Predicting Runway Configurations and Arrival and Departure Rates at Airports: Comparing the Accuracy of Multiple Machine Learning Models

Ramakrishna Raju
KBR Inc.
Cambridge, MA, USA
ramakrishna.raju@us.kbr.com

Rohit Mital
KBR, Inc.
Colorado Springs, USA
rohit.mital@us.kbr.com

Bruce Wilson
Volpe National Transportation Systems Center
Cambridge, MA, USA
bruce.wilson@dot.gov

Kamala Shetty
Federal Aviation Administration
Washington, D.C., USA
Kamala.Shetty@faa.gov

Michael Albert
Federal Aviation Administration
Washington, D.C., USA
Michael.J.Albert@faa.gov

***Abstract*—In response to the needs of various stakeholders, the FAA has developed an automated tool that provides values for expected next day airport runway configurations and their respective arrival and departure rates. The arrival and departure rates at an airport define the capacity of an airport and as such are a critical piece in determining whether not a capacity/demand imbalance is expected. A forecast of these imbalances facilitates next-day planning and provides to stakeholders transparency regarding potential traffic constraints. The use case that motivated the work described in this paper is to develop various machine-learning (ML) models and compare their performance against the current automated tool, which finds operational periods with similar facility (airport) weather and uses the most frequently occurring runway configuration and rates as the capacity forecast for the next day. Extensive model development across five airports and four years of operational data and forecasted weather showed that the ML models consistently had higher accuracy than the automated tool. Among the ML models, superior options were found, but a single best model across all airports and predictions was not. In a follow-on effort, the authors hope to explore opportunities for working with the user community, incorporate their feedback, and determine if one or more ML models can be used in practice.**

Keywords—machine learning, runway configuration, airport arrival rate, airport departure rate

I. Introduction

The Federal Aviation Administration's Air Traffic Control System Command Center (ATCSCC) provides air traffic flow management (ATFM) for the National Air Space System (NAS) [1]. It supports air traffic control, helping optimize traffic flows during periods of time when demand exceeds, or is expected to exceed, the capacity of any given NAS resource. Effective ATFM contributes to a safe and efficient airspace by managing the workload of air traffic controllers and reducing delays, the latter of which are costly to both airlines and the public. In pursuit of safe and efficient airspace, the ATCSCC collaborates with Air Route Traffic Control Centers, Terminal Radar Approach Control Facilities, Air Traffic Control Towers, and aviation industry partners to identify constraints, formulate and communicate mitigation strategies, and facilitate action among all the stakeholders. These strategies include the use of Traffic Management Initiatives (TMIs) such as reroutes, Ground Delay Programs, Ground Stops, and Airspace Flow Programs.

ATFM occurs at different time horizons, ranging from the strategic to the tactical [2]. In recent years, the FAA's Air Traffic Organization has focused on improving the pre-tactical phase, including the use of more data-driven analysis that utilizes data from past operations [3]. A driving component of this effort has been the implementation of a planning process called PERTI (Plan, Execute, Review, Train and Improve) with a goal of using lessons learned from past performance to help drive the timing and selection of ATCSCC interventions that include TMIs. Under PERTI, the Advanced Planning team at the ATCSCC is charged with the "Plan" portion of the process; the goal is to continually increase the quality and lead time for airspace constraint planning by utilizing historical performance data for similar days and identifying successful strategies. This group hosts the daily advanced planning webinar at 1430 (Eastern Time), during which an operations plan for the upcoming day is communicated with stakeholders and input on weather, constraints, and goals are gathered. An update to the plan is developed by 1700 and distributed to stakeholders and an additional planning webinar occurs at 2115.

Collaborative Decision Making (CDM) is a joint government/industry initiative that pools expertise within the FAA, general aviation, airlines, private industry, and academia to provide solutions to current ATFM challenges [4]. In 2019, under the CDM umbrella, the Stakeholder Engagement Team (SET) was established to provide recommendations to the PERTI group within the ATCSCC to enhance and expand the Advanced Planning process. One such recommendation was the development of an automated tool available to all stakeholders

that provides values for expected next day airport runway configurations and their respective arrival and departure rates. The arrival and departure rates at an airport define the capacity of an airport and as such are a critical piece in determining whether not a capacity/demand imbalance is expected. This data-driven decision support tool should improve the quality of the Advanced Planning process by reducing reliance on the operational experience of the individual planner and by providing greater transparency regarding potential constraints to stakeholders.

