# Towards Fair and Explainable AI in Aviation: Case study on Runway Configuration

Pouria Razzaghi
*Metis Technology Solutions, Inc.*
*NASA Ames Research Center*
Mountain View, USA
pouria.razzaghi@nasa.gov

Kenny Chour
*Metis Technology Solutions, Inc.*
*NASA Ames Research Center*
Mountain View, USA
kenny.chour@nasa.gov

Milad Memarzadeh
*Aviation Systems Division*
*NASA Ames Research Center*
Mountain View, USA
memarzadeh.milad@gmail.com

Farzan Mansour
*Crown Consulting Inc.*
*NASA Ames Research Center*
Mountain View, USA
farzan.masrourshalmani@nasa.gov

Krishna M. Kalyanam
*Aviation Systems Division*
*NASA Ames Research Center*
Mountain View, USA
krishna.m.kalyanam@nasa.gov

***Abstract*—Artificial intelligence (AI) is playing an increasingly transformative role in Air Traffic Management (ATM), yet its integration raises critical concerns about fairness, transparency, and robustness. This study addresses these concerns through the lens of Responsible AI (RAI) in aviation by focusing on a decision-support model for runway configuration management. We propose and evaluate a suite of fairness-aware and explainable AI techniques applied to an offline reinforcement learning-based Runway Configuration Assistance (RCA) tool. First, we quantify model bias using a feature-sensitive F1-score permutation framework. Next, we implement two in-processing bias mitigation methods: a modified Meta Fair Classifier (MFC) adapted for multi-class classification, and a feature-wise adversarial debiasing approach that does not require predefined sensitive attributes. We further explore combinations of these with pre-processing strategies such as relabeling. In parallel, we enhance model transparency by employing two interpretability techniques: Layer-wise Relevance Propagation (LRP) and Kernel-SHAP. These methods enable insight into both the global and local behavior of the model. The proposed methods are validated using operational data from three major US airports. The results demonstrate significant improvements in fairness metrics and interpretability, with minimal compromise to predictive performance, supporting the case for responsible AI adoption in safety-critical aviation applications.**

***Index Terms*—Artificial intelligence, Responsible AI, Air Traffic Management, Bias mitigation, Explainability, Runway Configuration**

## I. Introduction

Artificial intelligence (AI), as the latest technology, is assuming an unavoidable role in the new configuration of Air Traffic Management (ATM) systems. In the case of entities, where air traffic is continually climbing, AI-enabled tools become hopeful choices for the solutions to make the operation more efficient and providing the possibility for decision making in real-time. However, the incorporation of AI technology into such safety critical application fields poses a number of concerns as to its robustness, reliability, transparency, and requirements regarding fairness. Responsible AI (RAI) in the context of ATM is designing, developing, and deploying AI models in ways that are in line with ethical principles, operational requirements, and regulatory standards. Unlike conventional AI systems, responsible AI is designed first to be more accountable in its actions and secondly to be more understandable and trustworthy for the human operators who must use its decision criteria. This introduction indeed highlights a number of main issues and resolutions on RAI in ATM but retains no specific information about the collision-avoidance systems. The RAI is a broader concept that covers other usage areas as well, such as traffic flow optimization, resource allocation, weather impact analysis, and workload management for air traffic controllers.

A responsible AI approach in ATM requires:

- Transparency and Explainability: AI systems should produce outputs that are human interpretable and understandable. In this way, decisions taken by AI systems can be traced and checked when necessary.
- Robustness and Reliability: The identified ATM, being a highly dynamic and safe environment, must ensure that AI systems will work even under weirdly adverse conditions and that unexpected events will not worsen the situation.
- Bias and Fairness: AI should comply with ethical policies in that it should show fairness and prevent biases that could be harmful to operations and stakeholders.

The careful following of these principles, in ATM, can lead to industry transformation, allowing more efficient and more adaptive operations while at the same time ensuring the trust and oversight of a human operator and regulatory body. This paper deals with methods and techniques for the development of RAI in different areas of ATM, ensuring that AI-driven solutions will positively influence the future of aviation. To address the challenges outlined above, this study introduces a set of contributions aimed at improving both the fairness

and interpretability of a machine learning (ML) model, as summarized in the following.

