ATD-2 Integrated Arrival/ Departure/Surface (IADS) System Machine Learning Services

Airport Configuration Prediction Model (ACPM)

Shaymaa Khater, Juan Rebollo

Mosaic ATM Inc.

NASA Ames Research Center, Moffett Field, CA

William J. Coupe

NASA Ames Research Center, Moffett Field, CA

August 2021

Table of Contents

Model Name 3
Problem Statement 3
Technical Approach 3
Model Features 4
Model Inputs & Outputs 5
Data Sets 5
Model Results / Evaluation 6
Open Source Repository 6
Reference Documentation 6
Appendix: OpenAPI Specification 6

Model Name

Airport Configuration Prediction Model (ACPM)

Problem Statement

Airport configurations have a large impact on many components of the air traffic management system. A specific airport configuration determines which arrival and departure runways are active at a given time, under certain operating conditions, and the available airport capacity. Accurate predictions of future airport configuration reduces airport capacity uncertainty, which enables traffic flow management to better match demand to capacity. Both surface and airspace operations can benefit from better understanding of future airport configurations. Better knowledge of which runways will be active ahead of time enables more accurate predictions of taxi times and better utilization of resources on the airport surface.

Airport configuration selection is a complex decision making process that involves several operational and human factors. ACPM is a ML model that predicts a set of airport configurations at a given airport, and up to 6 hours ahead. There are separate trained models for each airport, and the model to query must correspond to the airport whose configuration is to be predicted.

The model uses 3 data sources:

1. Weather data source: As weather is one of the key drivers for airport configuration, ACPM model includes weather data elements like wind direction, wind speed, cloud and temperature. This data is obtained from Mosaic's weather database for Localized aviation MOS Program (LAMP) system. The database includes an hour-by-hour weather forecast data for the next 25 hours for more than1600 stations in the Continental United State (CONUS), Alaska, Hawaii and Puerto Rico.
2. Traffic data: ACPM needs to estimate the future traffic (Departure/Arrival counts) to predict the airport configuration. For this, NAS-wide fuser data tables were used.
3. Airport Configuration: ACPM uses historical data for airport configurations, specifying the active arrival and departure configuration over time. This is obtained from Data-link Automatic Terminal Information Service (D-ATIS) data.

Technical Approach

ACPM uses a recursive multi-step machine learning approach to predict airport configuration. This is a known time series forecasting technique which involves running a one-step model multiple times, where the prediction from the prior time step is used as input to generate the prediction for the following time step. When predicting airport configuration, the airport configuration predicted at the prior time step is used as input in the following time step, and in the first step the value is set to the current configuration. For this approach, three parameters are defined, the step size, the overall prediction look ahead time and the prediction model running in each iteration/step. The models deployed use 30 minutes for step size, 6 hours as overall prediction look ahead time and 3 hours as model look ahead. Within the multi-step forecasting approach, a variety of ML models can be used to predict the airport configuration at each step. The models deployed use XGBoost classifier, as it leads to better performance than other classifiers.

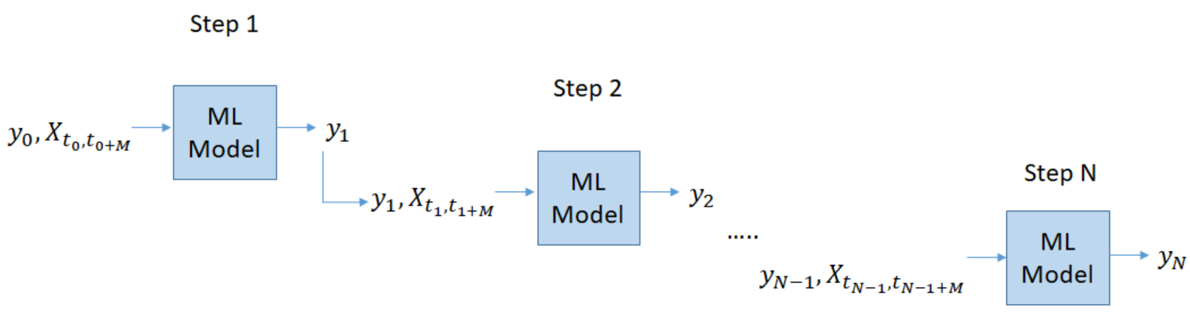

Figure above is for Recursive Multi-step Forecasting approach. Here *M* is the number of steps in the model look ahead, *N* is the number of steps in the prediction look ahead, y is the target variable, and X_{t_a,t_b} is the feature vector with feature data from time step *t_a* to time step *t_b*.