To help achieve the objectives of the SET the FAA's Office of Performance Analysis has built a tool that provides historical runway configuration and rate information to help inform same-day and next-day planning. Given a set of terminal area weather conditions, coming either from a Terminal Area Forecast (TAF) or from a user-specified set of conditions, the tool returns information on the historical distribution of configurations and rates under similar weather conditions. Additionally, the tool has a module that will flag expected periods of demand/capacity imbalances, where the expected capacity comes from historical data mapped to the current TAF and expected demand comes from historical TFMS and current schedule data.

The tool developed by the Office of Performance Analysis employs logic ("business rules") to match a given weather forecast with observed weather at the same airport. The matching instances of observed weather have mode values (most common) for runway configuration, airport arrival rate (AAR), and airport departure rate (ADR). For the business rules tool, these mode values become the prediction for the airport's configuration and capacity during the upcoming operational day.

The FAA wished to examine if artificial intelligence or machine learning (ML) could improve upon the business rules performance for predicting configurations and capacity. To this end, the Office of Performance Analysis requested that we develop one or more ML models and compare its predictions to those of the business rules. If the application of ML models were found to yield better prediction accuracy, a secondary objective of the study was to identify which type of ML models might yield the best results for this particular use case.

II. Source Data and Business Rules Development

This paper compares the performance of a set of predictions obtained using business rules against the same set obtained using AI/ML models. The business rules and models were both developed using the same set of TAFs, airport configurations, and airport rates. This section begins with a description of the TAFs and airport data. A description of the business-rule development follows.

A. Model Development Data

The source for all operational data used in the tool described in the introduction and in this research is the FAA's publicly available Aviation System Performance Metrics system [5]. TAF data in general originate from NOAA and were collected from the Aviation Weather Center's data server [6] and from a third party site called Ogimet [7].

TABLE I. Filter Criteria for Determining Similar Weather in Existing Configuration and Rate Tool

Weather Element	Filter Criteria
Wind direction	± 30° of TAF value
Wind speed	± 5 knots of TAF value
Wind gust	± 5 knots of TAF value for wind gust or wind speed
Visibility	same <1, 1-3, 3-5, >5 statute mile interval as TAF value
Ceiling	same <500, 500-1000, 1000-3000, >3000 feet interval as TAF value
Meteorological Condition	same category as TAF value (thunderstorm, snow/ice, rain, obscuration, other, none)
Hour	same as those in the forecast period

The same business rules that underlie the Office of Performance Analysis tool were used to 'predict' configurations and rates and to serve as a baseline against which the performance of the machine learning models could in part be evaluated. The tool (hereafter the 'business rules approach') uses TAFs from the National Weather Service's Aviation Weather Center for a selected airport and decodes the raw TAF into its hourly forecast for wind, wind gust, ceiling, visibility, and meteorological condition. For each forecasted hour, four years of historical terminal weather, airport configuration, and capacity rates are filtered down to a set of historical hours similar to the hour's forecasted weather. The business rules approach returns the most frequently used runway configuration and how often it occurs within the filtered dataset. It also returns the range of rates (15^{th}–85^{th} percentile) found in the filtered dataset that are associated with the most frequently used runway configuration.

What constitutes similar weather is determined by the intersection of a number of filters on the individual weather elements of the historical data as listed in Table I. These business rules were developed using the input of air traffic control subject matter experts. The binning of the visibility and ceiling values specifically were derived from Aviation Weather Service's METAR board, an established resource for the ATCSCC, which color codes forecast hours according to the weather's potential impact on air traffic operations at the airport [8]. Some iterative, but non exhaustive, analysis was performed to determine the wind direction, wind speed, and wind gust filter windows that would result in higher predictive power in the results.

Computationally inexpensive, this simple, business rule-based filtering approach is easily understood by the end users. However, accuracy, measured as the percentage of time the ruleset returned an airport configuration, arrival rate range, and departure rate range that matched the realized values, varies significantly among airports and in some cases is too low compared to an advanced planner's estimate. The simplicity of this approach is not able to fully model the real world complexities that drive configuration and rate selection, such as the local operating characteristics of different airports, the interactions between the weather elements, or hysteresis in the changes in configuration in response to changing conditions.

III. ML Model Development

In addition to the business rule-based filtering approach, Machine Learning (ML) models were developed to predict Runway Configurations, AAR and ADR. The Machine

Learning models were trained on the TAF data set and once trained, were used to make predictions. Described in the sequel, four types of ML models were developed so that we could evaluate the performance and accuracy of the models relative to the business rules and to one another.

A. Data Preparation

ASPM and TAF Data for five airports (ATL, BOS, EWR, JFK and LGA) spanning across a four-year timeframe (February 08, 2017 – February 08, 2021) was used for training and testing the models. Data for each airport was contained in a separate file. The ASPM sample size was ~35,000 for each airport and the average TAF data sample size was ~480,000 (the sample size varies a little for each airport based on the number of forecasts).