### A. Contribution

This study focuses on the development of responsible AI techniques for runway configuration management, an offline model-free reinforcement learning (RL) tool to create a Runway Configuration Assistance (RCA) providing decision support to air traffic controllers [1]–[3]. This study addresses two key pillars in responsible AI models: bias mitigation and explainability. It was underscored that the focus was on multi-label classification models. The contributions are listed below:

1) ***Bias Mitigation Techniques*** : One of the most common concerns about the outcome of an ML model is the bias of the results about particular specific groups. To mitigate bias in the RCA tool, we conduct different bias reduction techniques, including in-processing and pre-processing methods. Our contributions are threefold:
   - *Modifying Binary Bias Mitigation Techniques into Multi-Class Models*: Existing bias mitigation techniques such as the Meta Fair Classifier (MFC) [4] are developed for binary classification problems. In this work, we propose a methodology to extend these techniques for multi-class ML models like the RL model used in the RCA tool. By adapting MFC for multi-class classification, we demonstrate its ability to significantly reduce bias while preserving model performance.
   - *Nonsensitive adversarial debiasing*: We adapt the adversarial debiasing model (AdDE) [5] to be independent of pre-specified sensitive features by applying it to the whole space of input features. The adversarial method on a per-feature basis allows the model to learn invariant representations w.r.t. any individual feature, and hence enhance overall fairness without specific identification of sensitive variables.
   - *Combining Bias Mitigation Techniques*: In addition to in-processing methods, we employ a pre-processing technique such as relabeling to further address biases in the training data. We compare the impact of combining relabeling with MFC, AdDe, and dropout-based techniques. In this comprehensive evaluation, we compare the effectiveness of individual and combined bias mitigation approaches. Compared to standalone techniques, the combination of in-processing and pre-processing methods results in superior bias reduction.
2) ***Explainability Models*** : Explainability is critical in understanding how AI models make decisions. To enhance the transparency of the RCA tool, we applied two different explainability methods: LRP [6] and Kernel SHAP [7]. The methods provide insight into how different inputs affect the model's decisions and a better understanding of the AI system outcomes.

## II. Literature Review

As AI models become more complex, interpretability and transparency are critical, especially in safety-critical applications. The integration of AI into critical systems such as aviation requires consideration of fairness and explainability. This can ensure ethical and effective decision-making, improve operational efficiency, safety protocols, and the overall passenger experience. However, this transition is accompanied by notable concerns about biases embedded within AI systems [8].

Bias mitigation research in AI for aviation is centered on the development of effective methods to detect, minimize, and mitigate biases. This encompasses a systematic exploration of data representation, algorithmic fairness, and what the deployment of AI in the real world implies [9]. As air travel demands increase and aviation operations become increasingly complicated, the requirement for equitable AI solutions becomes essential to ensure operational integrity and public trust [10].

In addition, the concept of explainability in AI is important to enable transparency and accountability, particularly in decision-making models such as runway configuration. XAI gives insight into AI decision-making processes, enabling stakeholders to know the cause of the results and ensuring adherence to regulatory standards is the most important. By adding fairness and explainability, researchers hope to build artificial intelligence systems that foster collaborative interactions between machine intelligence and human knowledge, improving safety and operational efficiency in aviation [11]. As the aviation industry grapples with the complexities of AI implementation, an ongoing conversation about fairness and bias minimization strategies is imperative. These involve multidisciplinary practices that integrate social science and technological know-how. It is essential to develop sound policies that inform ethical practices in artificial intelligence. Dedicated to creating an equitable and transparent AI not only improves technological results but also imposes the core values of fairness and trust among aviation operations [12].

### A. Fairness

Fairness in AI is an important area of study, especially in models where decision-making is expected to have an impact on individuals and groups. The various types of bias that can arise during the implementation of AI, impacting the design, development, deployment, and evaluation of AI systems. Representation bias occurs when the training datasets for AI models do not reflect the diverse populations that they are intended to serve. The lack of diversity may cause the model to perform poorly for marginalized groups, since the model may not generalize well as a result of insufficient training data. If a dataset does not reflect the actual distribution of a population, the AI model derived from it can amplify existing inequalities or introduce new ones. Evaluation bias occurs when the AI model evaluation does not represent the population in consideration. This may occur when perfor-

mance metrics are not suitable or when the test datasets have as much balance as the training datasets.

