Article

Responsible AI for Air Traffic Management: Application to Runway Configuration Assistance Tool

Milad Memrazadeh [1,*], Zili Wang [2], Farzan Masrour Shalmani [3], Pouria Razzaghi [4], Krishna Kalyanam [1]

[1] Aviation Systems Division, NASA Ames Research Center
[2] NASA OSTEM Intern, Iowa State University
[3] Crown Consulting, Inc., NASA Ames Research Center
[4] Metis Technology Solutions, Inc., NASA Ames Research Center
* Correspondence: milad.memarzadeh@nasa.gov

Abstract: The complexity and magnitude of airspace operations are ever increasing, which creates new challenges for the air traffic controllers. With the increase in volume of operations, the size of available data is also increasing. Data-driven AI solutions can provide actionable information for complex decision-making processes that controllers face and assist them in improving the efficiency and safety of the operations. However, for such solutions to be trusted by the users and stakeholders, they need to go through a comprehensive validation process. In this paper, the literature in the development of responsible AI is studied and a subset of the framework is applied to an AI tool proposed for airport runway configuration management. The focus of this study is on the two main challenges of: (1) detection and mitigation of existing *bias* in the training data and the trained AI tool; and (2) quantification and improvement of the AI tool's *robustness* to potential sources of noise in the data. We validate several responsible AI techniques in three major US airports and quantify their effectiveness in reducing the detected bias and improving the robustness of the model performance to adversarial noise in the input data.

Keywords: responsible AI; AI decision support tools; air traffic management.

1. Introduction

The National Airspace System (NAS) of the United States is facing an ever-increasing trend in both quantity and complexity of the airspace operations. This means that the air traffic controllers (ATCOs) are facing unprecedented increase in workload. On the other hand, the above-mentioned trend has produced a rising amount of operational data that characterize an untapped potential for developing automated solutions to assist ATCO on a daily basis. With the rise in the data, one area of focus for the research community has been the development of Artificial Intelligence (AI)/Machine Learning (ML) based solutions to aid the complex decision-making processes of the ATCOs.

Data-driven AI solutions can provide actionable information for complex decision-making problems that ATCOs face and assist them in improving the efficiency and safety of airspace operations. For such solutions to be trusted by the potential users and other stakeholders, they need to go under a comprehensive validation processes, before they can be deployed in real-world operations. As a result, one valuable area of research is developing a responsible AI framework, specifically tuned for decision-making processes in Air Traffic Management (ATM). In general, responsible AI refers to the practice of designing, developing, and deploying AI solutions in a way that is ethical, transparent, accountable, and fair. This includes several core principles and concepts such as explainability, interpretability, bias and fairness, robustness, satisfaction and safety, and transparency [1,2].

The main contribution of this paper is to adopt the general responsible AI framework, and tune it for its application in the ATM domain. Hence, only a subset of the principles that are the most relevant for development of AI tools in this domain are emphasized. As the first step, we focus on two core principles that are most relevant to ATM. Our contributions include deploying and evaluating the effectiveness of several techniques to address: (1) detection and mitigation of bias in input data and model's decision-making mechanism; and (2) robustness of the model's output to the potential sources of noise in the input data. This paper specifically focuses on the Runway Configuration Assistance (RCA) tool [3,4], developed in collaboration with retired controllers and Subject Matter Experts (SMEs) to assist ATCOs in airport runway configuration decision-making. The RCA tool is chosen because it is one of the more mature tools that has been validated against historical data in multiple airports across the NAS and is the ideal candidate for quantifying responsible AI techniques.

1.1. Runway Configuration Management

Runway configuration management involves selecting the optimal runways and directions of operation for aircraft arrivals and departures. Every airport, depending on its geometry and number of runways, has a set of unique configurations that can be used by the ATCO. The main factors affecting the choice of runway configuration are surface winds, traffic load, meteorological conditions (e.g., cloud ceilings and visibility), and operational procedures (e.g., noise abatement). Once an airport is in an active runway configuration, choosing the optimal time-window to switch the configuration to a different one is a crucial yet challenging decision that ATCOs make multiple times every day. A sub-optimal selection of the runway configuration or poor timing of configuration changes can result in significant increase in taxi times for aircraft on the surface of the airport, unnecessary aircraft holding (in the air or on the ground, causing long delays), and undesired go-arounds for arriving traffic in the air.

The Runway Configuration Assistance (RCA) tool [3] is an AI solution developed to address the runway configuration management problem. The RCA tool leverages an offline model-free reinforcement learning (RL) method, called conservative Q-Learning (CQL) [5] to learn a near-optimal policy for runway configuration management solely based on historical data. It is important that learning takes place offline, because there is no simulator available to mimic real-world operations for training AI solutions in an online fashion.

RCA tool is validated against historical data from three major US airports, namely, Charlotte Douglas International Airport (CLT), Dallas Fort Worth International Airport (DFW), and Denver International Airport (DEN) [4]. These three airports represent a broad range of complexity associated with runway configuration management, with CLT being "low", DFW being "mid", DEN being "high". The RCA tool demonstrated significant performance improvement over several baseline methods such as supervised learning [6] and offline implementation of deep Q-Network (DQN) [7]. In this paper, a generic framework is developed for (1) detection and mitigation of existing bias in the training data as well as the trained AI tools, and (2) quantification and improvement of the AI tool's robustness to different sources of noise in the training data. Then, it is applied to the RCA tool and quantify its effectiveness.

2. Related Work

Significant research has been done in both areas of bias mitigation and robustness analysis of AI/ML models. Our goal is to only review a subset of the literature that is most relevant to the topic of this study, and we refer the reader to recent surveys for more comprehensive reviews [8–10]. Bias refers to the systemic and unfair influence of factors and

variables that are used in a model development that can lead to incorrect or discriminatory outcomes and predictions. There are several sources that can cause a systemic bias in a trained AI/ML service, among with the most relevant ones are: (1) data bias, arising from the quality, quantity, or the representativeness of the training data; (2) feature bias, caused by incomplete data or bias in the developed feature engineering/selection algorithms; (3) model bias, that are caused by human decisions and affect the design and implementation of the AI/ML services; (4) algorithm bias, arising from a specific algorithm that is used in the training; and (5) operational bias, resulting from deployment, usage, or interpretation of the models. In this paper, we mostly focus on detection and mitigation of data, feature, and model biases.