First, the two data sets were merged into one data set using the field called GMTDATE which exists in both the data sets.

The next step was to identify and mitigate data quality issues, perform data transformations, convert text data into numeric values, etc. Common data quality issues included missing arrival or departure runways, runway configurations reported as a closed airport (such records were deleted), null values for AAR or ADR, and null values for weather type (which actually denotes no precipitation).

B. Feature Engineering

Feature engineering was applied to the data set to prepare it for the ML model training. A feature is a property on which analysis or prediction is to be done. The features are selected initially based on the domain knowledge and understanding of the data – identifying those input features which were likely to impact the outcomes of the predictive model. In addition to all the TAF inputs such as Wind Angle, Wind Speed, Visibility, Cloud Ceiling, etc., other features were identified from the data set. Based on the real-world domain knowledge of the system, these latter features were initially thought to be relevant and important to model outcomes .

TABLE II. ML MODEL FEATURES

Feature	How Created	Feature Type
Year	Extracted from GMTDATE	Numerical
Month	Extracted from GMTDATE	Categorical
Week	Extracted from GMTDATE	Numerical
Day of Week	Extracted from GMTDATE	Numerical
Hour	Extracted from GMTDATE	Numerical
RWYCONF persistence RWYCONF pred lead time	From RWYCONF. Provides RWYCONF at the time of prediction to be used with the time for which we are predicting	Categorical
AAR persistence/ AAR pred lead time	From AAR. Provides AAR at the time of prediction to be used with the time for which we are predicting	Numerical
ADR persistence/ ADR pred lead time	From ADR. Provides ADR at the time of prediction to be used with the time for which we are predicting	Numerical

As an example, attributes of time – such as year, month, day of the week, etc. normally have an operational effect on runway configurations and arrival / departure rates. Therefore, these properties are extracted from the data, and added as features to the input data set used to train the model. Table II shows other examples of features selected for the ML models.

Another important aspect of the feature engineering is the adjustments needed for the prediction lead time. Models need to trained for three different prediction lead times – 3 hours, 14 hours and 24 hours. These intervals were selected in anticipation of the most common patterns of prediction by the end users of the system.

The example shown in Fig. 1 illustrates feature engineering to predict runway configurations for the Atlanta (ATL) airport for a prediction lead time of 24 hours. In the figure, Row 2 has a date and time value of 3/1/2017 0:00 hours. This time is close to the time at which at which the TAF was issued, referred to here as the *initial time.* Assume that at this time, we wish to predict runway configuration and rates 24 hours into the future, i.e., for 0:00 hours on 3/2/2017 (Row 26). This second time is referred to as the *valid time,* and the 24-hour gap is referred to as the *prediction lead time.* As shown in Fig. 1 the runway configurations for these two times differ, reflecting changes in the airport's configuration over that 24-hour period.

	LOCID	GMTDATE	RWYCONF
2	ATL	3/1/2017 0:00	26R, 27L, 28 | 26L, 27R
3	ATL	3/1/2017 1:00	26R, 27L, 28 | 26L, 27R
4	ATL	3/1/2017 2:00	26R, 27L, 28 | 26L, 27R
5	ATL	3/1/2017 3:00	26R, 27L, 28 | 26L, 27R
6	ATL	3/1/2017 4:00	26R, 27L, 28 | 26L, 27R
7	ATL	3/1/2017 5:00	26R, 27L, 28 | 26L, 27R
8	ATL	3/1/2017 6:00	26R, 27L, 28 | 26L, 27R
9	ATL	3/1/2017 7:00	26R, 27L, 28 | 26L, 27R
10	ATL	3/1/2017 8:00	26R, 27L, 28 | 26L, 27R
11	ATL	3/1/2017 9:00	26R, 27L, 28 | 26L, 27R
12	ATL	3/1/2017 10:00	26R, 27L, 28 | 26L, 27R
13	ATL	3/1/2017 11:00	26R, 27L, 28 | 26L, 27R
14	ATL	3/1/2017 12:00	26R, 27L, 28 | 26L, 27R
15	ATL	3/1/2017 13:00	26R, 27L, 28 | 26L, 27R
16	ATL	3/1/2017 14:00	26R, 27L, 28 | 26L, 27R
17	ATL	3/1/2017 15:00	26R, 27L, 28 | 26L, 27R
18	ATL	3/1/2017 16:00	26R, 27L, 28 | 26L, 27R
19	ATL	3/1/2017 17:00	8L, 9R | 8R, 9L
20	ATL	3/1/2017 18:00	8L, 9R | 8R, 9L
21	ATL	3/1/2017 19:00	8L, 9R | 8R, 9L
22	ATL	3/1/2017 20:00	8L, 9R | 8R, 9L
23	ATL	3/1/2017 21:00	8L, 9R | 8R, 9L
24	ATL	3/1/2017 22:00	8L, 9R | 8R, 9L
25	ATL	3/1/2017 23:00	8L, 9R | 8R, 9L
26	ATL	3/2/2017 0:00	8L, 9R | 8R, 9L

