival and departure runway prediction problems. There is considerable overlap, and code re-use, between the two pipelines. In this section, we first present an overview of the data used in this study, and then a description of various performance metrics of the models trained for various U.S. airports.

A. Data

Appropriate data are essential to conducting research using ML. For this work, we leveraged the Fuser technology previously developed for the ATD-2 project. To generate the datasets used in this research, the Fuser was configured to consume data from the following FAA data feeds: TFM Flight [19], TBFM [14], and STDDS ASDE-X / SMES [20]. The details of each of these data feeds is beyond the scope of this paper, but they provide comprehensive detail about the planned and actual operation of each flight in the U.S. from gate to gate.

The results presented here are based on models trained and evaluated on data from April 25, 2020 through December 31, 2020. Because of the large size of this dataset, as illustrated in the previous section, just 5% of the dataset is used for training, and 5% for evaluation. These values are approximate because the 5% value for training actually represents the fraction of *individual flights*, rather than rows. If we were to sample only rows without considering the panel nature of the data, then rows for the same flight might end up in both the training and evaluation samples. After removing the training sample, exactly 5% of the remaining rows are selected as the evaluation dataset. Note that both datasets contain a variety of lookahead values, and both are sampled randomly.

It is essential to acknowledge the role of the pandemic in the data used for this research. Flight operations were reduced by an enormous amount at the start of the pandemic, and this is reflected in the data. Were these models simply trained for the purposes of writing this paper, then this drastic change in the data would present few problems, as we would simply use pre-pandemic (e.g., 2018-2019) data to represent *normal* levels of air traffic. However, this step change in the data creates challenging problems *deploying* a model trained on pre-pandemic data, and achieving comparable performance using live data. Deploying trained ML models to operate as part of a shadow system for the ATD-2 project was a key requirement of this research, as described earlier. As a result, many of the results presented in this section rely on models trained using data from during the pandemic. This was done to maximize the likelihood that when the models were deployed (still during reduced traffic levels) the conditions used in training the models would be similar to those present.

B. Arrival Runway Assignment Models

Arrival runway assignment prediction models have been trained for a variety of airports. An initial summary of model accuracy is shown in Table 4. These results reflect the performance of the model trained using XGBoost on the evaluation sample, using the features listed in previous sections. Note that these accuracy metrics include observations that are sampled from a variety of lookahead values.

This initial summary of model performance demonstrates that the models are performing at a reasonable accuracy level, given the relative complexity of each airport (e.g., KDFW has more runways used for arrivals than KEWR, so we

Table 4 Arrival Runway Accuracy Metrics for Various Airports

Airport	XGBoost Classifier Accuracy
KDFW	0.618
KDAL	0.726
KCLT	0.777
KIAH	0.699
KJFK	0.765
KEWR	0.939
KLGA	0.973
KPHL	0.791
KBOS	0.915

should expect more uncertainty). However, these results should also be evaluated in a variety of other dimensions, as outlined below.

1. Performance of Different Algorithms

Models were trained using Logistic Regression for some airports. The performance of these models is compared against the XGBoost models in Table 5. The same training and evaluation samples were used for each algorithm.

Table 5 Comparison of Different Algorithms for Arrival Runway Prediction

Airport	XGBoost Classifier Accuracy	Logistic Regression Accuracy
KDFW	0.618	0.375
KDAL	0.726	0.495
KCLT	0.777	0.266
KIAH	0.699	0.250
KJFK	0.765	0.527
KEWR	0.939	0.567
KPHL	0.791	0.431

From these results, it is clear that the XGBoost models perform significantly better than those trained using Logistic Regression. Although there may be interesting explanations and insight related to the relative performance of these algorithms, to help achieve the objectives of this research, we are satisfied to identify the superior performance of the XGBoost models and use them going forward.

2. Comparison to Expert-Driven approach

As described in the introduction, the legacy ATD-2 system used at several airports has decision trees for predicting runway assignments developed through data analysis and interviews with SMEs. In Table 6, we compare the accuracy of the XGBoost models with that of the expert-driven models. In this comparison, the accuracy metrics for the XGBoost models presented earlier are repeated, but the metrics for the legacy ATD-2 systems are sampled at the fix crossing event, where the accuracy should be highest.