Several statistical techniques can be used to detect bias in the AI/ML services. Such bias might be due to the imbalance in the distribution of the training data, class membership, and/or feature attributes. The bias can also exist in model's predictions, where fairness metrics such as demographic parity and equalized odds or statistical techniques such as Chi-square test, t-test, and ANOVA can be used to detect it. For example, equalized odds assess whether the predictions are consistent across different features and group memberships of a specific feature, while Chi-square test can determine whether the differences in the model predictions across different groups of a feature are statistically significant. In the case of runway configuration management, a feature can be wind direction, and specific group membership of this feature can be north-ward wind. Sometimes an AI service is biased on a feature among all available in the input data, while other times, the bias exists only in a specific group membership of features. Details regarding different features within the input data and their group memberships for the case study in this paper will be discussed later.

Once bias is detected, different techniques can be used to mitigate it and make the model more robust to other potential sources of bias. Generally, bias mitigation techniques are divided into three categories, depending on when they are applied: (1) pre-processing: which mitigates bias in the training data before it reaches the model; (2) in-processing: which mitigates bias while training a model; and (3) post-processing: which mitigates bias of a trained model as the after-the-fact analysis. Among these three categories, pre-processing and in-processing techniques are significantly more popular and utilized [9]. As a result, we build on pre-existing work and focus mainly on these categories. Among these two, pre-processing techniques are model-agnostic and thus can easily be applied to a wide variety of models without modification of the underlying algorithms, making them versatile techniques for bias mitigation.

Relabeling is a popular pre-processing technique, which analyzes parts of the data where there is a suspicion that the ground-truth labels might be erroneous, or show a large variance, and correct their labels. This is specifically applicable to the ATM domain and the RCA tool, because historical data shows great variations among the decisions made by different ATCOs, and this technique can mitigate such variance in the training labels before the models are trained on the data. For example, we observe that in a specific circumstance (including hour of the day, meteorological conditions, and traffic load), different runway configurations are used by different ATCOs. This could be due to several reasons including the subjectivity of the decision-making process, or effect of other operational variables that are not present in the data used for training of the AI tool. As a result, such variance in the labels of the data would cause a distraction in the training process. An extreme example of such variation is observed in DEN, where ATCO decisions on runway configuration show around 50% variation [11]. Relabeling identifies such data samples, reviews their circumstance, and corrects their labels. Several techniques have been proposed in the

past to apply relabeling, among which the most popular ones are messaging (a ranking technique to determine best candidates for relabeling) [12,13], and k-nearest neighbors [14].

Another popular pre-processing technique is sampling, which is widely utilized for bias mitigation [15–17]. These techniques change the distribution of the samples in the training data to increase/decrease impact of certain classes/features/groups of data and re-weight their importance during the training process. Such re-distribution of the importance among the training data, would result in the model being less biased on the majority representation in the training data. Perturbation [30,31] is another technique that can both reduce the bias of the model and improve its robustness to sources of noise. Perturbation techniques induce small noise into the training data to quantify its effect on the model's output. Such perturbations can be further included in the training data to improve the robustness of the model to those sources of noise. These perturbations can be random and/or adversarial. Adversarial noises usually use the parameters of the trained AI model to introduce the smallest amount of noise in the input data, which results in the greatest change in the model's output. As a result, they are considered among the most aggressive sources of perturbation that can hurt the trained AI model.

In-processing techniques mitigate bias during the training process of the AI/ML tool. One of the most popular methods in this category is regularization, which is a technique proposed to penalize discrimination or significant reliance on a specific feature or group membership within the input data [23]. Regularization primarily serves to prevent a model from overfitting on training data and generalize better to unseen data in the operational setting [24], however, it has shown to reduce the amount of bias in model's predictions as well. Among other popular in-processing techniques are representation learning and adversarial learning. Representation learning uses data transformation (either linear or non-linear) to remove the bias from the input data, while maintaining as much information as possible [18–22]. On the other hand, adversarial learning uses concepts of game theory and puts the AI tool in competition with an adversary (a separate AI model) that specifically tries to induce bias in the original AI tool, and through simultaneously training of the two models, reduces the overall bias [25–29].

3. Methodology

In this section, the RCA tool is briefly discussed first, Then, we introduce a general statistical technique to detect bias in existing models and implement several bias mitigation techniques to reduce the existing bias in the RCA tool. Lastly, a comprehensive analysis is performed to quantify the robustness of the model to different sources of noise and the effect of adversarial training on improving the model's robustness.

3.1. RCA Tool

The Runway Configuration Assistance (RCA) tool [11] is a previously developed AI solution based on the family of offline model-free RL [34] to provide decision-support for ATCOs in runway configuration management. It frames the problem as a Markov Decision Process (MDP) [35], where the state space S is comprised of relevant information for the ATCO decision-making such as wind conditions, meteorological conditions (cloud ceiling and visibility), traffic load (scheduled arrival and departure aircraft), and time of the day. Given this information, a policy for runway configuration selection, $\pi : S \rightarrow A$, can be computed based on maximizing a long-term expected utilities (or rewards):

$$\begin{aligned} V^{\pi}(s_t) = &\max_{\pi} f_U(s_t, \pi(s_t)) + \\ &+\gamma \sum_{s_{t+1} \in S} p(s_{t+1} \mid s_t, \pi(s_t)) V^{\pi}(s_{t+1}) \end{aligned} \tag{1}$$