Fig. 1. Runway configurations over 24-hour period

The data shown in Fig. 1 is part of a much larger training data set. To train the model for a 24-hour prediction lead time, a

new input feature needs to be added to each row of data to make the association. This new feature added to row 2, the runway configuration contained in Row 26. Now, because Row 2 and Row 26 have been associated with this new feature, when the ML models are trained against the data, the model can learn how to predict runway configurations 24 hours from the current time. The same principle applies to the 3-hour and 14-hour prediction lead times.

C. Candidate Models

Four models were used in this study - Logistic regression, Random Forest, CatBoost and Neural networks. Each of these is a standard model and a natural fit for this use case.

For prediction of runway configuration AAR/ADR the target variable were treated as discrete values rather than continuous values and therefore classification was applied instead of regression.

There were other models such as SVM and K-NN that are popular in the industry. We chose not to use K-NN as it is generally not good for multiclass classification unless the sample size is very high (50k or more per class). Also it is very sensitive to outliers such as unusual weather conditions. Tree-based algorithms such as Random Forest can handle multiclass classification as well as imbalanced dataset, hence they were considered a better fit for this use case than K-NN models. SVMs are generally used when the feature set is huge and the number of rows are lesser in comparison. Given the dataset we have, it is also imbalanced where a certain category is present much more than others.

The models were evaluated using three metrics accuracy, precision, and recall. However, this paper focuses primarily on the metric of accuracy, here defined as percentage of correct predictions for the test data.

To evaluate the models, the accuracy of each model was compared with the accuracy of three other models (a) A no-skill model that uses majority predictions, (b) A persistence model that predicts the target variable to be the same as they are at the time of prediction (i.e. the model assumes that values of runway configurations, AAR and ADR will persist) and a (c) a business rules model that applies the business rules to the data set to make the predictions. These three values could be evaluated against the ML model prediction to see how the ML models performed.

A **Random Forest Model** is an ensemble of Random Decision Tree Models [9] [10]. The ensemble is run during training time and the aggregation of the ensemble is used to classify the results. The key advantage of a Random Forest model is its higher accuracy (relative to using simple decision trees), but running the ensemble algorithm requires more time for training. A highly simplified overview of how the Random Forest model is applied to predict Runway Configurations is shown in Fig. 2, which illustrates how a Random Forest model would process TAF features of wind angle and visibility.

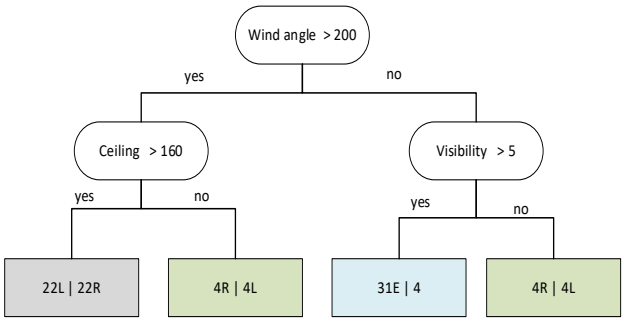

Fig. 2. Processing of TAF features in a random-decision-tree model

Logistic Regression models calculate the probability of each configuration by using a mathematical function called the sigmoid function. Each configuration is assigned a probability value between 0.0 and 1.0. This algorithm is popular because it is simple and fast, and yields accuracy which is comparable to more heavy-duty models.

CatBoost is an open-source gradient boosting framework similar to Random Forest [11]. Trees are trained sequentially whereas Random Forest trees are trained in parallel. The CatBoost algorithm is specially designed to handle categorical variables more efficiently and intelligently than traditional Random Forest models, and this leads to more predictive value in data sets which are very categorical in nature. CatBoost gained popularity after many competition winning solutions on Kaggle where its accuracy was higher than other models.

Neural Networks(NN) or Artificial Neural Networks (ANN) [12] are a part of machine learning architectures that mimic how a human brain works. They are different from traditional machine learning algorithms such as Random Forest models. In traditional machine learning, the algorithm is given a set of relevant features to analyze, however, in deep neural networks, the algorithm is given raw data and derives the features itself. There are many neural network architectures available. We chose Recurrent Neural Network as the architecture for using deep learning neural network in predicting runway configuration, arrival and departure rates.. A recurrent neural network (RNN) is a class of artificial neural networks where connections between nodes form a directed graph along a temporal sequence. This allows it to exhibit temporal dynamic behavior.