There are two interesting trends in this comparison between the ML model results and the expert-driven models. First, the performance of the expert-driven model at KCLT exceeds that of the ML model, reflecting how well-tuned the legacy ATD-2 system is at that facility. In contrast, for the Dallas-area airports, the ML model is able to achieve superior performance. The important difference between KCLT and KDFW is that, during the pandemic, KCLT mostly continued operating in the same fashion (albeit with reduced traffic) while at KDFW, several operational changes were

Table 6 Comparison of ML and Expert-Driven Approaches for Arrival Runway Prediction

Airport	XGBoost Classifier Accuracy	ATD-2 Decision Tree Accuracy
KDFW	0.618	0.524
KDAL	0.726	0.578
KCLT	0.777	0.911

implemented (e.g., arrival runway closed for maintenance). The decision trees in the legacy system were not updated to reflect these operational changes. This highlights the advantage of using an ML approach that is early to update relative to SME informed decision trees.

3. Confusion Matrices

The ways in which each model might make incorrect predictions is also of interest. One way to evaluate these incorrect predictions is through the use of a confusion matrix. In Figure 1, the relatively simply confusion matrix for KDAL is shown. Rather than show counts (which would be large numbers) the fraction of the evaluation dataset in each cell is shown as a percentage. Warmer colors (e.g., yellow) indicate a larger fraction of the dataset, while cooler colors (e.g., blue) indicate a smaller fraction. It is clear that the bulk of the observations fall on the diagonal, in which the model correctly predicts the arrival runway. However, an interesting effect for KDAL is that there is a significant amount of incorrect predictions on a parallel runway (e.g., predict 13L, land 13R). In some sense, these incorrect predictions are not as bad as those for which the "flow" direction of the airport is incorrect (e.g., 13 vs 31).

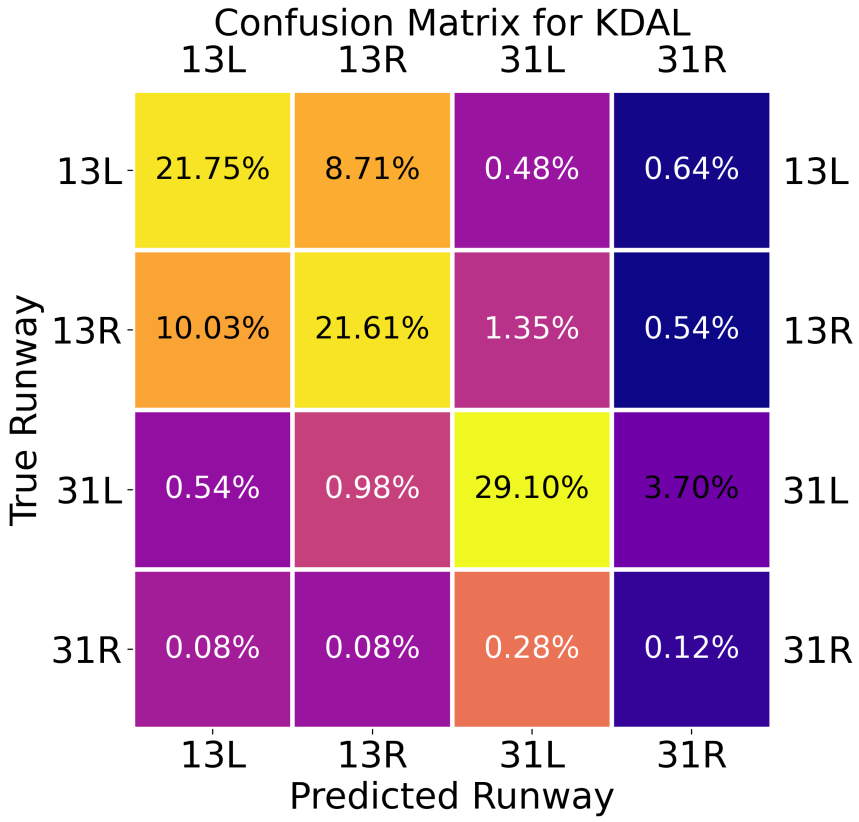

Fig. 1 KDAL Confusion Matrix

In Figure 2, the confusion matrix for KCLT is shown. This matrix exhibits a sort of block diagonal structure, with blocks for the two directions of the primary group of runways (i.e., 18 and 36). In line with the high accuracy for KCLT, the cells with the highest fraction of observations are along the true diagonal.