where $\gamma \in [0, 1)$ is a discount factor, which tunes the importance of future utilities in comparison to the current ones and is treated as a hyperparameter. $p(s_{t+1} \mid s_t, \pi(s_t))$ is the probability of starting in state s_t, taking action $\pi(s_t)$, and ending up in the state s_{t+1} according to the transition (dynamics) function $f_T : S \times A \to S$, that defines the dynamics of the states. $f_U : S \times A \to \mathbb{R}$ is the utility function that is specifically designed with feedback from the SMEs. It includes traffic throughput, average transit times on the surface of the airport, and number of go-arounds as a result of high winds. For further details regarding the utility function, refer to [11]. If the AI agent has full knowledge of all components of the MDP, then dynamic programming [35] can be used to find the optimal policy from Eq. (1). However, in many real-world applications such as ATM, the transition (dynamics) function is complex, and as a result not feasible to learn from limited historical data. Furthermore, any error introduced in learning the model (or simplifying the dynamics), would affect the quality of the policy obtained and its applicability in practice. As a result, the RCA tool relies on a family of offline model-free control, specifically a state-of-the-art Conservative Q-Learning (CQL) algorithm [5] to learn a near-optimal policy solely based on historical data.

3.2. Bias Detection

We develop a general statistical method, similar to Chi-square test, to quantify the bias. The developed method is generic and can be applied to any supervised learning or RL tasks, and hence applicable to the RCA tool. Let us consider: (1) the input data $X \in \mathbb{R}^{N \times F}$, where N is the number of instances of data, and F is the number of features (e.g., wind conditions, arrival traffic, etc.); (2) a trained model $M_\theta : X \to \hat{Y}$, where θ are the parameters of the model, and $\hat{Y}$ is its outputs (predictions); and (3) the ground-truth labels (e.g., optimal runway configurations) for each input data, i.e., Y. It is assumed that any of the features in the input data can be categorized into groups based on the range of values that they take on. For example, the wind direction is discretized into 8 bins each categorizing 45-degrees. Figure 1 illustrates the group memberships for the wind direction as an example. Further details regarding the features in the training data and their group memberships are provided later in Section 4.1. Lastly, one needs to select a performance metric to quantify the performance of the model, where in here, $F1$-score (harmonic mean of precision and recall) is used as the performance metric of choice.

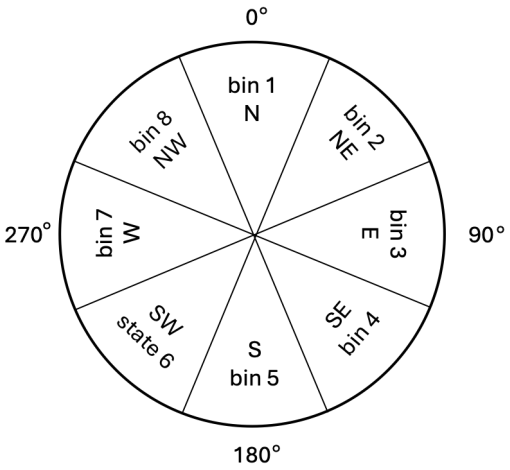

Figure 1. Discretization of wind direction into 8 bins.

Algorithm 1 shows the pseudo-code for the statistical technique developed to quantify the bias. In line 3, the baseline performance of the trained model is calculated, M_θ, on each specific group memberships of each feature $f \in F$, i.e., X_g^f, where $g \in G_f$, a set containing

all groups of feature f. For example, f is wind direction, and g is a specific group within this feature, that can be the East-ward winds (bin 3 in Figure 1). Lines 4-7 calculate the expected performance for each group memberships of a feature, if the group assignment was completely random. For example, if for 10% of the data, the wind direction group membership was East, we randomly assign another 10% of the input data to take East group membership for wind direction and calculate the performance of the model. This step is repeated a large number of times to calculate the expected value, μ_g, and the standard deviation, σ_g, of the expected performance in line 8. Finally, in lines 9-14, if the actual performance of the model, $F1^*_g$, is between the confidence bound of $\mu_g - 2\sigma_g$ and $\mu_g + 2\sigma_g$, the concolusion is that there is no statistically significant bias for the group g of feature f. On the other hand, if the performance is outside of this bound, an existing negative or positive bias is quantified, depending on what side of the bound the performance lies. If the performance is below the lower limit, then the model is performing significantly worse for the specific group, and is considered as negative bias, while if the performance is above the upper limit, the performance is better than expected, hence the name of positive bias.

Algorithm 1 Bias detection method

Input: data $X \in \mathbb{R}^{N \times F}$, trained model M_θ, and labels Y
Output: no/negative/positive bias

1: **for** each feature $f \in F$ **do**
2: **for** each group membership $g \in G_f$ **do**
3: calculate $F1^*_g = \texttt{eval}(M_\theta(X^f_g), Y_g)$
4: **for** $i = 1, ..., I$ **do**
5: randomly permute group memberships in X^f
6: calculate $F1^i_g$ based on new memberships
7: **end for**
8: calculate μ_g and σ_g as mean and standard deviation of $F1^i_g \quad \forall i = 1, ..., I$
9: **if** $\mu_g - 2\sigma_g \leq F1^*_g \leq \mu_g + 2\sigma_g$ **then**
10: there is no bias for group g
11: **else if** $F1^*_g < \mu_g - 2\sigma_g$ **then**
12: there is a negative bias for group g
13: **else if** $F1^*_g > \mu_g + 2\sigma_g$ **then**
14: there is a positive bias for group g
15: **end if**
16: **end for**
17: **end for**

3.3. Bias Mitigation

Among the numerous techniques that exist for bias mitigation, we summarized the most relevant ones in section 2. In this section, the implementation details of those techniques are explained and they are applied to the RCA tool to mitigate its existing bias and improve its robustness.

The first implemented technique is relabeling, as it is an efficient technique to improve the quality of the training data. Relabeling adjusts the ground-truth labels for data entries that might be erroneous or outliers. This could be due to error in data entries, and or variations in subjective decisions that are present in the historical data. The k-nearest neighbors algorithm is adopted for this purpose, where for each data instance in the training data, we find k closest neighbors ($k = 5$ is assumed in this paper) to it and adjust its ground-truth labels if the following two conditions are met: (1) at least 80% of the nearest neighbors agree on a different label (i.e., runway configuration); and (2) the average distance of the nearest neighbors to the data instance is less than a threshold (this threshold is defined based on the nature of training data and domain knowledge). Once the training

data are relabeled, a model is trained using the modified data and is evaluated to quantify the mitigated bias.