D. Model Training

After data preparation and feature engineering, the data was split into a training data set and testing data set and the models were fitted and trained. There was one model for each combination of the 5 airports, 4 ML model types and 3 Prediction Lead times.

E. Model Validation

Model validation to assess accuracy was done using two different methods – Random Sampling and Walk-Forward Validation. In random sampling, the test samples are drawn at random from the entire data set. A 70-30 split was followed and therefore 30% of the data was randomly sampled and separated as the testing data set for model validation. While random

sampling is a popular method for validating models, one limitation in this use case is that the temporal nature of data can be broken sometimes by having a test sample from an earlier period in time being used to test against a model which also includes future time periods in the data.

To mitigate this, another sampling technique was used – Walk Forward Validation [13]. In this technique, the model is first trained on n weeks, and then samples from the $n+1$ week are used for validation. Next the model is trained on $n+1$ weeks and test samples from the $n+2$ week are used for validation. This process is repeated from a value matching the first week of data until it reaches the end of the data set. Effectively the model is incrementally "walking through" the entire data set week by week to perform its validation. The accuracy of the model is the statistical average of all the test results across $n-1$ weeks. This validation method results in improved measurement of the accuracy since it respects the temporal nature of the data.

F. Model Tuning

The model tuning phase involves adjusting the parameters of the model to improve the accuracy. After an initial set of features were selected in the feature engineering stage and the models were trained and tested, the resulting model accuracy was measured. The process was repeated by adding additional features, removing certain features and modifying the approach to feature engineering.

The model tuning efforts resulted in the addition of new features into the dataset. One example of tuning was altering the datetime feature in the original data. The original GMTDATE feature was separated into individual characteristics such as hour/week/year and so on, which resulted in better model learning. Another example was the addition of business-rules model predictions to the ML models. Running the business-rules algorithms and adding business rules predictions as new features helped the ML model improve their accuracy.

IV. Implementation

All the ML modeling was done on the Amazon Web Services (AWS) infrastructure. AWS offers a rich set of tools for training and implementing ML models. The tools used included:

- S3 Buckets – AWS data storage capability used to store the files used by the application
- CodeCommit Repository – Source code repository used to maintain source code
- CodePipeline – Capability used to deploy code from the CodeCommit Repository to the S3 bucket
- Jupyter notebook – Interactive tool used to develop ML models in Python
- SageMaker – AWS service which allows users to create Jupyter notebooks
- EC2 instances – AWS compute resources used to run models
- Lambda – AWS capability that allows users to run serverless code and scripts
- API gateway – AWS service that provides Application Program Interfaces (API) to allow users to connect user applications to AWS services

A. Training environment.

The data for the training and testing the models was stored on an Amazon S3 buckets. Amazon SageMaker was used training and testing the various machine learning models. Jupyter notebook instances were created for different combinations of airports, machine learning model types, prediction lead times and data sampling type. Each Jupyter notebook is a self-contained artifact which contains all the python source code needed to run the models. Once a model was developed, trained and tested, it was deployed on an EC2 instance which serves as its runtime.

B. Deployment environment

A web-based user interface was developed to enable analysts to compare results from the various approaches used for prediction of runway configurations, airport arrival rates and airport departure rates. architecture for the deployed system.

V. Model Performance

Once deployed, users can employ the browser- based User Interface (UI) shown in Figure 3, to run predictions and visualize the returned results and model metrics such as prediction accuracy. The user interface displayed in one table the results of the ML predictions for Runway Configurations, Arrival and Departure rates for and in a second table the results returned by the application of the traditional business rules model, so that users could readily compare the ML models with the Business Rules model. The UI was developed using bootstrap.js a Javascript framework.

As discussed in the preceding section, individual ML models were developed for each of the following combinations:

- 3 Prediction targets - Runway Configurations, Airport Arrival Rate (AAR) and Airport Departure Rate (ADR)
- 5 different airports – ATL, BOS, EWR, JFK and LGA
- 4 ML model types – Logistic Regression, Random Forest, Cat Boost and Neural Networks
- 2 Sampling types for validation : Random Sampling and Walk Forward Validation Sampling
- 3 Prediction Lead Times: 3-hours, 14-hours and 24-hours

This results in some 360 ML models, each having its own accuracy. In this section, the overall trends of the study are presented first in Tables III–V, and then a small, representative subset of the significant results are presented.

