# *Applying Machine Learning Tools for Runway Configuration Decision Support*

Krishna M. Kalyanam
Aviation Systems Division
NASA Ames Research Center
Moffett Field, CA, USA
krishna.m.kalyanam@nasa.gov

John Crissman, Rebekah Yang
Center for Data Management Analytics
CNA Corporation
Arlington, USA
crissmanj@cna.org, yangr@cna.org

Milad Memarzadeh
Universities Space Research Association
NASA Ames Research Center
Moffett Field, CA, USA
milad.memarzadeh@nasa.gov

Kanvasi "TJ" Tejasen
Office of NextGen
FAA
Washington, D.C., USA
kanvasi.tejasen@faa.gov

**Determining optimal runway configurations at airports, a responsibility assigned to air traffic controllers, is a challenging task. The decision-making process is intricate and involves consideration of many factors such as prevailing wind condition, convective weather, visibility, cloud ceilings, departure and arrival demand, traffic flow, equipment status, and other airport constraints. In a previous work, we developed a Runway Configuration Assistance tool using an offline reinforcement learning method called conservative Q-learning. In this paper, we evaluate and validate our Runway Configuration Assistance tool as a decision support for air traffic controllers. We validated our tool using three airports with differing levels of complexity: Charlotte Douglas International Airport, Denver International Airport, and Dallas Fort Worth International Airport. We quantified the performance of the Runway Configuration Assistance tool based on (1) agreement with historical air traffic controller decisions and (2) violation of decisions that would be obvious to subject-matter experts. Our tool showed promising results in both performance metrics for the three airports, despite the complexities in the runway configuration decision-making process. We also discuss challenges in using machine learning in general to aid air traffic management and identify deployment considerations for the Runway Configuration Assistance tool.**

## I. Introduction

The runway configuration at an airport determines the active runways in use for arrival and departure of aircraft at any given time. Air traffic controllers (ATCOs) are responsible for determining the runway configuration of an airport and thus designate which runways are active. Runway configuration decision-making is challenging because many factors are involved in determining the optimal runway configuration for current traffic, prevailing winds and future (forecast) weather conditions.

Weather is a primary factor that affects runway configuration decisions. Current weather and forecast weather, including but not limited to wind direction, wind speed, visibility, ceiling, thunderstorms, icing, turbulence, and windshear must be considered. For example, runways are constructed to align with known wind patterns at an airport to allow aircraft to arrive and depart into the wind. Landing into the wind aids the aircraft in achieving slower speeds, whereas landing with tailwinds can lead to dangerous landing conditions. Thus, when wind direction changes, runway configurations and flight paths typically are adjusted accordingly. Severe weather such as thunderstorms, icing, lightning, and turbulence near an airport may require a change in runway configuration to enable flight paths that avoid adverse weather conditions. Visibility and cloud ceiling conditions also influence runway configuration selection. Runways may be selected based on visual flight rules (VFR) or instrument flight rules (IFR). For example, under VFR, simultaneous arrivals and parallel arrival runways may be feasible, but under IFR, a single runway may be required for air traffic control separation standards.

Traffic flow also influences the runway configuration decision. Traffic flow includes both the arrival and departure demand at the airport and surrounding airports. Airports may have different configurations that are optimal at higher and lower capabilities. Furthermore, the traffic mix may also be a factor. For example, larger international aircraft may require longer runways and thus a different configuration. Finally, surrounding airport traffic and airspace flows may affect runway configuration decisions, especially in major metroplexes where



Sponsor: FAA NextGen Portfolio Management and Technology Development Directorate

multiple airports need to coordinate the effects of changes in runway configuration.

Additionally, runway conditions, runway availability, approach availability and airport constraints factor into the configuration. Approach and runway availability may be dictated by weather, equipment status, ceilings, visibility, and runway visual range values. Airport constraints may include construction, repairs, gate status, parking, noise abatement and impact on neighboring airports. For example, to comply with noise abatement requirements, airports may require a predefined configuration at certain time periods. Ad hoc constraints such as Notices to Airmen (NOTAMs) or Traffic Management Initiatives (TMIs) also factor into configuration decisions. NOTAMs may report equipment outages (e.g., navigational aids) that require changes to flight plans, and TMIs may impose delays to aircraft arriving at an airport.

Determining the best runway configuration for a given set of conditions in a timely manner is important for both safety and efficiency reasons. If the runway configuration is not ideal for current conditions, arriving aircraft may experience delays and increased safety risks. Aircraft unable to land safely (e.g., because of tailwinds) may be forced to engage in airborne holding or diversion until conditions change. Aircraft may also have to perform "go-arounds" if landing conditions are below approach minimums. In addition, changes in configuration require coordination and time to implement. Changes must be planned with affected facilities, including Terminal Radar Approach Control facilities (TRACONs) and Air Route Traffic Control Centers (ARTCCs). As such, unnecessary or suboptimal configuration changes may cause inefficiencies through extraneous coordination efforts and time spent. These inefficiencies may also cause compounding impacts to aircraft through additional delays.