Model Features

Input	Description	Example
Timestamp	Time now (at prediction time)	2021-01-10 10:00:00
Airport_configuration_name_current	Current airport configuration	D_17R_18L_A_17C_18R
Arr_count_"X"	Number of arrivals at the airport between (timestamp +(X-30) minutes) and (timestamp + X minutes)	10
Dep_count_"X"	Number of departures at the airport between (timestamp +(X-30) minutes) and (timestamp + X minutes)	15
Cloud_"X"	Cloud forecast at the airport between (timestamp +(X-30) minutes) and (timestamp + X minutes)	FW
Cloud_ceiling_"X"	Cloud ceiling forecast at the airport between (timestamp +(X-30) minutes) and (timestamp + X minutes)	8
Lightening_prob_"X"	Lightening_prob forecast at the airport between (timestamp +(X-30) minutes) and (timestamp + X minutes)	L
Precip_"X"	Precip forecast at the airport between (timestamp +(X-30) minutes) and (timestamp + X minutes)	TRUE
Temperature_"X"	Temperature forecast at the airport between (timestamp +(X-30) minutes) and (timestamp + X minutes)	70
Visibility_"X"	Visibility forecast at the airport between (timestamp +(X-30)	7

Input	Description	Example
	minutes) and (timestamp + X minutes)	
Wind_direction_"X"	Wind direction forecast at the airport between (timestamp +(X-30) minutes) and (timestamp + X minutes)	12
Wind_gust_"X"	Wind gust forecast at the airport between (timestamp +(X-30) minutes) and (timestamp + X minutes)	4
Wind_speed_"X"	Wind_speed forecast at the airport between (timestamp +(X-30) minutes) and (timestamp + X minutes)	3

Note: for "X" features: The number of features depends on the prediction_lookahead, prediction_delta and model_lookahead. Values need to be provided up to (prediction_lookahead + model_lookahead - prediction_delta). In the deployed models, prediction_lookahead = 360min, prediction_delta = 30min and model_lookahead = 180min, so values used in the model for X=[30,60,90 ... ,510].

Model Inputs & Outputs

Predicted airport configuration (e.g., D_17R_18L_A_17C_18R) from 30 minutes up to 6 hours ahead of time.

See OpenAPI specification in the appendix.

Data Sets

- Input data - The model used fused dataset of TFMS, TBFM feeds. Traffic features (future arrival/departure count) were derived from these data sources. D-ATIS feed was used to obtain the current airport configuration needed to train the model. LAMP weather data feed was used to obtain wind, temp, and other weather features needed to train the model.
- Training / Test data : Different versions of the model is trained/ tested on two data ranges:
  - 2019 Data

Airport	Start_time	End_time
CLT	08-01-2019	12-31-2019
DFW	08-01-2019	12-31-2019
EWR	08-01-2019	12-31-2019
JFK	08-01-2019	12-31-2019
DAL	08-01-2019	12-31-2019

- 2020 Data

Airport	Start_time	End_time
CLT	07-01-2020	10-31-2020
DFW	07-01-2020	10-31-2020
EWR	07-01-2020	10-31-2020
JFK	07-01-2020	10-31-2020
DAL	07-01-2020	10-31-2020

- 80% of the data was used for training and 20% for testing

Note: The registered models on MLflow server are the ones trained on 2020 Data.

Model Results / Evaluation

The table below shows the average accuracy calculated over the prediction look ahead time measured for different airports.

Airport	Training accuracy	Testing Accuracy
CLT	0.75	0.74
JFK	0.79	0.76
EWR	0.78	0.74
DFW	0.76	0.76
DAL	0.78	0.77

Open Source Repository

https://github.com/nasa/ML-airport-configuration

Reference Documentation

Khater, S., Rebollo, J., Coupe, W., “A Recursive Multi-step Machine Learning Approach for Airport Configuration Prediction,” AIAA AVIATION Forum, Washington, DC, USA, 2021.

Appendix: OpenAPI Specification

Explore

Airport Configuration Prediction Client

1.0.0 OAS3

Airport Configuration Prediction Client

Servers

http://localhost:9107/ - Airport Configuration Prediction Client

Computed URL: `http://localhost:9107/`

Server variables

port 9107

default

POST `/airport-config-prediction`

Schemas

AirportConfigPredictionData {

```
airportConfigurationNameCurrent  string
                                 example: D_17C_17R_18L_18R_A_13R_17C_17L_18R
                                 nullable: true

                                 current airport configuration

arrCounts                            [...]
cloud                                [...]
cloudCeiling                         [...]
depCounts                            [...]
lightningProb                        [...]
precip                               [...]
temperature                          [...]
visibility                           [...]
windDirection                        [...]
windGust                             [...]
windSpeed                            [...]
}
```

AirportConfigPredictionRequest {

```
airport*           string
                   example: KDFW

                   airport name

data*                  [...]
increment*         integer($int32)
                   example: 30

                   lookahead increment in minutes

timestamp*         string($date-time)
                   example: 2020-07-06T13:46:01Z

                   timestamp

}
```

AirportConfigPredictionResponse {

```
airport*        string
                example: KDFW

                airport name

configs*            [...]
increment*      integer($int32)
                example: 30

                lookahead increment in minutes

}
```

AirportConfigPredictionResponseConfig {

```
error*          string
                example: Unknown airport_configuration_name_current

                error description

pred                [...]
}
```