TABLE III. ACCURACY OF BUSINESS RULES MODELS VS. DIFFERENT ML MODELS (WHEN APPLYING RANDOM SAMPLING)

Airport	Business Rules	Random Forest	Logistic Regression	Catboost	Neural Network
EWR	0.66	0.80	0.63	0.83	0.85
LGA	0.47	0.80	0.45	0.82	0.85
JFK	0.43	0.75	0.46	0.78	0.79
BOS	0.42	0.72	0.41	0.77	0.79
ATL	0.63	0.86	0.65	0.87	0.88

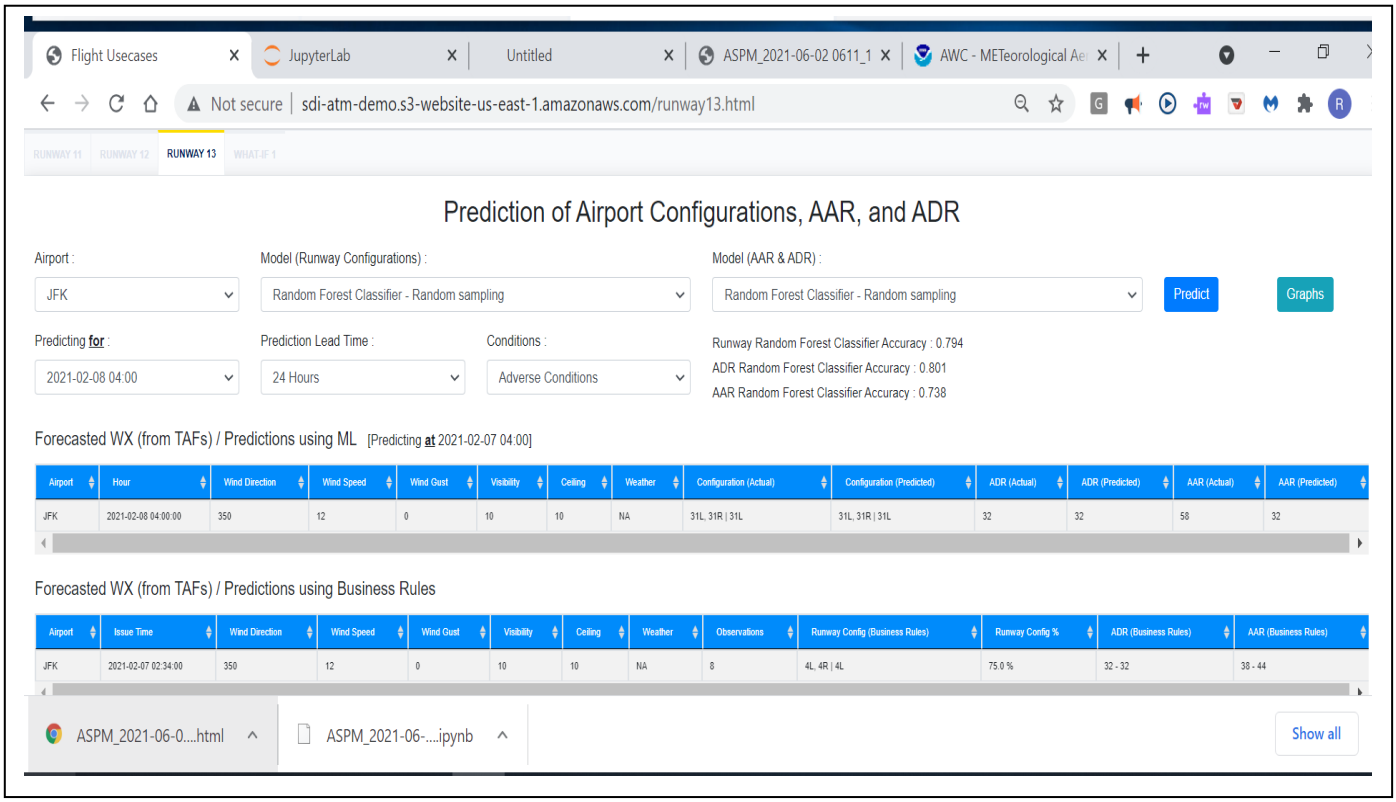

Fig. 3. User interface

Table III shows the accuracy of business rules models and ML model types, for five airports. These results are for the random sampling validation (the ML models) and average the results across prediction lead times of 3, 14, and 24 hours. Other than the logistic regression model, the ML models show consistently higher accuracy in predicting runway configuration, AAR, and ADR.

Table IV compares the accuracy of different ML model type and sampling type combination with each other when predicting the three target variables in the models – runway configuration, AAR and ADR. In the table a sampling type of RS represents Random Sampling Validation and WF represents Walk Forward Validation. Accuracy is averaged across all airports and prediction lead times. Here Random Sampling validation tended to produce more accurate predictions and Logistic Regression tended to produce the least accurate predictions.