### *A. Current Runway Configuration Management Decision-Making Process*

In current practice, runway configuration decisions are made by air traffic controllers (ATCOs) based on acquired knowledge. The decision to change the configuration relies on the experience of the controllers on duty at any given point in time. Experienced controllers can successfully manage runway configurations, but there is a steep learning curve, and each airport is different. The controller must consider all factors described previously to make informed decisions. Less common or more complex situations may pose a challenge for the controller. For example, rapidly shifting winds may require multiple configuration changes, and determining the optimal time to initiate changes is nontrivial. Anticipating and predicting changes in advance is key to a smooth configuration change to allow coordination between the tower, TRACON and ARTCC to occur as soon as possible. Because the process considers past human decisions and quantifiable data, machine learning (ML) models can be used to provide insights that enhance the runway configuration decision-making process. These models can suggest changes for the ATCOs to supplement their airport-specific experience and provide parallel recommendations based on historical data as supporting evidence for ATCO decisions.

### *B. Existing Machine Learning Literature for Runway Configuration Management*

In recent years, there has been an increase in the use of ML to evaluate the runway configuration decision process. Generally, the literature in this area can be grouped into two main categories: *model-based* and *model-free*, where the word *model* describes the underlying system dynamics.

In model-based approaches, once a model of dynamics is learned, the optimal policy for runway configuration decisions can be found by solving an optimization problem using the learned model. Different techniques have been used to identify the optimal policy, such as heuristic search [1, 2], discrete-choice modeling [3], mixed-integer programming [4], dynamic programming [5], and queuing theory [6-8]. Model-based approaches are interpretable and provide guarantees on the near-optimality of obtained policy for the runway configuration. However, the performance of model-based approaches and the associated guarantees depend on the accuracy of the learned model, which can be a challenge in real-world scenarios given the limitations in data availability.

In contrast, *model-free* approaches directly learn the optimal policy from data without relying on learning a model for the underlying dynamics. These approaches can be categorized into two distinct groups: model-free control methods and data-driven supervised learning techniques. Model-free control approaches mainly rely on reinforcement learning (RL) techniques [9]. They have been widely adopted and deployed in the aviation domain [10], with Monte Carlo tree search [11] and Q-learning [12-14] being the most popular methods. These approaches usually learn a good policy by making decisions (i.e., acting) in a simulated or operational environment, and they learn near-optimal policies based on the feedback received in response to the actions. Model-free approaches are generally easy to implement and efficient to scale. However, they rely on a significant number of interactions in either a simulation environment or an operational setting to learn a good policy (called online RL). These interactions can be costly, especially when applied to real-world systems such as air traffic management, because the algorithm tends to explore poor decisions when the interactions are limited. Data-driven supervised learning techniques use vast amounts of available historical data and learn to imitate ATCO with the least amount of error [15-19]. They are easy to generalize from airport to airport and scale to the National Airspace System (NAS). However, they have one fundamental drawback: the optimization formulation of these techniques is designed to mimic ATCO with the least amount of error, so they cannot identify and correct mistakes or inefficiencies in the historical decisions. Moreover, their predictions are not supported by any evidence of better outcomes. Simply put, they learn to mimic historical decisions, both good and bad.

### C. *Runway Configuration Assistance Tool*

To address the limitations of the abovementioned techniques in runway configuration decision-making, we developed a solution based on a family of model-free control methods called offline RL. Offline RL combines the power of RL with data-driven supervised ML and attempts to learn a good policy by relying only on the historical data and decisions. This feature addresses the major shortcoming of online RL and removes the need for an online interaction with the simulation or operational environment [20]. However, the brute force implementation of online RL solutions in an offline mode suffers from *distributional shift,* where the policy that the RL algorithm learns from historical data is significantly different from the policy that was used (by ATCO in an operational setting) to collect the data (referred to as behavioral policy). The result is that the algorithm is overly optimistic (and likely wrong) when exposed to *out-of-distribution* (OOD) data, a setting that is not well represented in the historical data.

As described in previous work, we leveraged a state-of-the-art offline RL algorithm called conservative Q-learning (CQL) [21, 22] to develop a Runway Configuration Assistance (RCA) tool. CQL uses a simple regularization technique to alleviate the fundamental challenge of offline RL mentioned above. Our RCA tool is a decision-support tool that is intended to aid ATCOs in determining runway configurations. This paper describes developmental decisions that we made for the RCA tool and the validation efforts that we undertook to compare the RCA tool against historical ATCO decisions and subject-matter expert (SME) decisions. Finally, we discuss risks that need to be addressed and implementation issues that would need to be considered if the tool were to be deployed in the future.