Avoidance of bias in AI is a multi-faceted process that employs numerous various techniques at various stages of ML pipeline: pre-processing, in-processing, and post-processing. Each stage offers various opportunities to identify and alleviate bias [13].

- *Pre-processing*: methods aim at minimizing bias prior to feeding data into the machine learning system. This may involve enriching training sets for a more representative sample or employing transformations with the aim of altering biased data distributions [14]. It is essential that aviation technology leaders make data evaluation and quality improvement their key priority to improve fairness outcomes from the beginning.
- *In-processing*: techniques are utilized during the training phase of the model. Such methods are typically modifications to the algorithms for fairness-performance balancing. Recent studies demonstrate that such techniques can induce a fairness-performance tradeoff, whereby improvements in fairness happen at the expense of machine learning performance measures such as accuracy [15]. For example, research has demonstrated that the application of bias mitigation techniques can effectively lower the values of fairness metrics in various applications [16]; however, such changes can also lead to performance losses. Thus, the correct selection of suitable algorithms is necessary to achieve an optimal trade-off.
- *Post-processing*: methods calibrate the output of trained models to make them more fair after prediction. This may include adjusting decision thresholds or re-weighting the outputs according to protected attributes. These methods are effective in improving fairness; their performance can vary significantly depending on the specific model and context applied. Practitioners are therefore encouraged to carefully study the impacts of such changes on general model performance.

### B. Explainability models

Explainable Artificial Intelligence (XAI) is a term used for a range of tools and frameworks used to explain the predictions made by AI models. Although conventional ML models tend to be "black boxes," XAI seeks to improve transparency in decision-making processes and the drivers behind these decisions. Transparency could be vital due to the demand for full knowledge of AI decision-making processes.

In aviation, the use of artificial intelligence technologies requires a lot of attention to explainability, particularly given regulatory and safety issues. The explainability of model outputs is necessary not only to gain regulatory approval but also to achieve public trust in automated systems. As artificial intelligence systems make more critical decisions, the need for explainability becomes increasingly urgent.

In general, XAI techniques are classified into model-specific and model-agnostic techniques. Model-specific techniques are applied to specific architectures, such as Layer-wise Relevance Propagation (LRP) [6], which explains neural network predictions by propagating relevance scores backward through all layers to identify important input features. However, model-agnostic techniques like SHapley Additive exPlanations (SHAP) [7] and Local Interpretable Model-agnostic Explanations (LIME) [17], consider the model a black box and assess feature importance by analyzing the changes in output after the changes in the input feature. Gradient-based techniques, such as Integrated Gradients and Grad-CAM [18], utilize the internal gradients of the model to highlight the most influential regions of the input used to generate a prediction, and these techniques are commonly applied to image and text classification tasks. Additionally, surrogate models, i.e. decision trees or rule-based models, are often trained to be approximations of intricate models and to produce human-comprehensible rules. Trade-offs among interpretability fidelity, computational cost, and generalizability to different model classes are present within each approach, and the choice of an XAI approach usually depends on the application, model architecture, and degree of explanation required.

## III. Methodology

### A. RCA tool

Runway configuration management is the procedure to optimally choose active runways and their directions of use to accommodate arriving and departing flights. The collection of possible configurations differs by airport, based on the airport layout, number of runways, and their geometric alignments. The primary drivers for the selection of runway configurations include surface wind directions and traffic volume. The RCA Tool is an offline model-free reinforcement learning known as conservative Q-Learning (CQL) [19]. The tool exploits the offline capability of the CQL algorithm to learn a near-optimal policy for managing runway configurations from historical data alone. The validity of the RCA tool is tested with historical data from three major airports in the National Airspace System (NAS), namely Charlotte Douglas International Airport (CLT), Dallas Fort Worth International Airport (DFW), and Denver International Airport (DEN). In this work, we apply a variety of RAI methods, focusing specifically on bias mitigation and model explainability, on the RCA tool. We begin by measuring the bias in the current model through relevant fairness metrics. The bias in the RCA tool refers to the extent to which the model's decision-making is disproportionately influenced by specific input features, potentially leading to uneven or unfair outcomes across different operational scenarios. Next, we apply and compare two in-processing methods and examine their efficacy. Furthermore, we examine the effects of integrating pre-processing and in-processing approaches and evaluating their marginal and combined contributions towards fairness. Lastly, we employ explainability techniques on the RCA tool to render the model's predictions interpretable and comprehend the most significant input features, thereby enhancing the model's transparency and accountability.