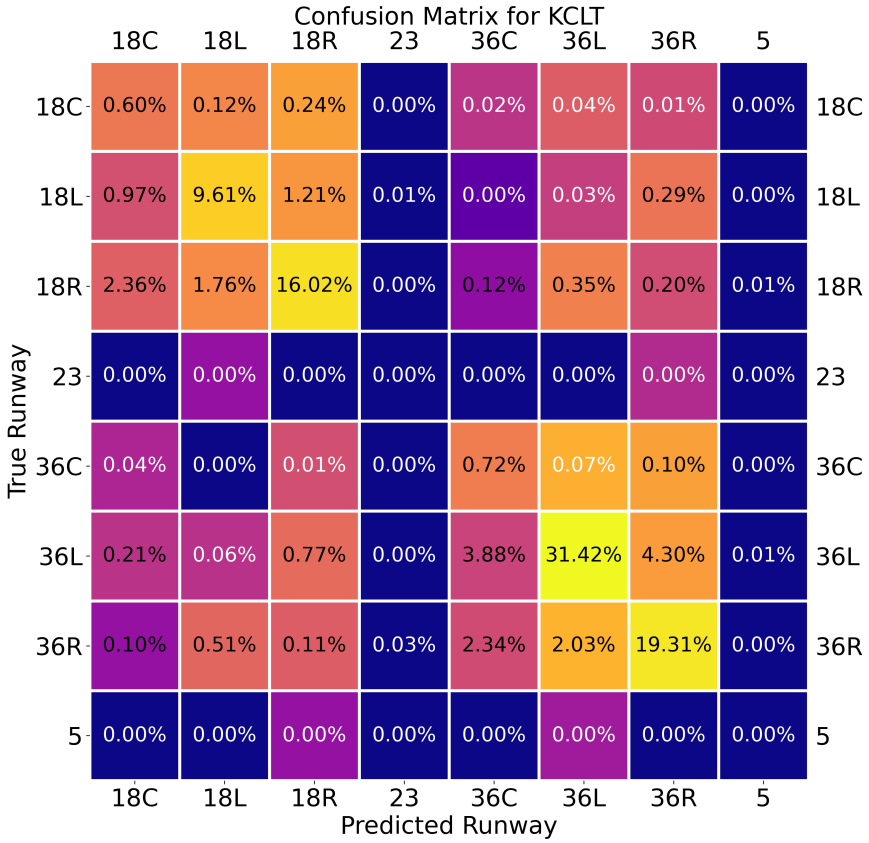

Fig. 2 KCLT Confusion Matrix

We have generalized this important notion about an incorrect prediction to a parallel runway being *less bad* than an incorrect prediction to a non-parallel runway. Table 7 shows the fraction of rows from the evaluation dataset for which the prediction was incorrect, but for which the numeric portion of the runway identifier (i.e., the direction) matched the true runway used. Only airports with some parallel runways are included in these data.

Table 7 Incorrect Prediction to Parallel Runways

Airport	Fraction Observations Incorrect, but on Parallel Runway
KDFW	0.266
KDAL	0.227
KCLT	0.194
KIAH	0.229
KJFK	0.167
KEWR	0.041
KPHL	0.095
KBOS	0.012

From these data, it is clear that some airports have a more flexible utilization strategy for parallel runways than other

airports (e.g., KDFW vs KEWR). In future work, we hope to improve these incorrect predictions to parallel runways by identifying features that may indicate such a balancing strategy is in use, and leverage the "less bad" nature of this incorrect prediction in the model training itself.

4. Evolution of Predictions over Time

In previous analysis sections, results have been evaluated together across all lookahead times. In some ways, this is quite a fair evaluation strategy, because much of the data used in making these predictions is static (excepting the TBFM-assigned runway). However, it is also important to acknowledge and evaluate the dynamic nature of these models.

To that end, Figures 3 and 4 depict the accuracy of models for KCLT and KJFK over time, as flights approach and land at the airport. These two airports make an interesting contrast, as the accuracy for KJFK is relatively constant, while the accuracy for KCLT is steadily increasing. This likely reflects differences in the flexibility of each airport to make changes to arrival runway assignment planning as flights progress.