## II. METHODOLOGY

### A. *Model Development and Subject-matter expert Engagement*

In the presence of an accurate simulation environment, or access to the operational environment, the online model-free RL algorithms can interact with the system and learn policies based on the feedback that they receive from the environment [9]. As discussed before, in real-world challenges such as the runway configuration decision process, access to the operational environment for the purpose of learning a policy is not possible. To the best of our knowledge, no accurate simulator exists that can mimic the real-world operations and generate the data required for runway configuration decision-making. As a result, we developed the RCA model using the offline model-free RL methodology, CQL [21]. This approach was chosen because it is suitable for addressing real-world challenges such as runway configuration decision-making. It removes the need for real-time interaction with the system for data collection, and policies are learnt based on historical data.

CQL builds upon the popular Deep Q-Network [14] and addresses the main challenge of offline RL (i.e., distribution shift). The main goal in a Q-learning algorithm [12, 13] is to find the optimal Q-function, which quantifies the long-term expected sum of utilities given that the decision-maker follows the optimal policy. Once the optimal Q-function is learned, the optimal policy can be easily obtained. CQL addresses the distribution shift challenge in offline RL by regularizing the estimates of the values of the Q-function by the function approximator, which is a neural network here. This additional regularization technique keeps the estimates of the Q-values for the unlikely actions (based on historical data) low; hence, it lower-bounds the optimal Q-function. Because CQL learns the lower bound for the optimal Q-function, a policy chosen based on the learned function will be conservative and take less risky actions. CQL also prevents the over-estimation of the Q-values for OOD data and function approximation error.

In designing the RCA tool, we continuously engaged with SMEs to incorporate their domain knowledge. The SMEs are former ATCOs with many years of experience. They played a vital role in conceptualizing the RCA problem and defining its different components, such as the important features to be included in the state space, identification of major runway configurations as actions, and different elements that are important to shape the utility function. Their involvement ensured that the framework accurately represented the complexities of runway selection decisions and considered external variables such as traffic and wind conditions. Their expertise influenced the problem formulation, data processing methods, choice of solution methodology, choice of airport case studies to create the RCA model, and validation of the chosen approach, ensuring that the resulting model would be well aligned with the complexities and safety considerations inherent in air traffic management.

### B. *Model Validation*

Based on feedback from SMEs, we chose three major US airports for the validation of the RCA tool. These three airports represent the variety of airports across the NAS on the runway configuration decision-making complexity spectrum. Our first case study is Charlotte Douglas International Airport (CLT). CLT is an example of a major airport with relatively few runway configuration options. As depicted in Fig. 1 (top panel),