Next, we implement two independent sampling techniques to evaluate their effectiveness in mitigating bias. The first one is an importance sampling scheme designed to balance the class distribution of the training data (referred to as *Class Blc* in figures). Table 1 provides the class distribution among the data for the three airports, i.e., CLT, DEN, and DFW. The class balancing method assigns weights to data instances in the data loader during the training, with the weights being inversely proportional to the class relative frequency. Intuitively, this gives data from under-represented classes a higher probability of being sampled during training, which improves their representativeness. The second balancing technique focuses on increasing the representativeness of the specific group memberships of features in the data that show negative bias in the baseline model's performance (referred to as *Feature Blc* in figures). We refer to the trained model on the original data without any bias mitigation as the baseline model.

Configuration [Arr/Dep]	Usage [%]
CLT	
N/N	60.8
S/S	39.2
DEN	
SE/SE	18.8
S/S	15
N/NEW	14.5
S/SEW	12.6
N/N	12.3
NE/NE	11.7
NW/NW	8.6
SW/SW	3.4
E/E	1.6
NS/EW	1.2
W/W	0.3
DFW	
SSE/S	61.5
NNW/NNW	21.3
S/S	7.6
N/NNW	5.1
NNW/N	3
N/N	1.1
SSE/NNW	0.2
NNW/S	0.1
NW/NW	0.1

Table 1. Major runway configurations and their frequency based on 2018-2019 data for CLT, DEN, and DFW.

Lastly, we implement regularization, which is one of popular techniques used to mitigate bias and also address overfitting. The RCA tool is a multi-layer feed-forward neural network, and as a result, dropout [24] is utilized to implement regularization in training of the model. We use 50% rate of dropout in each layer of the Q-network.

3.4. Robustness and Adversarial Training

We mainly adopt adversarial training to provide a systemic approach for quantifying and improving the robustness of the RCA tool. Different sources of noise or error can

affect the performance of the model and this could be due to inaccuracies in the source data, limitations of sensor precision, etc. The focus of the adversarial training is to improve model's robustness to different sources of noise and increase its truth-worthiness. Three different types of noise are implemented for the purpose of experiments in this paper. The first and most general source of noise is random noise; given input data X, perturbed data $\tilde{X}$ is created as follows:,

$$\tilde{X} \sim \texttt{Unif}(X + e, x - e) \tag{2}$$

Here, `Unif` is the uniform distribution, and e is the magnitude of the perturbation.

The other two techniques are adversarial noises, that use the parameters of the trained model to create a bounded amount of perturbation in the input data that would result in the most deviation in its output. As a result, they are a stronger perturbation to the input data (compared to the random noise) and directed attacks to hurt the model's performance. If random noise can be thought to represent data inaccuracies in the average case, then adversarial noise can be regarded as worst-case scenarios. The first adversarial noise is Fast Gradient Sign Method (FGSM) [32], which is a one step gradient-based algorithm. FGSM generates an adversarial example by adding a small perturbation to the input data that maximizes the loss function, and is defined as follow,

$$\tilde{X} = X + e \cdot \texttt{sign}(\nabla_x J(\theta, x, y)) \tag{3}$$

Where $\texttt{sign}(\cdot)$ is the sign function that computes the sign of each component in the input vector, and $\nabla_x J(\theta, x, y)$ is the gradient of the loss function with respect to the input data.

The second methods is Projected Gradient Descent (PGD) [33], which is an iterative algorithm that extends the FGSM approach and is defined as follow,

$$\tilde{X}_{t+1} = \texttt{proj}_{B(x,e)}\left(\tilde{X}_t + \alpha \cdot \texttt{sign}(\nabla_{\tilde{x}_t} J(\theta, \tilde{x}_t, y))\right) \tag{4}$$

Here, $\tilde{X}_0 = X$, t is the iteration counter, $\texttt{proj}_{B(x,e)}$ is the function that projects the perturbed data onto the ball of radius e centered at X, and α is the learning rate. The projection operation ensures that the perturbation remains within the desired threshold of error. PGD can be seen as a stronger attack than FGSM due to its iterative nature of refining the perturbation to find a more effective attack.

First, the robustness of the RCA tool to these sources of noise in the input data is quantified. Then, the training data is augmented with perturbed data and a new model is trained from scratch on the augmented data to evaluate the effectiveness of the adversarial training in improving the robustness of the model.

4. Results and Discussion

In this section, we summarize the findings of applying bias mitigation and adversarial training techniques to the RCA tool and quantify their effectiveness.

4.1. Data

FAA's Aviation System Performance Metrics (ASPM) reports [1], NASA's Sherlock Data Warehouse [2], METeorological Aerodrome Reports (METARs) [3], and Terminal Aerodrome Forecast (TAF) reports [4] are fused to create a dataset for training the RCA tool. We use two

[1] https://aspm.faa.gov/
[2] https://sherlock.opendata.arc.nasa.gov/sherlock_open/
[3] https://aviationweather.gov/data/metar/
[4] https://aviationweather.gov/data/taf/

years of data (2018-2019) from the three major US airports (CLT, DEN, and DFW) for the purpose of experimental validation in this paper.

The data is aggregated by a quarter of an hour, meaning each year results in 35,040 instances of data, so overall we have 70,080 data for each airport. The features included in the analysis are hour of the day, wind conditions (direction and speed), cloud ceiling, visibility, arrival demand, and departure demand. Wind direction is categorized into eight bins according to Figure 1, wind speed is categorized into six bins (0-5, 5-10, 10-15, 15-20, 20-25, >25 knots), cloud ceiling is categorized into five bins (0-2500, 2500-7500, 7500-15000, 15000-30000 feet, and no ceiling), visibility is categorized into five bins (0-3, 3-6, 6-7.5, 7.5-9, $\geq$10 miles), and arrival/departure demands are categorized into four bins (0-10, 10-20, 20-30, >30 aircraft). To properly evaluate the effectiveness of the implemented techniques, the entire data is divided into 80% training and 20% testing sets, randomly, and each of the techniques are applied only to the training set, keeping the testing set representative of the real-world unseen data.