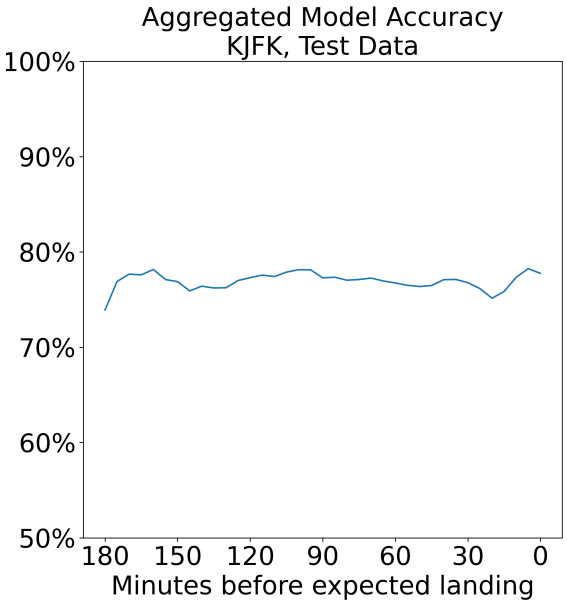

Fig. 3 KJFK Model Accuracy over Time

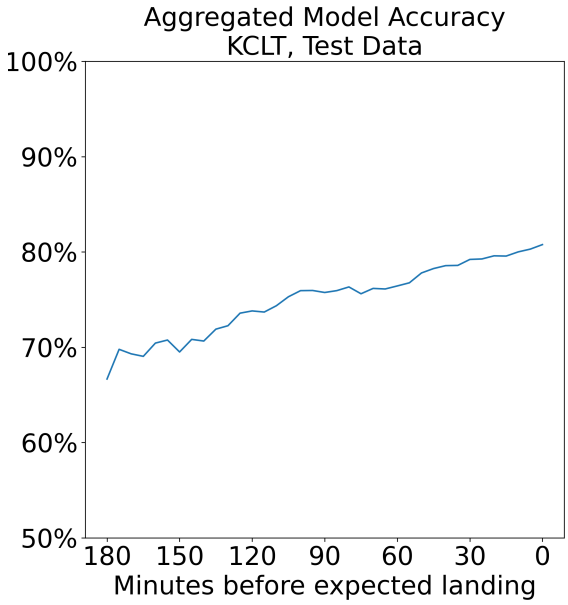

Fig. 4 KCLT Model Accuracy over Time

C. Departure Runway Assignment Models

Departure runway assignment prediction models have been trained for a variety of airports. An initial summary of model accuracy is shown Table 8. These results reflect the performance of the model trained using XGBoost on the evaluation sample, using the features listed in previous sections. Note that these accuracy metrics include observations that are sampled from a variety of lookahead values. The accuracy levels are slightly higher for these departure runway models than for the arrival runway models, indicating that the decision heuristic employed by the controllers is better able to be captured (e.g., is more consistent) than for the arrival runway prediction problem. However, these results should also be evaluated in a variety of other dimensions, as outlined below.

1. Performance of Different Algorithms

Models were trained using Logistic Regression for some airports. The performance of these models is compared against the XGBoost models in Table 9. The same training and evaluation samples were used for each algorithm.

From these results, it is clear that the XGBoost models perform significantly better than those trained using Logistic Regression. For the purposes of this research, we are not concerned about the root cause of this differential, just the trend that XGBoost seems to produce better-performing models. Some preliminary results indicated that there was potential for tuning the hyperparameters of the Logistic Regression models to achieve better performance, but still not at

Table 8 Departure Runway Accuracy Metrics for Various Airports

Airport	XGBoost Classifier Accuracy
KDFW	0.821
KDAL	0.813
KCLT	0.886
KIAH	0.797
KJFK	0.932
KEWR	0.971
KLGA	0.977
KPHL	0.902
KBOS	0.894

Table 9 Comparison of Different Algorithms for Departure Runway Prediction

Airport	XGBoost Classifier Accuracy	Logistic Regression Accuracy
KDFW	0.821	0.548
KDAL	0.813	0.839
KCLT	0.886	0.314
KIAH	0.797	0.657
KJFK	0.932	0.497
KEWR	0.971	0.583

a level comparable with even the stock implementation of the XGBoost classifier.

2. Comparison to Expert-Driven approach

As described in the introduction, the legacy ATD-2 system used at several airports has decision trees for predicting runway assignments developed through data analysis and interviews with SMEs. In Table 10, we compare the accuracy of the XGBoost models with that of the expert-driven models. In this comparison, the accuracy metrics for the XGBoost models presented earlier are repeated, but the metrics for the legacy ATD-2 systems are sampled at the pushback event, where the accuracy should be highest.