Identify applicable sponsor/s here. *(sponsors)*

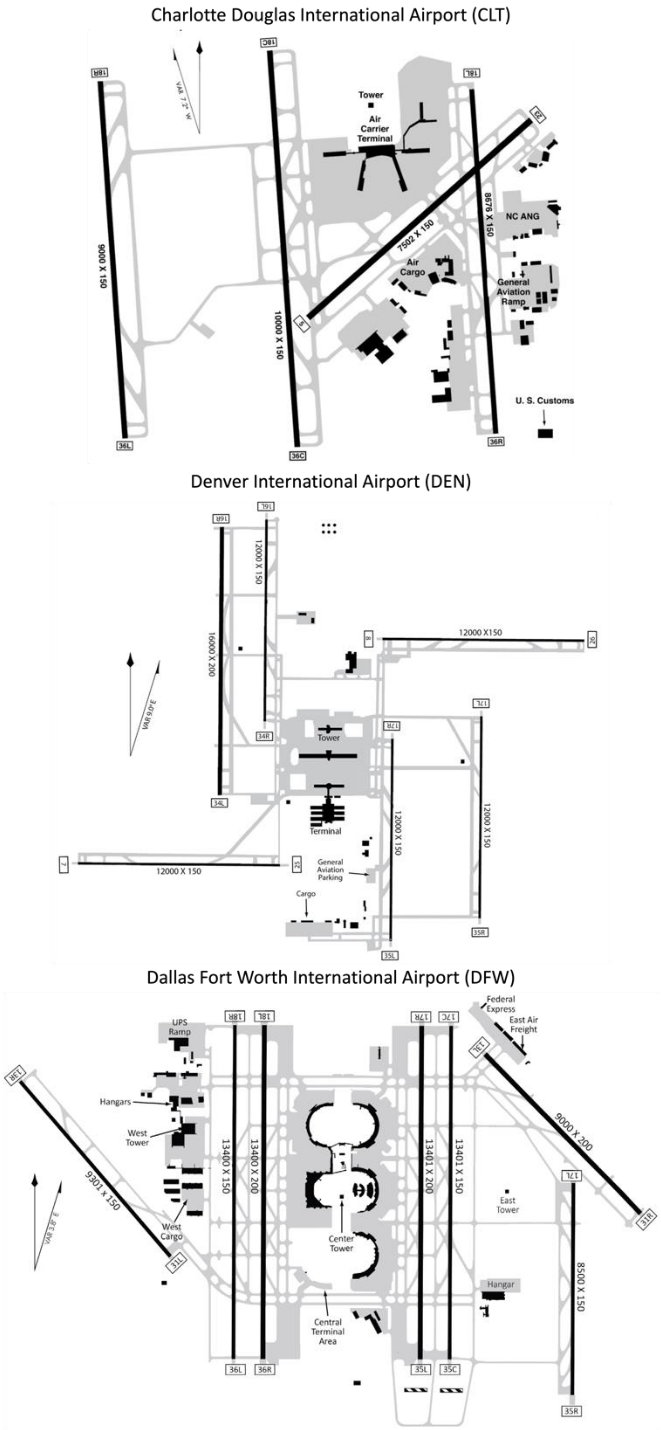


Figure 1. Runway diagrams for CLT (top), DEN (middle), and DFW (bottom), the three airports studied in this paper.

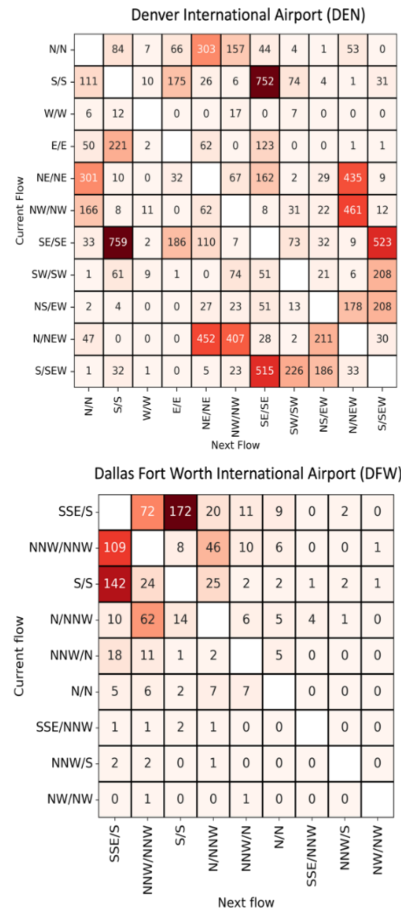


Figure 2. Heatmaps of the runway configuration changes performed by ATC in a year at DEN (top) and DFW (bottom).

CLT has three parallel runways and a short diagonal one that is rarely used. It has two major configurations: north flow (use of runways 36R/C/L) and south flow (use of runways 18R/C/L).

On the other end of the complexity spectrum is Denver International Airport (DEN). DEN is an example of an airport with a complex array/menu of runway configurations. As depicted in Fig. 1 (middle panel), it has six runways, among which four (34L/16R, 34R/16L, 35L/17R, and 35R/17L) are north/south bound and two (7/25 and 8/26) are east/west bound. Based on our comprehensive analysis of the data for the years 2018 and 2019 and feedback from SMEs, 11 major configurations were identified. For example, for the configuration named N/NEW, northbound runways (34R/L and 35R/L) are used for both arrival and departure, whereas runways 8 (eastbound) and 25 (westbound) are used only for departure. Fig. 2 (top panel) shows the heatmap of changes in the runway configurations by ATCO in a single year. The rows show the flow (configuration) at each time interval, and the columns show the flow (configuration) at the next time interval. The diagonals of this matrix represent no changes in the configuration because the diagonals represent most of the data; we have masked them in this figure to better illustrate the flow changes and the runway configuration decision-making complexity. Most of the flow changes are intuitive; for example, the configuration SE/SE transitions to either S/S or S/SEW most of the time, depending on the changes in the operational conditions. However, we also observe major changes in the configuration often at DEN (e.g., switches from NE/NE to SE/SE or N/N to S/S).

The third airport that we chose was Dallas/Fort Worth International Airport (DFW), which is representative of a major airport with multiple runways and moderate runway configuration decision-making complexity. As shown in Fig. 1 (bottom panel), DFW has five runways that are north/south bound (35/17 R/C/L and 36/18 R/L) and two runways that are northwest/southeast bound (13/31 R/L). Based on the analysis of data and feedback from SMEs, we identified nine major configurations. Fig. 2 (bottom panel) shows a heatmap of changes in the runway configurations by ATCO in a single year. As shown, runway configuration usage at DFW has a significant imbalance. The top four configurations are used most of the time (96% of the time based on data from the years 2018 and 2019), and the other five configurations are rarely used. Table 1 shows the definitions of the major runway configurations identified with the help of SMEs for the three airports.

We obtained the data for 2018 and 2019 for the three airports for training of the RCA tool from two main sources: the Federal Aviation Administration (FAA) Aviation System Performance Metrics reports[1] and National Aeronautics and Space Administration (NASA) Sherlock Data Warehouse[2]. We discretize time into 15-minute intervals. The state space (i.e., relevant features for decision-making) of the runway configuration problem comprises: (1) hour of the day, (2) wind direction and speed, and (3) meteorological conditions, which are categorized into two unique states of VFR and IFR depending on the visibility and the cloud ceiling. The utility function is chosen based on factors that affect the decision-making process of ATCO as well as inputs from the SMEs. The function comprises (1) traffic throughput, (2) average transit times on the surface of the airport, (3) penalty for aircraft performing go-arounds, and (4) decision-making inertia to penalize sudden changes of the runway configurations from a time interval to the next one. The last term is designed to improve the stability of the decision-making by the RCA tool.