TABLE IV. AVERAGE ML MODEL ACCURACY ACROSS ALL AIRPORTS AND PREDICTION LEAD TIMES

Model and Validation Type(*)	RWYCONF	AAR	ADR
CAT boost-RS	0.827	0.769	0.844
Catboost-WF	0.645	0.637	0.696
Logistic Regression-RS	0.509	0.489	0.554
Logistic Regression-WF	0.472	0.524	0.587
Neural Network-RS	0.838	0.796	0.865
Random Forest-RS	0.791	0.752	0.817
Random Forest-WF	0.646	0.634	0.680

TABLE V. PREDICTION ACCURACY ACROSS ALL MODELS, SAMPLING TYPES AND AIRPORTS FOR DIFFERENT LEAD TIMES

Prediction Lead Time	RWYCONF	AAR	ADR
3 hours	0.720	0.698	0.757
14 hours	0.654	0.637	0.703
24 hours	0.652	0.637	0.701

Table V shows how accuracies vary with different prediction lead times, once again 3, 14, and 24 hours. Results are averaged across all model types, sampling types and airports. The 3-hour prediction lead time has the highest accuracy, and the 14 and 24-hour prediction lead times have almost identical accuracy.

With 360 different models across prediction goals (AAR, etc.), five airports, four model types, two sampling types, and three prediction lead times, there are many choices for both reporting the results and selecting the best model. In internal development, we always plotted the three prediction goals side by side. For each prediction goals, we also plotted side-by-side the accuracies for the no skill model (majority prediction), persistence (configuration and rates are the same as those at the time of the prediction), business rules, and a single ML model. Fig 4. Shows this internal reporting, here for airport ATL, a Random Forest classifier, random sampling, and a 24-hour prediction lead time.

VI. DISCUSSION

For the use case of predicting airport runway configurations, AARs, and ADRs, this study explored two key questions:

- Do ML models yield better accuracy than the business rules models?
- Which ML models had the highest accuracy for this use case?

For different combinations of five airports, four ML model types, three prediction lead times and two sampling types, 360 ML models were developed. Each model had unique characteristics and performance.

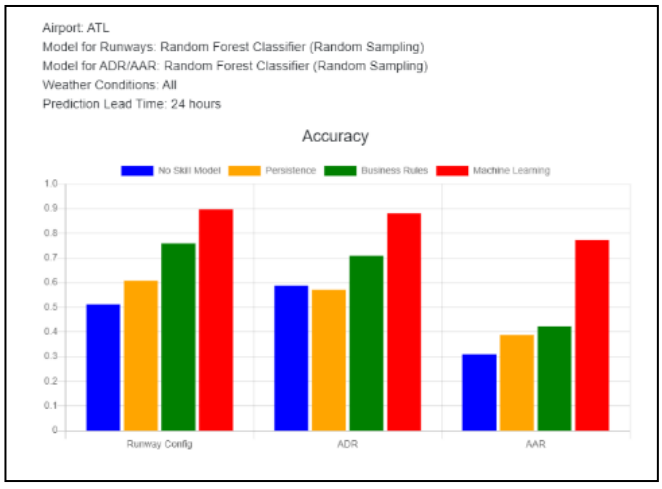

Fig. 4. Sample model performance comparison for random-forest classifier

The answer to the first question "Do ML models yield better accuracy than the business rules models?" was found to be affirmative. With one exception, the logistic regression model, all ML model types consistently delivered better accuracy than the business rules model. In some cases, the ML models had significantly better accuracy than the business rules models, as presented in Table III.

The second question "Which ML models had the highest accuracy for this use case?" does not have a simple answer, and the selection of a model with the highest accuracy depends on a number of factors. Some models performed very well on some airports and prediction lead time combinations and performed poorly on others. There was also variation in model accuracy when different sampling techniques were used to validate the models. Table IV shows model accuracy for prediction of Runway Configurations, AAR and ADR for a combination of 4 model types and 2 sampling techniques applied across the entire data set for all five airports, and across all three prediction lead times.

The lack of a clear best model does not preclude us from making several observations on findings from this work. These observations include the following general trends.

Impact of sampling on accuracy: validation with random sampling yields better accuracy than validation with walk forward validations. As we had discussed in the ML Model Development section, random sampling has a known limitation – it breaks the temporal nature of the data. When making a prediction, a random test sample from an earlier time period will benefit from learning which happened in a later time period, and is not an accurate reflection of how the real-life predictions will work. The walk forward validation is validation method which better mirrors real-world behavior.