Table 10 Comparison of ML and Expert-Driven Approaches for Departure Runway Prediction

Airport	XGBoost Classifier Accuracy	ATD-2 Decision Tree Accuracy
KDFW	0.821	0.828
KDAL	0.813	0.654
KCLT	0.886	0.950

There are two interesting trends in this comparison between the ML model results and the expert-driven models. First, the performance of the expert-driven model at KCLT exceeds that of the ML model, reflecting how well-tuned the legacy ATD-2 system is at that facility. In contrast, for the Dallas-area airports, the ML model is able to achieve superior performance. The important difference between KCLT and KDFW is that, during the pandemic, KCLT mostly continued operating in the same fashion (albeit with reduced traffic) while at KDFW, several operational changes were implemented. These operational changes did not result in updates to the decision trees used in the legacy system.

3. Confusion Matrices

In Figure 5, the relatively simply confusion matrix for KLGA is shown, and in Figure 6, the confusion matrix for KDFW is shown.

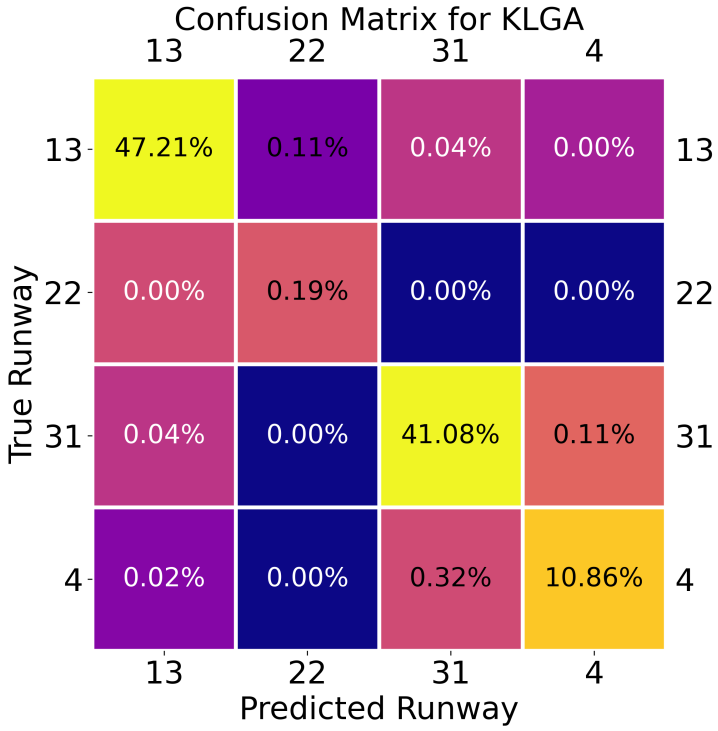

Fig. 5 KLGA Confusion Matrix

Again the fraction of the evaluation dataset in each cell is shown as a percentage. It is clear that the bulk of the observations fall on the diagonal for each airport, for which the model correctly predicting the departure runway.

From this confusion matrix, is is also clear that the model does face some confusion about the use of the diagonal runways at KDFW (i.e., the 13 and 31 runways). There are also several off-diagonal blocks corresponding to incorrect predictions at KDFW when these runways were (or were not) expected to be used. This is clearly an area where some additional features may improve model performance, since there is clearly some strategy in the controllers' runway assignment decisions to use those diagonal runways (e.g., GA versus airline flights, parking stand location).

We have generalized this important notion about an incorrect prediction to a parallel runway being *less bad* than an incorrect prediction to a non-parallel runway. Table 11 shows the fraction of rows from the evaluation dataset for which the prediction was incorrect, but for which the numeric portion of the runway identifier (i.e., the direction) matched the true runway used. Only airports with some parallel runways are included in these data.

From these data, two trends are clear. First, the Dallas-area airports continue to exhibit greater flexibility in runway utilization, as was observed for arrivals. Second, and perhaps more importantly, the value of this metric across all airports is lower than it was for arrivals. This suggests that departure runway assignments are more predictable (as reflected in the higher accuracy metrics) than arrival runway assignments. This is an important finding, because the primary focus of the experiments being conducted in the ATD-2 project is planning runway utilization for departures.