## III. Results and Discussion

### A. Tool Validation Results

To obtain the results, we divide the processed data randomly into three sets: training (60%), validation (20%), and testing (20%). The training and validation sets are used to train the RCA tool and perform hyperparameter tuning, and the testing set is used to estimate the performance of the trained model. The results shown in this section are only those of the final trained model on the testing set (unseen data during training and hyperparameter tuning). We quantify the performance of the RCA tool according to two metrics. The first is agreement with historical decisions (noted as agreement). For this metric, we show how often RCA's recommendation agrees with the historical decisions made by ATCO. We quantify agreement as both average agreement across the different configurations (Fig. 3) and the confusion matrix that shows the level of agreement for each configuration separately (Fig. 4). The second metric is violation of obvious decisions (noted as violation). For this metric, we identify obvious decisions as provided by the SMEs and estimate the percentage of time that RCA violates such decisions (Fig. 3). For example, in the case

Table 1. Definitions of the major configurations at the three airports.

| Config. [Arr/Dep] | Arrival Runways | Departure Runways | Usage Frequency [%] |
|---|---|---|---|
| **CLT** | | | |
| N/N | 36R/C/L | 36R/C | 60.8 |
| S/S | 18R/C/L | 18C/L | 39.2 |
| **DEN** | | | |
| SE/SE | 16R/L,17R/L, 7,8 | 16R/L,17R/L, 7,8 | 18.8 |
| S/S | 16R/L,17R/L | 16R/L,17R/L | 15 |
| N/NEW | 34R/L,35R/L | 34R/L,35R/L, 8,25 | 14.5 |
| S/SEW | 16R/L,17R/L | 16R/L,17R/L, 8,25 | 12.6 |
| N/N | 34R/L, 35R/L | 34R/L, 35R/L | 12.3 |
| NE/NE | 34R/L,35R/L, 7,8 | 34R/L,35R/L, 7,8 | 11.7 |
| NW/NW | 34R/L,35R/L, 25,26 | 34R/L,35R/L, 25,26 | 8.6 |
| SW/SW | 16R/L,17R/L, 25,26 | 16R/L,17R/L, 25,26 | 3.4 |
| E/E | 7,8 | 7,8 | 1.6 |
| NS/EW | 34R/L,35R/L,16R/L, 17R/L | 8,25 | 1.2 |
| W/W | 25,26 | 25,26 | 0.3 |
| **DFW** | | | |
| SSE/S | 13R,17R/C/L,18R | 17R,18R/L | 61.5 |
| NNW/NNW | 31R,35R/C/L,36 R/L | 31L,35C/L, 36R/L | 21.3 |
| S/S | 17R/C/L,18R | 17R,18R/L | 7.6 |
| N/NNW | 35R/C/L,36 R/L | 31L,35L,36R | 5.1 |
| NNW/N | 31R,35R/C/L,35R/L | 35R/L,36R/L | 3 |
| N/N | 35R/C/L, 36R/L | 35R/L,36R/L | 1.1 |
| SSE/NNW | 13R,17C/L,18R | 31L,35L,36R | 0.2 |
| NNW/S | 31R,35R/C,36L | 17R,18L | 0.1 |
| NW/NW | 31R/L | 31R/L | 0.1 |

[1] https://aspm.faa.gov/

[2] https://sherlock.opendata.arc.nasa.gov/sherlock_open/

study of CLT, if the wind is blowing strongly (more than 15 knots) from the north and the tool suggests the south configuration, the outcome is considered a violation of the tailwind criteria for landing and takeoff.

Fig. 3 shows the performance of the RCA tool based on the two metrics for the three airports. As can be seen, the performance for CLT—an airport where runway configuration is less complex—is outstanding; 77% of its recommendations agreed with historical decisions and 0% violated conventional norms. Note that the goal of the developed tool is not to agree with historical decisions 100% of the time. Ideally, the model should disagree with a historical decision if a better alternative solution could have been implemented in retrospect. In contrast, for DEN—an airport where runway configuration is highly complex—the tool achieved about 47% agreement with historical decisions; however, the violation metric is still low at around 0.4%. Note that there are significant variations in the historical decisions made by the ATCOs for DEN. When we implemented DEN's most frequent configuration in each specific state, the tool reached only 54% agreement with the actual decisions made, which shows how complex the decision-making process is for DEN. Finally, the RCA tool achieved an outstanding performance for DFW (moderately complex), with 74% agreement with historical decisions and 0.3% for the violation metric.