Impact of model type on accuracy: When applying random sampling, the model with the highest accuracy is the Neural Network model validated. This model and sampling combination yields the best accuracy for all three target variables – Runway Configurations (0.838), AAR (0.796) and ADR (0.865). When applying walk forward validation, Random Forest performed the best for Runway Configuration (0.646) but CatBoost was a very close second (0.645). However, CatBoost outperformed other models for AAR (0.637) and ADR (0.696).

Impact of airport on accuracy: Every airport performs differently. When we looked across the data set, no common patterns could be observed. This could be attributed to the small data sample as only 5 airports were studied. The results section showed some of the differences between airports for different combinations of model types and prediction lead times.

Impact of prediction lead time on accuracy: As can be intuitively anticipated, a shorter prediction lead time leads to improved accuracy. Table 3 in the results section shows that when we compare average prediction accuracy across all model types and airports. As the table confirms, the smaller the prediction lead time, the better is the prediction accuracy. Surprisingly, however, the 14 and 24-hour prediction lead times have almost identical accuracy.

A. Extensions

Although 360 models may suggest that the modeling is complete and the code is ready to be deployed in the field, there are numerous possibilities for extending the work described here. The model tuning described earlier, adjusting the parameters of the model to improve the accuracy, can be continued, should potential end users find the current accuracy insufficient. Deep learning allows for considerable tuning and adjustment. As another extension, additional input features can be added to the model. Besides the weather, there are many factors that influence the choice of a runway configuration and rates. A closed runway, of course, preclude many possible configurations, Runway closures are announced in NOTAMs, and NOTAM text could be mined and used as training and input data.

Finally, predications under benign and adverse weather could be examined more closely. Benign weather provides considerable latitude in selecting a runway configuration. This flexibility is convenient for operations but less so for modeling, since the coupling between weather and the runway configuration and rates will be weaker. Adverse weather such as strong winds and low visibility limit the allowable runway configurations and rates. While the authors did examine the influence of benign and adverse on prediction accuracy, the findings were not conclusive, and further analysis is warranted.

ACKNOWLEDGEMENTS

The authors of this paper acknowledge the continued support of Bryan Baszczewski and Derek Robinson from the Federal Aviation Administration and John Massimi of the Volpe National Transportation Systems Center. The authors also thank KBR's Strategic Development Initiative for sponsoring this project and gratefully acknowledge the contributions of the following individuals, in this paper: Saumitra Modak (Data Scientist), Anand Shah (Data Scientist), Riaz Moradian (Full stack developer) and Kabir Kapur (Software Intern).

REFERENCES

[1] Federal Aviation Administration, "Traffic flow management in the National Airspace System," October 2009. [Online].

[2] EUROCONTROL and Federal Aviation Administration, ", "U.S./Europe - Comparison of ATM-Related Operational Performance 2017," 2019.

[3] Federal Aviation Administration, "FY 2020 ATO Business Plan," 12 May 2020. [Online]. Available: https://www.faa.gov/about/plans_reports/media/2020/ato_business_plan.pdf.

[4] Federal Aviation Administration, "Collaborative Decision Making," [Online]. Available: https://cdm.fly.faa.gov.

[5] Federal Aviation Administration, "Aviation System Performance Metrics (ASPM)," [Online].

[6] Aviation Weather Service, "Text Data Server (TDS) ver 1.3," [Online]. Available: https://aviationweather.gov/dataserver.

[7] G. B. Valor, "OGIMET," [Online]. Available: http://ogimet.com/home.phtml.en.

[8] Aviation Weather Service, "METAR information," [Online]. Available: https://aviationweather.gov/metar/help?page=board.

[9] Y. Liu, Y. Wang and J. Zhang, "New Machine Learning Algorithm: Random Forest," in *Information Computing and Applications*, Berlin, 2012.

[10] M. Bardach, M. Gringinger, M. Schrefl and C. G. Schuetz, "Predicting Flight Delay Risk Using a Random Forest Classifier Based on Air Traffic Scenarios and Environmental Conditions," in *AIAA/IEEE 39th Digital Avionics Systems Conference*, 2020.

[11] A. V. Dorogush, V. Ershov and A. Gulin, "CatBoost: gradient boosting with categorical features support".

[12] A. Tealah, "Time series forecasting using artificial neural networks methodologies: A systematic review," *Future Computing and Informatics Journal,* vol. 3, no. 2, pp. 334-340, 2018.

[13] Vikash, "https://medium.com," 21 June 2020. [Online]. Available: https://medium.com/@Vikashov/validation-techniques-for-time-series-and-non-time-series-datasets-38833b4341cf. [Accessed 15 July 2021].