4. Evolution of Predictions over Time

In previous analysis sections, results have been evaluated together across all lookahead times. In some ways, this is quite a fair evaluation strategy, because much of the data used in making these predictions is static (excepting the TBFM-assigned runway). However, it is also important to acknowledge and evaluate the dynamic nature of these models.

To that end, Figures 7 and 8 depict the accuracy of models for KDAL and KIAH over time, as flights approach and land at the airport. These two airports are simply examples of consistent behavior we observe across airports: there are

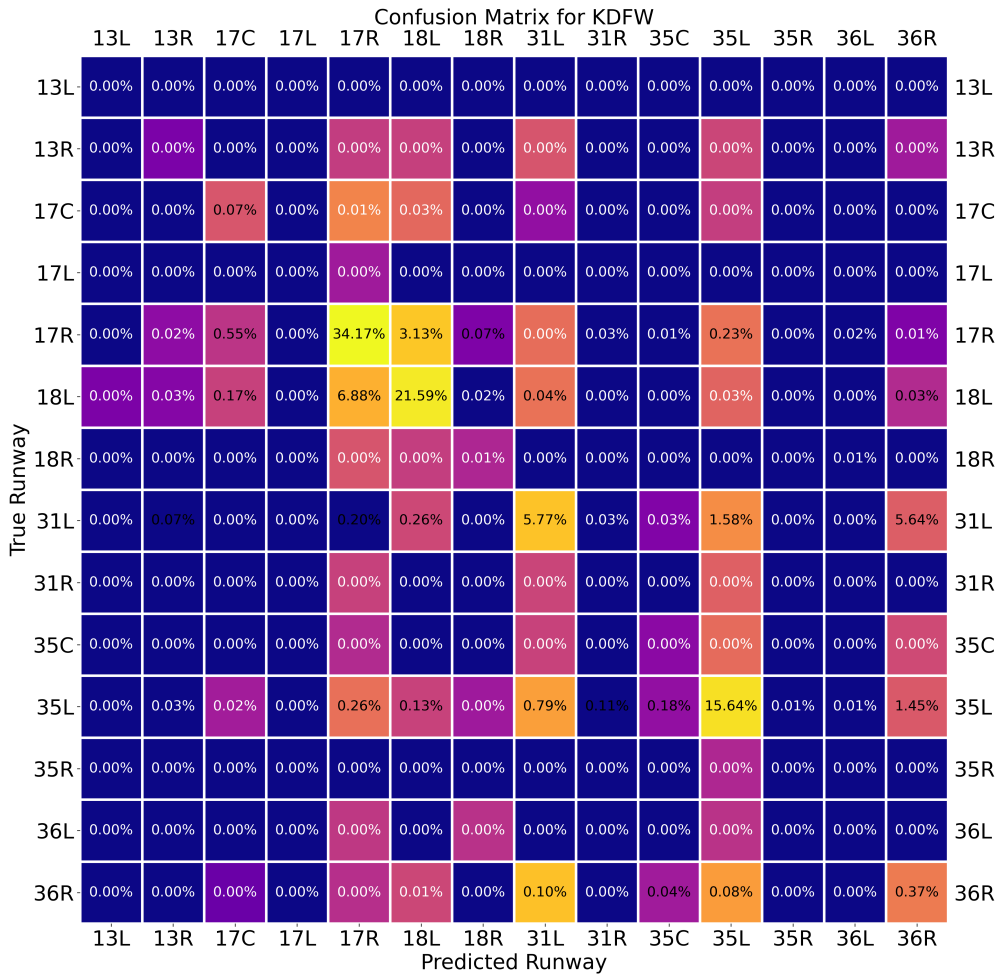

Fig. 6 KDFW Confusion Matrix

Table 11 Incorrect Prediction to Parallel Runways

Airport	Fraction Observations Incorrect, but on Parallel Runway
KDFW	0.121
KDAL	0.170
KCLT	0.081
KIAH	0.034
KJFK	0.017
KEWR	0.014
KPHL	0.071
KBOS	0.018

relatively few changes in accuracy leading up to the pushback event, because there are relatively few changes in the features input to the model leading up to pushback.