4.2. Bias Mitigation

The main goal of bias mitigation in the case of RCA tool, is to mitigate and decrease the negative bias, i.e., cases where the performance is significantly lower than expected. This is due to the fact that higher than expected performance in the ATM domain is not considered an adverse bias. We measure the amount of negative bias as the difference between the actual performance of the baseline model, $F1^*_g$, and the lower bound of expected performance (refer to Algorithm 1), $\mu_g - 2\sigma_g$, i.e., negative bias $= \min\left(\mu_g - 2\sigma_g - F1^*_g, 0\right)$. Figure 2 shows the average existing negative bias for each feature (the average is performed across different group membership of a feature), for the three airports of CLT, DEN, and DFW. In this figure, *Baseline* refers to the original model with no bias mitigation technique applied and values closer to zero represent less detected negative bias.

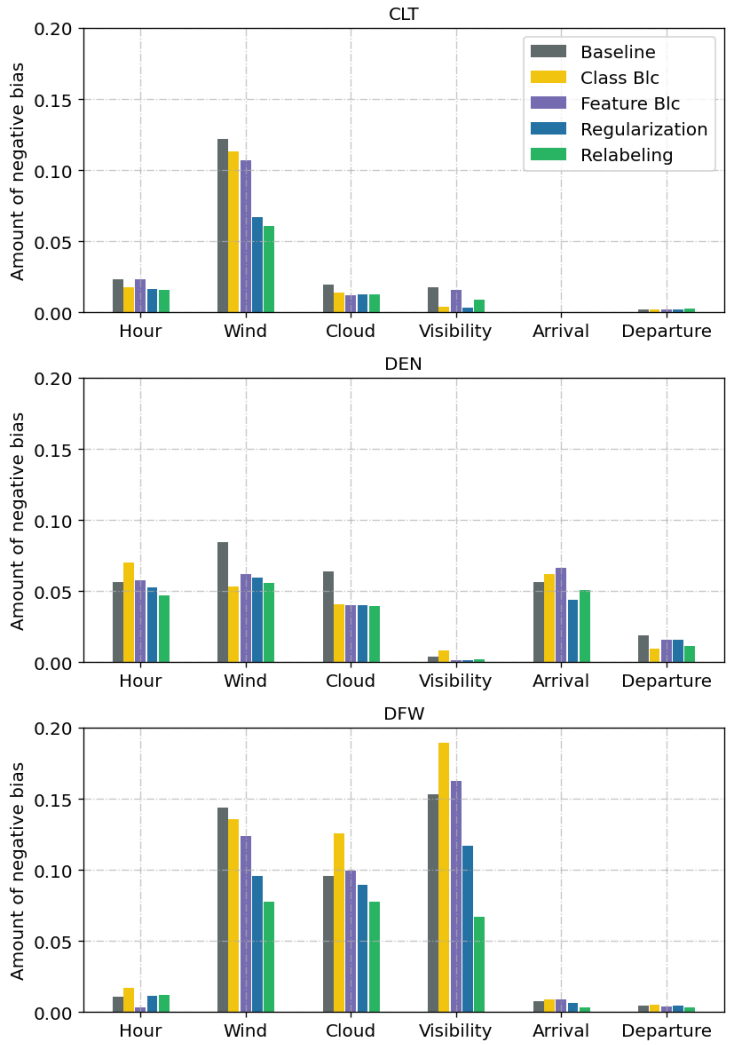

Figure 2. Mitigation of negative bias based on different techniques.

Overall, we see some similarities and differences between the three airports. CLT, being an airport with the least complex runway configuration among the three, shows only a significant negative bias in wind conditions. This is intuitive, since wind is the major factor affecting the choice of runway configurations. On the other hand, DEN and DFW, being the more complex airports, show significant existing bias in most of the features. For example, the hour of the day, and arrival demand show significant bias for DEN, whereas for CLT and DFW, they represent negligible bias. On the other hand, the meteorological conditions such as wind, cloud ceiling, and visibility show significant negative bias for DFW. This difference between DEN and DFW could be due to the significant variation in the weather and operational conditions at DEN on an hourly basis, which is related to the location of the airport. This difference is also evident in the usage distribution reported in Table 1, where we can see that six configurations at DEN are used more than 10% of the time, while there are only two configurations at DFW that are utilized more than 10%.

Average changes in the existing bias imposed after applying each of the techniques are reported in Table 2. The most effective technique for reducing bias is relabeling, followed by regularization. Two main messages can be inferred from this finding: (1) the effectiveness of the relabeling technique emphasizes the subjectivity and variability of the ATCO's decision-making, which is present in the historical operational data; it further emphasizes that the ground-truth labels for the data in the case of runway configuration management are highly variable, error-prone and need careful review; lastly, the variability among the

ATCO's decisions might be emphasizing the importance of other features that are affecting the decision-making process, but are not included in the input data of the RCA tool due to unavailability, e.g., noise abatement procedures, operational procedures, etc. and (2) alleviation of bias by regularization shows that the RCA tool might be overly relying on specific features or specific group memberships within a feature that is causing bias in the predictions; increasing the size of data, and/or features used in the analysis, along with the use of regularization techniques should alleviate this source of bias.

Feature	Class Blc	Feat. Blc	Reg.	Relab.
Hour	+18%	**-23%**	-11%	-14%
Wind	-17%	-17%	-36%	**-43%**
Cloud	-11%	-23%	-26%	**-30%**
Visibility	+20%	-22%	**-55%**	-53%
Arrival	+20%	+25%	-29%	**-39%**
Departure	-13%	**-14%**	-6%	-12%
Average	+3%	-12%	-27%	**-32%**

Table 2. Percentage changes in the existing bias by each technique. The percentages are averaged for the three airports.