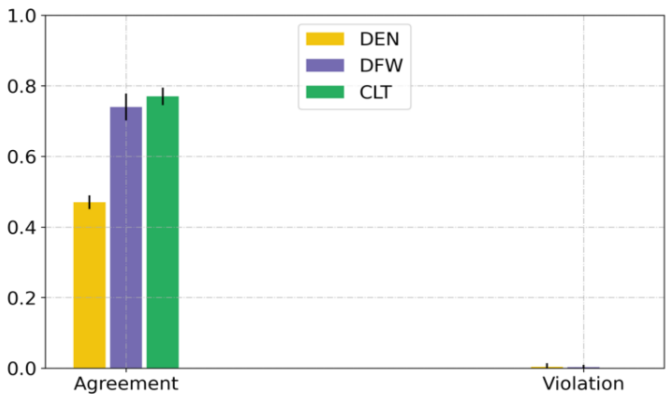


Figure 3. Performance of the RCA tool for three airports, DEN (yellow), DFW (purple), and CLT (green), based on the metrics of agreement with historical decisions and violation of obvious decisions.

Looking closer at the performance of the RCA tool at each airport sheds light on success and failure modes of the tool. Fig. 4 shows the confusion matrices for the performance of the RCA tool at each airport. Columns show the predicted configuration by the RCA tool, and rows depict the actual configuration that was used by the ATCO in each instance. The diagonal elements show the agreement of the RCA tool with historical decisions, and the off-diagonal elements are the disagreements.

In the case of DFW, we can see that the RCA tool performs well for the top four configurations (96% of the training data) while ignoring the minority configurations. Ignoring less frequent outcomes is a common problem in the ML community when dealing with imbalanced data, and our ongoing work is to alleviate this drawback. From the confusion matrix of DEN, we can see that although the agreement of the RCA tool with historical decisions is 47% on average, most disagreements occur between similar configurations. For example, we can see that the RCA tool primarily disagrees with the ATCO when predicting the S/S configuration—the tool instead predicts SE/SE, which is a sister configuration to S/S.

Note that not all disagreements are model errors. In some instances, the disagreement is due to the ATCO selecting an alternate configuration because of other operational constraints not captured in the training data. For example, in the case of CLT, through discussion with SMEs, we found out that when there are large fluctuations in the wind conditions, where say the surface wind changes from North to South and back to North again in a short period of time, the ATCOs typically do not respond to this change. However, without a mechanism to identify those cases, the RCA tool would respond and recommend a configuration change. Part of our ongoing research is to include forecast data so the model can learn this ATCO behavior and not react to short-lived changes in the wind conditions. We are working on improving the performance metrics so as not to penalize the model for minor disagreements compared to more obvious mistakes.

### *B. Machine Learning Considerations in Runway Configuration Management*

Although ML has the potential to enhance the current runway configuration decision-making process, risks to using an ML-based tool must be considered. In air traffic management, the risks associated with blind trust in ML demand careful consideration. The potential ramifications of suboptimal decisions made by ML models regarding airport configuration include adverse outcomes such as arrival/departure delays, prolonged transit times on the airport surface, and safety-related consequences such as go-arounds. The concern revolves around the prospect of ATCOs trusting the RCA tool to the extent of automating portions of their decision-making processes. A potential overreliance on ML models introduces a notable risk factor given the potential failure modes of these models and the possibility of errors. The safety implications of using an ML tool to aid runway configuration decision-making underscores the critical need for a nuanced approach to manage and mitigate associated risks effectively. The FAA is currently developing a Responsible AI framework and corresponding guidance to ensure that ML tools are sufficiently validated and supported for real-world use. The framework aims to supplement current software assurance guidance to account for the specific risks inherent in ML. The framework will provide systematic approaches and checklists for ensuring ML tools are developed responsibly and at the assurance level appropriate for their functions.