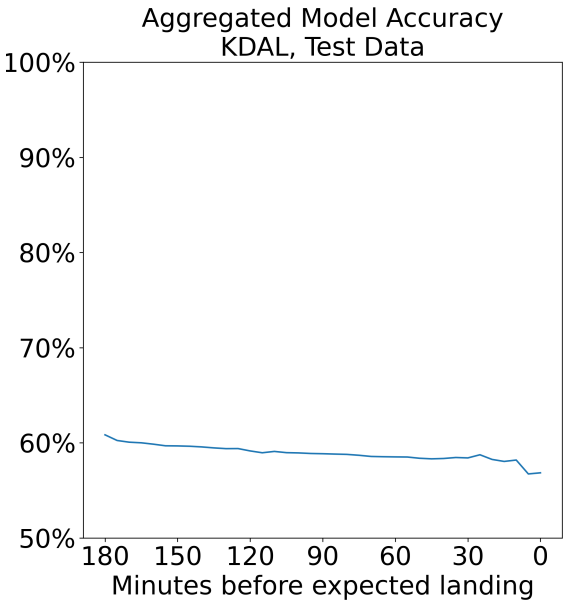

Fig. 7 KDAL Model Accuracy over Time

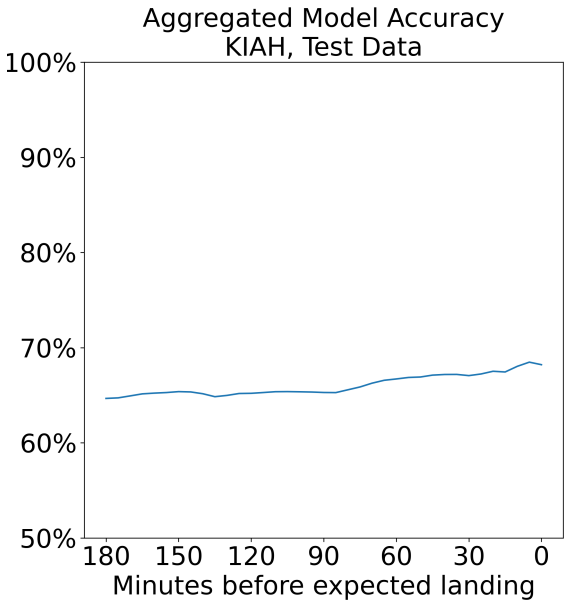

Fig. 8 KIAH Model Accuracy over Time

5. Use of Different Time Periods

Finally, we present results demonstrating that our use of data from 2020 (during the pandemic) yielded models of similar quality (in aggregate) to models trained on data from earlier periods. To make this comparison, we trained a model on the same time period (April 25 through December 31) from 2019 for KDFW. This airport was selected for comparison because the degradation of ATD-2 decision tree model accuracy apparent in Tables 6 and 10 suggested operational changes. The data in Table 12 compares the performance of the two models trained on different time periods, and evaluated using samples taken from their "own" year of data. Additional classification metrics besides accuracy (as shown previously) are presented, including precision, recall, and area under ROC curve (AUC).

Table 12 Comparison of Departure Runway Models from 2019 and 2020

Metric	2020 Model	2019 Model
Accuracy	0.821	0.851
Misclassification to parallel runway	0.121	0.111
Precision	0.824	0.841
Recall	0.821	0.851
AUC	0.921	0.913

The results from this comparison suggest that the XGBoost algorithm is able to produce a model of equivalent quality using either data from a "normal" time period, or from the pandemic time period. Or, put another way, the problem of assigning flights to departure runways was equally predictable with the same features during each distinct period, even if those relationships may have changed somewhat.

IV. Conclusion and Ongoing Work

In this paper, we have described our work training ML models to predict arrival and departure runway assignments. This work shows initial promise for learning the heuristics used by controllers to assign flights to runways when landing or departing. Models have relatively high accuracy, likely high enough to support the use cases for which they are being evaluated on the ATD-2 project. The overall approach will enable broad deployment across a wide variety of U.S. airports using a standardized approach and dataset. In concert with models to predict other aspects of NAS operations,

we believe this data-driven machine learning approach will enable rapid testing and deployment of advanced prediction and decision-support tools.

References

[1] Isaacson, D., Davis, T., and Robinson, III, J., "Knowledge-based Runway Assignment for Arrival Aircraft in the Terminal Area," *Guidance, Navigation, and Control Conference (GNC)*, 1997. https://doi.org/10.2514/6.1997-3543.