### B. Bias quantification

We developed a general statistical method to quantify bias that can be applied to any supervised learning or RL tasks, and hence applicable to the RCA tool. This technique is described in detail at [20], and we present it here briefly. Let us imagine that we have the input data with different features (e.g., wind conditions, arrival traffic, etc.). After training the RCA model, for each input data, we have the output of the model (i.e., predicted runway configurations); and the ground-truth runway configuration, which is the historical decisions made by the controllers (ATCo). We assume that any of the features in the input data can be categorized into groups based on the range of values that they can get. For example, we discretize wind direction into 8 bins each categorizing 45-degrees. For example, a wind that is blowing from 22.5 to 67.5 from North is considered as a member of Northeast bin. Furthermore, we use F1-score (harmonic mean of precision and recall) as the performance metric. First, we calculate the baseline performance metric $(F1^*)$ of the trained model on each specific group memberships (e.g. East-ward winds) of each feature (e.g. wind direction). Then, we randomly permute the group memberships of a feature, while other feature are fixed and calculate the expected performance. We repeat this step a large number of times and calculate the expected value $(\mu)$, and the standard deviation $(\sigma)$. If the actual performance of the model $(F1^*)$, is between the confidence bound $(\mu - 2\sigma <= F1^* <= \mu + 2\sigma)$, we conclude that there is no statistically significant bias for that group. On the other hand, if the performance is below the lower limit, then the model is performing significantly worse for the specific group, and is what we call negative bias, while if the performance is above the upper limit, the performance is better than expected, hence the name of positive bias.

### C. Bias mitigation

To address concerns about fairness in ML predictions, here we employ different pre- and in-processing bias mitigation methods and also combinations of those individual methods. Fairness in ML refers to the principle that the outcomes of the model should not disproportionately disadvantage individuals or groups defined by sensitive attributes, such as race, gender, or age.

A comprehensive discussion of the pre-processing methods and their validation and also the regularization method are presented in [20]. For the purpose of supporting the analysis of combining pre-processing and in-processing techniques, a brief summary of the pre-processing approaches is provided below. In this paper, we perform two in-processing bias mitigation methods: the Meta Fair Classifier (MFC) and the Adversarial Debiasing with the objective of reducing biased outcomes associated with sensitive attributes. In-processing methods integrate fairness constraints within the learning algorithm itself, allowing the model to improve both predictive performance and fairness during training. Unlike pre- and post-processing solutions, in-processing techniques offer higher control over feature representation and the decision boundary and hence are specially well-suited for imposing fairness measures like equalized odds or demographic parity. The technical derivation and implementation of MFC and Adversarial Debiasing are presented below, and contributions of each of these methods to mitigate group-level differences without undermining model performance are discussed.

*1) Pre-processing Methods:* Here, we provide a summary of the bias mitigation approaches taken in the RCA tool, first presented in [20]. The first approach is relabeling, which improves the quality of the training data by rectifying possible label noise and inconsistencies resulting from data entry errors or subjective past judgments. We took a k-nearest neighbors (k-NN)-based strategy, where the label of each instance is revised if (1) at least 80% of its 5 nearest neighbors have a different label and (2) their average distance is less than a pre-specified threshold derived from domain knowledge. A model was trained on the relabeled data, while an independent, untouched test set was reserved for evaluation in order to maintain the original label distribution.

We also utilized two sampling-based pre-processing techniques. One is an importance sampling scheme that reweights instances inversely to their class frequency, promoting class balance representation during training. The second aims at group-aware reweighting, which makes underrepresented or disadvantaged feature groups more visible, discovered by baseline model bias analysis.