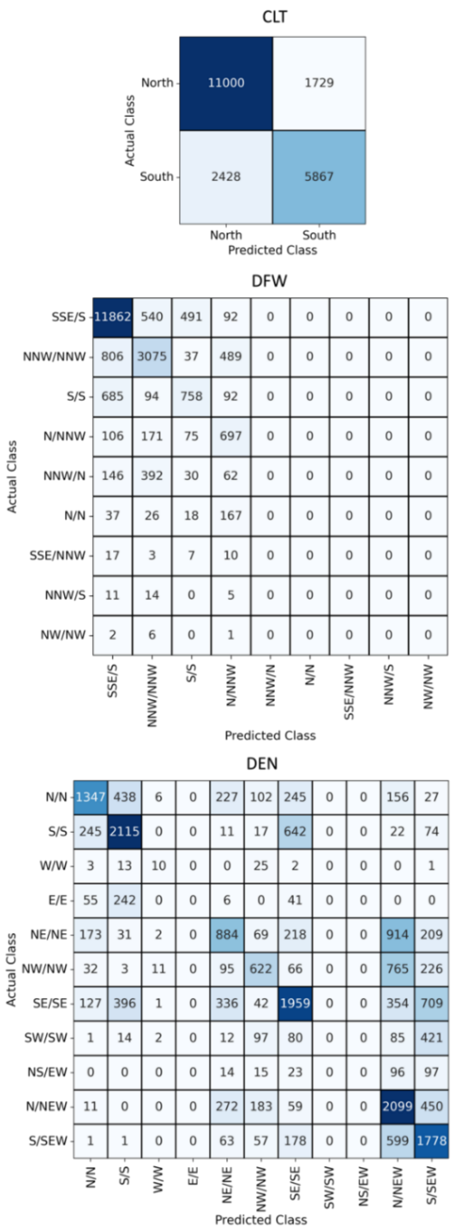


Figure 4. Confusion matrices for the RCA tool decisions with the three airports. Columns show predicted configuration, and rows show the actual configuration used. The diagonal elements represent correct predictions, and off-diagonal elements are the confusions/mistakes of the algorithm.

### C. *Deployment Considerations*

Although the RCA tool is still under development, it is important to consider how the tool might be evaluated for future deployment, if appropriate. A proposed deployment plan for the RCA tool should include several key steps to ensure its safe integration into service. Initially, pre-deployment testing could involve human-in-the-loop (HITL) simulation testing with SMEs from the target airport (CLT) to validate the tool's configuration predictions and assess its reliability. A suitable simulation tool would be needed to conduct this HITL testing.

Subsequently, the tool could be deployed in a “shadow” mode, where the RCA tool would run in parallel with the existing method ATCOs use for runway configuration decision-making. The output of the RCA tool would not be used for operations. The RCA tool would be used only for analysis. Data analysis would involve measuring SME acceptance or rejection of tool suggestions, and post-operational analyses would assess the effect on operational efficiency. Only when the RCA tool has been demonstrated to be satisfactory compared to the current runway configuration method, ATCOs would be provided operational access to the tool. Once trust in the system is established, the tool could be deployed for use in an operational setting to aid the ATCO in decision-making.

For the deployment strategy at any airport, collaboration with airport-specific SMEs may include HITL simulations and shadow deployments to test the RCA tool's functionality in predicting suggested configuration changes under various conditions. Detailed planning, coordination with facility managers, and adherence to facility procedures would be crucial for successfully deploying the RCA tool. A deployment plan would need to emphasize a gradual user acceptance approach, allowing users to adopt the tool's recommendations as trust and perceived benefits increase.

The RCA tool is intended as a decision-support tool for ATCO. The ATCO would make the final decision on whether to change the runway configuration, but the outputs of the RCA tool could help inform the decision. Thus, the tool would serve in an advisory position to provide information output to the ATCO to augment their existing resources [23]. Because of rapidly changing conditions, ATCO must make decisions quickly without time to formally consult similar historical situations. The RCA tool would be able to provide recommendations based on historical data.

The RCA tool is also intended to support decision-making for Traffic Flow Management (TFM) functions. Once the performance has been validated as acceptable, the tool could be used in conjunction with other TFM network planning capabilities to project expected upcoming operating states and support “what-if” analysis. The outcome of these TFM applications would be improved planning for stakeholders and airspace users.

## IV. Conclusion

Our RCA tool uses offline RL solution to overcome limitations inherent in online RL and model-based methods. The development and validation of the RCA tool involved active collaboration with SMEs using data from three major US airports. We used two metrics (agreement with historical decisions and violation of obvious decisions) and SME judgment to evaluate the performance of our RCA tool for three sample airports with varying degrees of runway configuration complexity (CLT, DEN, and DFW). Our historical decision metrics showed mixed results because of challenges related to operational constraints not captured in the training data (e.g., rapidly changing wind conditions) or model confusion between similar configurations. As such, we have planned future work to address these shortcomings. Still, our violation metric was less than 0.5% for each airport, indicating that the tool output rarely violates decisions that would be obvious to SMEs. We also explored considerations that would be needed if deploying the RCA tool in the future. Several considerations and risks are linked to implementing ML in air traffic management, highlighting the need for a measured approach to mitigate potential errors and safety concerns.

Overall, this paper captures the progress made in the validation of an ML tool for runway configuration decision-making by evaluating results against ATCO historical decisions and SME judgment for three major US airports. Although additional work is needed to address model challenges and future deployment considerations, the RCA tool has growing potential to provide decision support for complex runway configuration problems and other system wide TFM planning functions.

## References


[1] C. Provan and S. Atkins, “Tactical airport configuration management,” Integrated Communications, Navigation, and Surveillance Conference Proceedings, Herndon, VA, USA, 2011, 10.1109/ICNSURV.2011.5935346.

[2] R. Oseguera-Lohr, N. Phojanamongkolkij, G. Lohr, and J. Fenber, “Benefits assessment for tactical runway configuration management tool,” Aviation Technology, Integration, and Operations Conference, Los Angeles, CA, 2013, https://doi.org/10.2514/6.2013-4395.