Although there still remain significant amount of negative bias, even after applying a bias mitigation technique, we noticed that the majority of such bias exist in calm wind conditions (wind speeds less than 10 knots). Figure 3 shows the visualization of existing bias (green for no bias, red for negative bias and blue for positive bias), before (left panel) and after (right panel) applying relabeling technique to the RCA tool in the cases of CLT (top panels), DEN (middle panels) and DFW (bottom panels). In this figure, wind direction are shown as a radian plot (degrees from North), wind speed is shown as the distance from the center of the circle, and sizes of dots represent the amount of data that is in the training set for each category.

As it is illustrated in the figure, most of the existing negative biases under high wind conditions (more than 10 knots) have resolved after applying relabeling technique for all airports, and majority of the existing negative bias lies in the realm of calm wind conditions. This is a major finding, since for runway configuration management, key decisions are the ones that happen in the high wind conditions, and the choice of runway configuration in the calm wind conditions are not as important. A lot of other factors can affect the decision-making in the calm wind conditions that are not present in the input data, such as operational procedures, noise abatement procedures, airline preferences, and proximity of the runways to the terminals. Hence, the aggregate mitigation of bias reported in this paper represent the mitigation in general setting, while in the cases of high wind conditions, the mitigation effects are more significant. Another observation in the case of CLT is that majority of the remaining negative bias is in the cases of eastward and westward winds, which due to the north-south orientation of the runways, the model has no mechanism of addressing those cases.

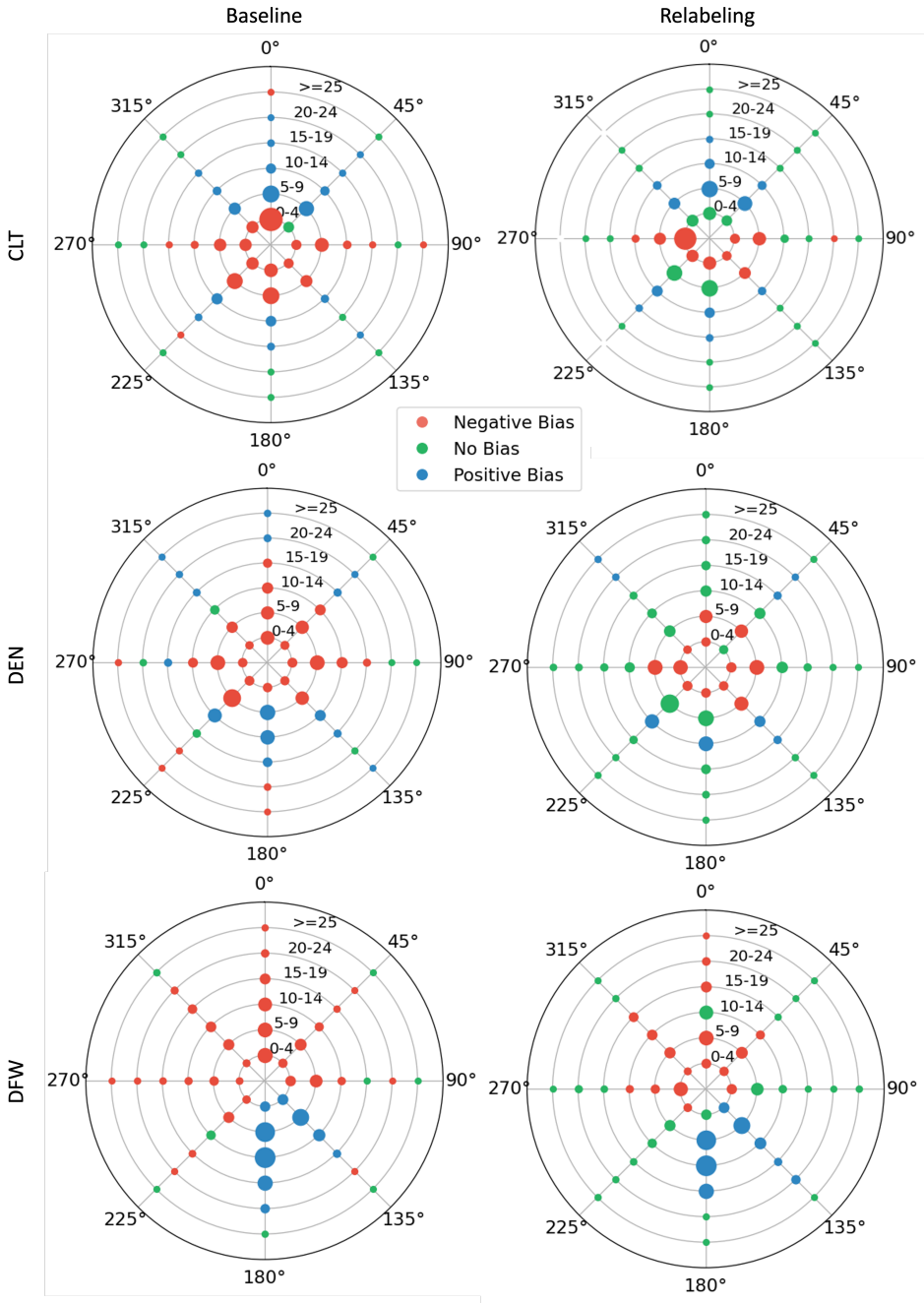

Figure 3. Illustrating existing bias in wind conditions.

4.3. Adversarial Training

In this section, we quantify the robustness of the RCA tool to random noise and two types of adversarial noise, FGSM and PGD. Then, the perturbed data are included in the training process of the RCA tool and the improvements in the robustness of the model to these sources of noise is quantified. Figure 4 (top panel) shows the drop in performance of the RCA tool (in terms of increase in the cross entropy loss) as a function of the magnitude of each source of noise, e, as defined in Eqs. (2-4). As expected, PGD has significantly more impact on the model performance compared two the other two approaches followed by FGSM. In Table 3, three stages of the drop in the performance of the model are identified, i.e., 5%, 10%, and 25% and find the smallest magnitude level of each noise that causes such drop in the performance (these values are average among the three airports).