As an in-processing approach, we applied dropout regularization [21] to the structure of the RCA tool feed-forward neural network. Dropout alleviates both model bias and overfitting by randomly dropping out a fraction of neurons during training, which encourages the model to learn more generalizable and stable representations.

*2) Meta Fair Classifier:* To ensure fairness in the predictions, we adopt the MFC method, which is particularly used in binary classification settings. In practice, common classifiers often show bias by optimizing solely for accuracy, unintentionally favoring the majority group. To mitigate this, one can add fairness constraints to the training objective. These constraints enforce group-level fairness metrics such as demographic parity $P(\hat{Y} = 1 \mid A = 0) = P(\hat{Y} = 1 \mid A = 1)$, equal opportunity (equal true positive rates) or equalized odds (equal true and false positive rates between groups). However, meeting these constraints poses a trade-off with predictive performance. The Meta Fair Classifier enforces fairness during learning by modifying the optimization goal to incorporate both accuracy and fairness. It is model-agnostic, it can be appended with any binary classification method (e.g., logistic regression or neural networks) throughout training. MFC formulates a meta-objective function:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{classifier}} + \lambda \cdot \mathcal{L}_{\text{fair}} \tag{1}$$

where $\mathcal{L}_{\text{classifier}}$ is the standard classification loss (e.g., cross-entropy), $\mathcal{L}_{\text{fair}}$ is a fairness penalty loss from the desired constraint (e.g., demographic parity gap), and $\lambda$ is a hyper-parameter controlling the trade-off. Throughout the training process, MFC learns a decision and representation boundary

that balances these factors effectively. Instead of heuristically modifying, MFC uses meta-learning to generalize fairness, thereby enabling flexible adaptation to the fairness definition most applicable to the specific application domain. MFC is used to debias a binary classifier model for decision-making applications where fair treatment between groups is critical. The strength of this method lies in its in-processing design, which allows direct control over model optimization; its ability to be flexible enough to fit varied definitions of fairness; and its applicability to a very wide range of classifiers. However, MFC is inherently limited to binary classification tasks, and adapting it to the multi-class scenario or regression problems adds more complexity. Also, while MFC can reduce group-level disparities, the trade-off with overall accuracy must be monitored carefully.

To apply MFC method on the RCA tool model as a multi-class ML model, we develop a methodology. To quantify differences in model performance across different categorical feature values, we compute a category-wise fairness loss that encapsulates within-feature inconsistency. We begin by calculating the per-category loss for every unique value for all categorical features in the input matrix. Specifically, for every feature, we iterate over its unique categories and calculate the subset of samples that fall under each category. For these subsets, we compute a performance metric—cross-entropy loss—between model predictions and true labels only over the samples belonging to that category. Formally, let $f_j$ denote the $j^{\text{th}}$ categorical feature with values $\mathrm{c}_1, c_2, \ldots, c_k$ and let $\mathcal{I}_i \subset \{1, \ldots, n\}$ represent the indices of samples where $f_j = c_i$. For each category $c_i$, we compute:

$$\mathcal{L}_i = \text{CrossEntropy}\left(\hat{Y}_{\mathcal{I}_i}, Y_{\mathcal{I}_i}\right) \tag{2}$$

where $\hat{Y}_{\mathcal{I}_i}$ and $Y_{\mathcal{I}_i}$ denote the predicted and actual labels for samples in category $c_i$, respectively. Next, we quantify the intra-feature disparity by computing the range (max-min) of the per-category losses within each feature:

$$\mathcal{L}_{\text{feature}} = \max_i(\mathcal{L}_i) - \min_i(\mathcal{L}_i) \tag{3}$$

which captures the extent of variation in predictive performance across the feature's categories. This step is repeated for all categorical features, resulting in a vector of per-feature disparity scores. Finally, we compute the overall fairness loss by taking the mean of the per-feature disparity values:

$$\mathcal{L}_{\text{fair}} = \frac{1}{m}\sum_{j=1}^{m} \mathcal{L}_{\text{feature}_j} \tag{4}$$

where $m$ is the total number of categorical features. This aggregate loss function penalizes the model for uneven performance across categories, encouraging more equitable predictive behavior. NaN values arising from categories with insufficient samples are masked out before computing the mean to ensure robustness. At the end, we add this fairness loss to the loss function in the training process of the RCA tool.