[3] J. Avery and H. Balakrishnan, “Data-driven modeling and prediction of the process for selecting runway configurations,” Transportation Research Record, 2600, 2016, https://doi.org/10.3141/2600-01.

[4] D. Bertsimas, M. Frankovich, and A. Odoni, “Optimal selection of airport runway configurations,” Operations Research, 59(6), 2011, pp. 1407–1419, https://www.jstor.org/stable/41316045.

[5] L. Li, J. Clarke, H. Chien, and T. Melconian, “A probabilistic decision-making model for runway configuration planning under stochastic wind conditions,” IEEE/AIAA 28th Digital Avionics Systems Conference, 2009, 10.1109/DASC.2009.5347528.

[6] A. Jacquillat, A. Odoni, and M. Webster, “Dynamic control of runway configurations and of arrival and departure service rates at JFK airport under stochastic queue conditions,” Transportation Science, 51(1), 2016, pp. 155–167, https://doi.org/10.1287/trsc.2015.0644.

[7] S. Badrinath, M. Li, and H. Balakrishnan, “Integrated surface–airspace model of airport departures,” Journal of Guidance, Navigation, Control, and Dynamics, 42(5), 2019, pp. 1049–1063, https://doi.org/10.2514/1.G003964.

[8] X. Bai and P. Menon, “Decision support for optimal runway reconfiguration,” Aviation Technology, Integration, and Operations Conference, Los Angeles, CA, 2013, https://doi.org/10.2514/6.2013-4397.

[9] R. S. Sutton and A. G. Barto, Introduction to Reinforcement Learning. Cambridge, MA: MIT Press, 2018. http://incompleteideas.net/book/the-book-2nd.html.

[10] P. Razzaghi et al., “A survey on reinforcement learning in aviation applications,” ArXiv, 2022, https://arxiv.org/abs/2211.02147.

[11] C. Browne et al., “A survey of Monte Carlo tree search methods,” IEEE Transactions on Computational Intelligence and AI in Games, 4(1), 2012, 10.1109/TCIAIG.2012.2186810.

[12] C. Watkins, “Learning from delayed rewards,” PhD Thesis, University of Cambridge, England, 2019, https://www.cs.rhul.ac.uk/~chrisw/thesis.html.

[13] C. Watkins and P. Dayan, “Q-learning,” Machine Learning, 8, 1992, pp. 279–292, https://doi.org/10.1007/BF00992698.

[14] V. Mnih, K. Kavukcuoglu, and D. Silver, “Human-level control through deep reinforcement learning,” Nature, 518, 2015, pp. 529–533, https://doi.org/10.1038/nature14236.

[15] R. Raju, R. Mital, B. Wilson, K. Shetty, and M. Albert, “Predicting runway configurations and arrival and departure rates at airports: Comparing the accuracy of multiple machine learning models,” IEEE/AIAA 40th Digital Avionics Systems Conference, San Antonio, TX, USA, 2021, 10.1109/DASC52595.2021.9594389.

[16] M. Ahmed, S. Alam, and M. Barlow, “A multi-layer artificial neural network, approach for runway configuration prediction,” Proceedings of the ICRAT, 8th International Conference on Research in Air Transportation, Castelldefels, Catalonia, Spain, 2018, https://api.semanticscholar.org/CorpusID:198188623.

[17] S. Khater, J. Rebollo, and W. Coupe, “A recursive multi-step machine learning approach for airport configuration prediction,” AIAA Aviation Forum, 2021, https://doi.org/10.2514/6.2021-2406.

[18] A. Churchill, W. Coupe, and Y. Jung, “Predicting arrival and departure runway assignments with machine learning,” AIAA Aviation Forum, 2021, https://doi.org/10.2514/6.2021-2400.

[19] T. G. Puranik, M. Memarzadeh, and K. M. Kalyanam, “Predicting airport runway configurations for decision-support using supervised learning,” IEEE/AIAA 42nd Digital Avionics Systems Conference (DASC), 2023, 10.1109/DASC58513.2023.10311186.

[20] S. Levine, A. Kumar, G. Tucker, and J. Fu, “Offline reinforcement learning: Tutorial, review, and perspective on open problems,” ArXiv, 2020, https://arxiv.org/abs/2005.01643.

[21] A. Kumar, A. Zhou, G. Tucker, and S. Levine, “Conservative Q-learning for offline reinforcement learning,” ArXiv, 2020, https://arxiv.org/abs/2006.04779.

[22] M. Memarzadeh, T.G. Puranik, K.M. Kalyanam, and W. Ryan, “Airport runway configuration management with offline model-free reinforcement learning,” AIAA SciTech Forum, 2023, https://doi.org/10.2514/6.2023-0504.

[23] K. T. Tejasen, P. Raju, W. Ryan, A. Jordan, and R. Yang, "Defining an initial classification scheme for non-deterministic AI technologies," 2022 Integrated Communication, Navigation and Surveillance Conference, pp. 1–10, https://www.doi.org/10.1109/ICNS54818.2022.9771499.