Next, we add the perturbed data to the original training set and train a new model on the augmented set and evaluate its effectiveness on the robustness of the model to the

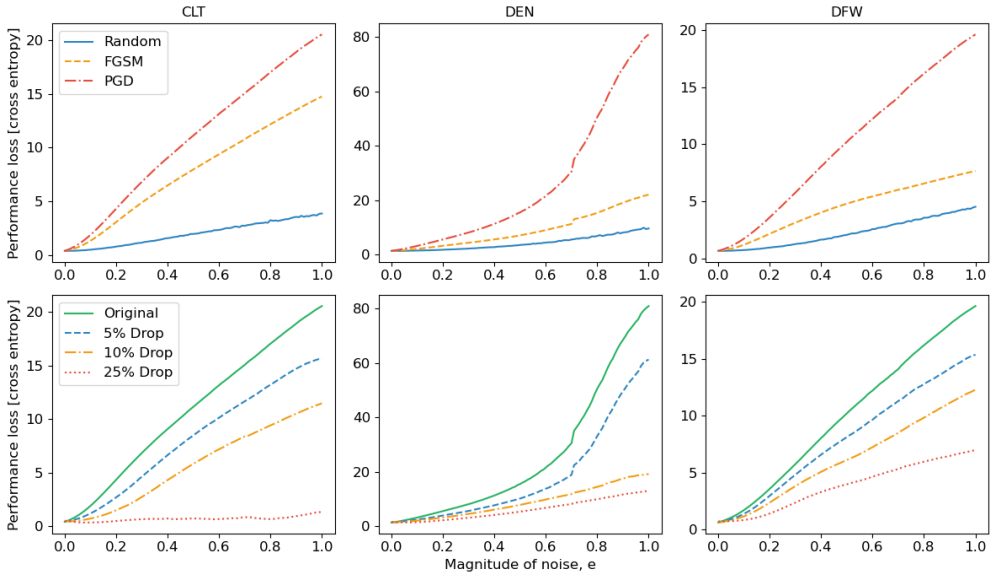

Figure 4. This figure shows (top pane): the effect of different sources of noise on the performance of the RCA tool; and (bottom panel): the improvements made in the robustness of the model by adding perturbed data (according to PGD method) to the training process.

Method	5% drop	10% drop	25% drop
Random	0.14	0.2	0.4
FGSM	0.02	0.04	0.09
PGD	0.02	0.04	0.08

Table 3. Quantified magnitude of each source of noise for specific drop in the model's performance.

sources of noise. Figure 4 (bottom panel) quantifies this effect for the case of augmenting data with PGD perturbation compared to the original model that does not contain perturbed data in the training. For example, in the case of 10% Drop noted in the figure, the training set is perturbed with magnitude $e = 0.04$ of PGD noise (according to the Table 3), and a new model is trained on the augmented training set that contains the original data plus the perturbed ones. Once the model is trained, its performance is evaluated on the test set.

Overall, we observe that perturbing the training data with noise of some sort (random and/or adversarial) and including the perturbed data in the training of the RCA tool improves its robustness to such noises significantly. This effect is more evident in the case of perturbing the training data with a higher magnitude of PGD noise. This finding might allude to the inherent sources of noise that exist in the operational historical data and shows that including noisy data in the training of the AI tools, have positive impacts on their robustness to noise in operations. As a result, researchers and practitioners in the field of ATM, should not aim to remove all sources of noise and uncertainty from their training data.

5. Conclusions

In this paper, the literature in responsible AI, its different components and principles, and their applicability to the domain of air traffic management is studied. The developed framework mainly addresses two fundamental challenges: (1) bias detection and mitigation in both training data and developed AI model; and (2) model's robustness to different sources of noise. In order to show-case the effectiveness of the developed framework, we applied it to a previously developed AI decision-support tool for runway configuration management. Using real-world historical data (years 2018-2019) from three major US

airports (CLT, DFW, and DEN), it is shown that (1) simple and scalable bias mitigation techniques such as relabeling and regularization are effective in mitigating existing bias in the AI tool, while not compromising performance. The performance improvement of the relabeling technique further emphasized the importance of data reviews that need to be done either through automated approaches or by the subject matter experts for the ground-truth data in the domain of air traffic management; and (2) adversarial training has significant effect on improving the robustness of the AI solutions to various sources of adversarial noise. Although the effectiveness of this technique varies in different airports, they showed an overall improvement to the robustness of the model, and hence, are recommended to be adopted by the AI developers in this domain. This paper is a first step towards developing a unified framework for implementation of responsible AI principles on the developed AI solutions in the domain of air traffic management.

Author Contributions: MM and KK designed the concept and initiated the project. FMS and MM developed the bias detection/mitigation methodologies and PR implemented it on the tool and obtained results. MM and ZW developed the adversarial training methodologies and ZW implemented the robustness analysis and obtained the results. All authors contributed to the writing of the paper.

Data Availability Statement: The datasets presented in this article are not readily available because of restrictions that exist on the data. Requests to access the datasets should be directed to the corresponding author.