*3) Adversarial debiasing:* Adversarial debiasing is an in-processing fairness technique that utilizes adversarial learning concepts to minimize biases while training an ML model. Inspired by the Generative Adversarial Networks (GANs) architecture, this approach adds another adversarial component to remove sensitive information from the representations produced by a main predictive model. The main idea is to train the predictive model in such a way that its output obscures any information about sensitive attributes, thereby reducing the potential for biased decision-making. In contexts where no predefined sensitive attribute is available, feature-wise adversarial debiasing can be employed to iteratively reduce the influence of each individual feature. The key goal is to limit the ability of any adversary to reconstruct or infer a particular input feature from the model's prediction, resulting in a more feature-agnostic and fair model. The architecture comprises two key components: the primary model $f_\theta(x)$, which predicts a target value $\hat{y} = f_\theta(x)$, and a set of adversarial models $\{g_{\phi_i}\}$, each associated with a particular input feature $x_i$. Each adversarial model $g_{\phi_i}$ is trained to predict feature $x_i$ from the output $\hat{y}$, while the primary model is trained to minimize both its task loss and the ability of adversaries to infer any feature. To accomplish this, the total loss for the primary model is defined as:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{task}} - \gamma \sum_{i=1}^{d} \mathcal{L}_{\text{adv},i} \tag{5}$$

where $\mathcal{L}_{\text{task}}$ is the standard loss (e.g., mean squared error between predictions and targets), $\mathcal{L}_{\text{adv},i}$ is the adversarial loss for feature $x_i$, and $\gamma$ is a hyperparameter controlling the strength of debiasing. In practice, adversarial models can be implemented as regression networks for continuous features or classification models for discretized features. The primary model is thus trained not only to fit the true targets, but also to degrade the performance of each adversary, promoting invariance to all features. The training process typically involves the utilization of gradient reversal layers (GRL) or alternating optimization methods. In each iteration, a forward pass is performed to obtain the prediction $\hat{y}$, which is then evaluated by each adversarial network to identify adversarial losses. A GRL is used to flip the gradient signal from the adversarial models before back-propagating to update the parameters of the main model, so that the adversarial objective is minimized on the adversary's end but maximized on the main model's end. After training, fairness can be measured by estimating the adversaries' ability to predict each feature; less predictive power means better feature obfuscation and improved fairness.

In this study, in order to handle bias in situations when sensitive attributes are not well-defined, we employ a generalized adversarial debiasing method, which considers all input features as potential sources of bias. The technique attempts to decrease the model's dependence on any single feature by introducing an adversarial objective during training. The adversarial component attempts to identify feature-specific information from the model's internal representations, while the primary model is concurrently trained to hinder such

inference, thereby allowing for the learning of more feature-agnostic representations. Formally, assume that the base model $f_\theta$ maps an input state $s \in \mathbb{R}^d$ to a distribution of action values $Q(s, a)$. The model has an internal representation layer $h = \phi_\theta(s)$, where $\phi_\theta$ is the encoder or hidden feature extractor. To promote fairness, an adversary $g_\psi$ is trained to predict information about individual features $s_i$ from the internal representation $h$. The adversarial loss is defined as:

$$\mathcal{L}_{\text{adv}} = \mathbb{E}_{s\sim\mathcal{D}}\left[\sum_{i=1}^{d} \mathcal{L}_{\text{adv},i}\left(g_\psi^{(i)}(\phi_\theta(s)), s_i\right)\right] \tag{6}$$

where $\mathcal{L}_{\text{adv},i}$ is a regression or binary loss depending on whether feature $s_i$ is continuous or categorical. Instead of using separate adversarial models for each feature, the implementation uses a shared adversary that is learned from the model's internal representations and output. The training objective of the primary model combines three distinct components: the CQL loss for conservative Q-learning, the Bellman error for value learning, and the adversarial loss, regulated by a hyperparameter $\lambda$, for avoiding feature leakage. The composite loss function can be expressed as:

$$\mathcal{L}_{\text{total}} = \alpha \cdot \mathcal{L}_{\text{CQL}} + \beta \cdot \mathcal{L}_{\text{Bellman}} + \lambda \cdot \mathcal{L}_{\text{adv}} \tag{7}$$

where $\alpha$, $\beta$, and $\lambda$ regulate the relative significance of each term. Notably, gradient reversal or alternating optimization is used so that the adversary is trained to maximize its predictive capability, whereas the main model is optimized to minimize it, thus imposing orthogonality between features and internal representations. This feature-based adversarial debiasing method allows for overall bias removal without pre-knowledge of which features are sensitive. It is particularly valuable in exploratory settings, where unseen sources of bias may be present in the data. In our scenario, we also employ class reweighting to mitigate class imbalance in action labels and train the adversary to minimize its prediction error with respect to these class-weighted inputs. It is evaluated by tracking both adversarial loss and model performance metrics across training iterations. The adversarial loss, when minimized correctly by the model and maximized by the adversary, indicates that feature-specific information has been effectively obscured in the learned representation, promoting fairness across all dimensions.

### *D. Explainability*

In decision-making problems, the transparency and interpretability of ML models are important to uphold trust and accountability. Due to black-boxish nature of ML models, it is hard to understand the reasoning of their predictions. To address that, XAI methods have been developed for explaining model decisions either by examining internal computations or estimating feature contributions externally. This paper presents two different XAI techniques: Layer-wise Relevance Propagation (LRP) and Kernel-SHAP to interpret the predictions of the model and identify the input features that contribute most to the predictions. While LRP provides an explanation based on the internal network structure, Kernel-SHAP provides a model-agnostic, theoretically grounded explanation based on cooperative game theory. Together, they offer a more complete view of model behavior and facilitate model validation and fairness testing.

*1) Layer-wise Relevance Propagation :* It is an interpretative approach particularly designed for neural networks. LRP is a method that works by redistributing the model's prediction score in a backward direction through all the layers to the input features, thereby assigning a relevance score to all features based on its contribution to the final output. Unlike gradient-based methods, LRP relies on the relevance conservation principle, which keeps the overall relevance intact across the different layers, where the total relevance is:

$$f(x) = \sum_i R_i \tag{8}$$

where $f(x)$ is the model output for input $x$, and $R_i$ is the relevance of the feature. Backward propagation of relevance uses special redistribution rules, such as the $\epsilon$-rule or $\alpha - \beta$-rule, which prevent numerical instability and that positive and negative contributions are treated correctly. In the present work, LRP is applied to a trained deep neural network for predictive analytics. The network output is recursively decomposed layer by layer all the way back to the input layer, hence enabling the detection of features (or time segments, in the case of time-series data) making important contributions towards a specific prediction. The outcome is a relevance heatmap, which can be plotted to emphasize which inputs had the most influence for single predictions. This is especially useful in areas where model transparency is essential, as it gives insight into how the model applies input data to arrive at decisions—beyond merely feature ranking. The strength of LRP lies in its fidelity to the structure developed by the model, making it suitable for the evaluation of feature interactions and contributions in neural models. However, LRP is model-specific and requires full access to the network architecture and weights, which diminishes its broad applicability to different types of model. However, it is a complement to external methods like SHAP by providing a mechanistic interpretation based on the internal dynamics of the model.

*2) Kernel-SHAP:* Kernel-SHAP is a model-agnostic explainability approach grounded in Shapley value concepts from cooperative game theory. Kernel-SHAP quantifies the contribution of each input feature towards an individual prediction by estimating the marginal contribution of the feature for all feature subset combinations. Formally, for a model f, the Shapley value for feature i is defined as:

$$\phi_i = \sum_{S\subseteq N\setminus\{i\}} \frac{|S|!(|N| - |S| - 1)!}{|N|!}\left[f(S \cup \{i\}) - f(S)\right] \tag{9}$$

where $N$ is the set of all features and $S$ is a subset of features not containing $i$. As exact computation becomes exponentially challenging with an increase in the number of features, Kernel-SHAP approximates Shapley values through weighted linear