[2] National Aeronautics and Astronautics Administration, "Airspace Technology Demonstration 2 (ATD-2): Integrated Arrival/Departure/Surface (IADS) Traffic Management," , 2020. URL https://aviationsystems.arc.nasa.gov/research/atd2/index.shtml.

[3] Eisinga, K., "Predicting Runway Allocation with Support Vector Machine and Logistic Regression," Master's thesis, Tilburg University, July 2016.

[4] Nakamura, Y., Mori, R., Aoyama, H., and Jung, H., "Modeling of Runway Assignment Strategy by Human Controllers Using Machine Learning," *2017 IEEE/AIAA 36th Digital Avionics Systems Conference (DASC)*, 2017, pp. 1–7. https://doi.org/10.1109/DASC.2017.8102099.

[5] Berge, M. E., Haraldsdottir, A., and Scharl, J., "The Multiple Runway Planner (MRP): Modeling and Analysis for Arrival Planning," *2006 IEEE/AIAA 25th Digital Avionics Systems Conference (DASC)*, 2006, pp. 1–11. https://doi.org/10.1109/DASC.2006.313684.

[6] Lohr, G., Brown, S., Atkins, S., Eisenhawer, S., Bott, T., Long, D., and Hasan, S., "Progress Toward Future Runway Management," *11th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference*, 2011. https://doi.org/10.2514/6.2011-6925.

[7] Heblij, S., and Wijnen, R., "Development of a Runway Allocation Optimisation Model for Airport Strategic Planning," *Transportation Planning and Technology*, Vol. 31, No. 2, 2008, pp. 201–214. https://doi.org/10.1080/03081060801948191.

[8] Gorman, S. M., Burke, J. M., Robeson, I. J., and Phipps, B. S., "Fuser Deeper Dive (Mediation & Use Cases)," 2019.

[9] Federal Aviation Administration, "National Flight Data Center 28 Day NASR Subscription," online, 2020. URL https://www.faa.gov/air_traffic/flight_info/aeronav/aero_data/NASR_Subscription/.

[10] Kille, R., "Summary of the July 1999 Internal Departure Delay Investigation for ZDV Along with New Departure Delay Data From ZOB, ZID, ZFW and ZDV," presentation, 2004.

[11] Federal Aviation Administration, "Wake Turbulence Recategorization," , February 2016. URL https://www.faa.gov/documentLibrary/media/Order/JO_7110_659C.pdf.

[12] Khater, S., Rebollo, J., and Coupe, W., "A Recursive Multi-step Machine Learning Approach for Airport Configuration Prediction," *Submitted to AIAA Aviation Forum*, 2021.

[13] Wesely, D., Churchill, A., Slough, J., and Coupe, W., "A Machine Learning Approach to Predict Aircraft Landing Times using Mediated Predictions from Existing Systems," *Submitted to AIAA Aviation Forum*, 2021.

[14] Federal Aviation Administration, "Time Based Flow Management," , August 2020. URL https://www.faa.gov/nextgen/cip/tbfm/.

[15] Berkson, J., "Application of the Logistic Function to Bio-Assay," *Journal of the American Statistical Association*, Vol. 39, No. 227, 1944, pp. 357–365.

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

[17] Chen, T., and Guestrin, C., "XGBoost: A Scalable Tree Boosting System," *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Association for Computing Machinery, New York, NY, USA, 2016, p. 785–794. https://doi.org/10.1145/2939672.2939785.

[18] Bălan, L., (Kiyo), K. K., Deriabin, D., Hoang, L., Ivaniuk, A., Dada, Y., Datta, D., Patel, Z., Wrigley, G., Danov, I., Stichbury, J., Khan, N., Tsaousis, N., Theisen, M., Walker, W., Nguyen, T., Westenra, R., Carvalho, L., Trevisani, M. D., Bertoli, S., Mawjee, S., sasaki takeru, Nijholt, B., Vukolov, D., Fischer, K., Vijaykumar, Minami, Y., bru5, and dr3s, "quantumblacklabs/kedro: 0.17.0," , Dec. 2020. https://doi.org/10.5281/zenodo.4336685, URL https://doi.org/10.5281/zenodo.4336685.

[19] Federal Aviation Administration, "TFMData Service," , ????. URL https://cdm.fly.faa.gov/?page_id=2288.

[20] Federal Aviation Administration, "SWIM Terminal Data Distribution System (STDDS)," , ????. URL https://www.faa.gov/air_traffic/technology/swim/stdds/.