Acknowledgments: The authors acknowledge the invaluable support and feedback received from subject matter experts affiliated with the Federal Aviation Administration's (FAA) Office of NextGen (ANG).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Federal Aviation Administration, "Roadmap for artificial intelligence safety 618 assurance" 2024.
2. B. Stahl, "Embedding responsibility in intelligent systems: from AI ethics to responsible AI ecosystems." Scientific Reports, 13, 05, 2023.
3. M. Memarzadeh, T.G. Puranik, K.M. Kalyanam, and W. Ryan. "Airport runway configuration management with offline model-free reinforcement learning." AIAA Scitech 2023 Forum, 2023.
4. K.M. Kalyanam, M. Memarzadeh, J. Crissman, R. Yang, and K. TJ Tejasen. "Applying machine learning tools for runway configuration decision support." 11th International Conference on Research in Air Transportation (ICRAT), 2024.
5. A. Kumar, A. Zhou, G. Tucker, and S. Levine. "Conservative q-learning for offline reinforcement learning." ArXiv, 2020.
6. T.G. Puranik, M. Memarzadeh, and K.M. Kalyanam. "Predicting airport runway configurations for decision-support using supervised learning." 2023 IEEE/AIAA 42nd Digital Avionics Systems Conference (DASC), 2023.
7. V. Mnih, K. Kavukcuoglu, and D. Silver. "Human-level control through deep reinforcement learning." Nature, 518:529–533, 2015.
8. M. Hort, Z. Chen, J.M. Zhang, M. Harman, and F. Sarro. "Bias mitigation for machine learning classifiers: A comprehensive survey." ACM Journal on Responsible Computing, 1:1–52, 2024.
9. S. Siddique, M.A. Haque, K.D. Gupta, R. George, D. Gupta, and Md.J.H. Faruk. "Survey on machine learning biases and mitigation techniques." MDPI Journal of Digital, 4:1–68, 2024.
10. T.P. Pagano, R.B. Loureiro, F.V.N. Lisboa, R.M. Peixoto, G.A.S. Guimaraes, G.O.R. Cruz, M.M. Araujo, L.L. Santos, M.A.S. Cruz, E.L.S. Oliveira, I. Winkler, and E.G.S. Nascimento. "Bias and unfairness in machine learning models: A systematic review on datasets, tools, fairness metrics, and identification and mitigation methods." Big Data Cogn. Comput., 7, 2023.
11. M. Memarzadeh and K.M. Kalyanam. "Runway configuration assistance: An offline reinforcement learning method for air traffic management." Journal of Aerospace Information Systems, 2024.
12. F. Kamiran and T. Calders. "Classifying without discriminating." IEEE 2nd international conference on computer, control and communication, 2009.
13. F. Kamiran and T. Calders. "Data preprocessing techniques for classification without discrimination." Knowledge and Information Systems, 33:1–33, 2012.

14. B.T. Luong, S. Ruggieri, and F. Turini. "k-nn as an implementation of situation testing for discrimination discovery and prevention." Proceedings of the 17th ACM SIGKDD, pages 502–510, 2011.
15. L.E. Celis, V. Keswani, and N. Vishnoi. "Data preprocessing to mitigate bias: A maximum entropy based approach." International Conference on Machine Learning, pages 1349–1359, 2020
16. J. Chai and X. Wang. "Fairness with adaptive weights." International Conference on Machine Learning, pages 2853–2866, 2022.
17. W. Du and X. Wu. "Fair and robust classification under sample selection bias." Proceedings of the 30th ACM International Conference on Information and Knowledge Management, pages 2999–3003, 2021.
18. R. Zemel, Y. Wu, K. Swersky, T. Pitassi, and C. Dwork. "Learning fair representations." International Conference on Machine Learning, pages 325–333, 2013.
19. E. Creager, D. Madras, J. Jacobsen, M. Weis, K. Swersky, T. Pitassi, and R. Zemel. "Fair and robust classification under sample selection bias." International Conference on Machine Learning, pages 1436–1445, 2019.
20. Y. Roh, K. Lee, S. Whang, and C. Suh. "Fr-train: A mutual information-based approach to fair and robust training." International Conference on Machine Learning, pages 8147–8157, 2020.
21. M.M. Kamani, F. Haddadpour, R. Forsati, and M. Mahdavi. "Efficient fair principal component analysis." Machine Learning, pages 1–32, 2022.
22. U. Gupta, A. Ferber, B. Dilkina, and G. Ver Steeg. "Controllable guarantees for fair outcomes via contrastive information estimation." Proceedings of the AAAI Conference on Artiicial Intelligence, 35:7610–7619, 2021.
23. T. Kamishima, S. Akaho, H. Asoh, and J. Sakuma. "Fairness-aware classifier with prejudice remover regularizer." Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pages 35–50, 2012.
24. N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. "Dropout: A simple way to prevent neural networks from overfitting." Journal of Machine Learning Research, 15:1929–1958, 2014.
25. N. Dalvi, P. Domingos, S. Sanghai, and D. Verma. "Adversarial classification." Proceedings of the tenth ACM SIGKDD, pages 99–108, 2004.
26. D. Lowd and C. Meek. "Adversarial learning". Proceedings of the eleventh ACM SIGKDD, pages 641–647, 2005.
27. V. Iosiidis, B. Fetahu, and E. Ntoutsi. "Fae: A fairness-aware ensemble framework." IEEE International Conference on Big Data, pages 1375–1380, 2019.
28. L. Oneto, M. Doninini, A. Elders, and M. Pontil. "Taking advantage of multitask learning for fair classification." Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society, pages 227–237, 2019.
29. T. Calders and S. Verwer. "Three naive bayes approaches for discrimination-free classification." Data Mining and Knowledge Discovery, 21:277–292, 2010.
30. M. Feldman, S.A. Friedler, J. Moeller, C. Scheidegger, and S. Venkatasubramanian. "Certifying and removing disparate impact." proceedings of the 21th ACM SIGKDD, pages 259–268, 2015.
31. J.E. Johndrow and K. Lum. "An algorithm for removing sensitive information: application to race-independent recidivism prediction." The Annals of Applied Statistics, 13:189–220, 2019.
32. I.J. Goodfellow, J. Shlens, and C. Szegedy. "Explaining and harnessing adversarial examples." ArXiv, 2014.
33. A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. "Towards deep learning models resistant to adversarial attacks." ArXiv, 2017.
34. S. Levine, A. Kumar, G. Tucker, and J. Fu. "Offline reinforcement learning: Tutorial, review, and perspectives on open problems." ArXiv, 2020.
35. R.S. Sutton and A.G. Barto. "Introduction to reinforcement learning." MIT Press, Cambridge, MA, 2018.