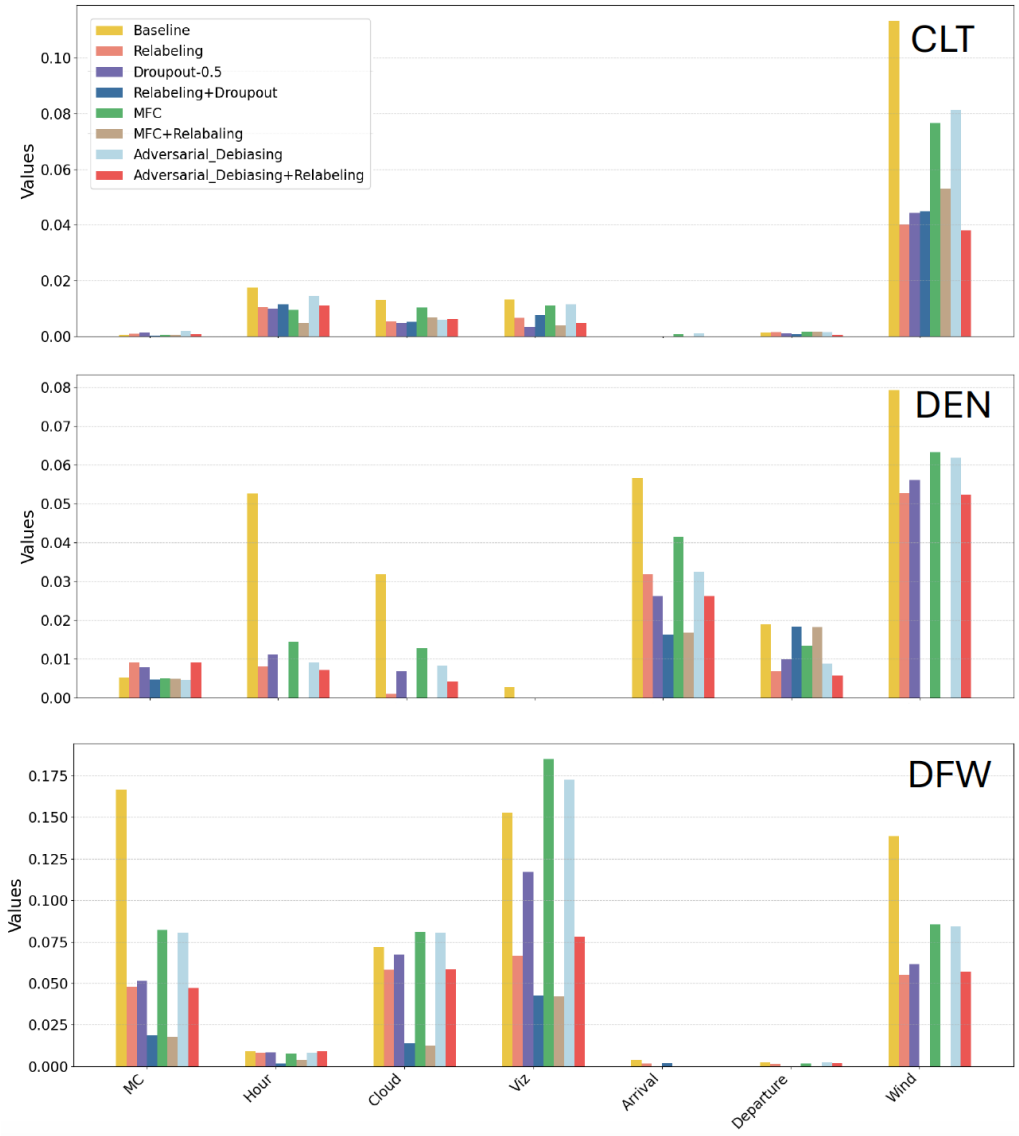


Fig. 1: Mitigation of negative bias based on different techniques. Baseline refers to the original model with no bias mitigation technique applied. The value on the y-axis is the summation of all differences of negative bias with the lower bound of expected performance ($|\mu - 2\sigma - F1^*|$).

regression over randomly selected feature subsets, making estimation computationally viable in practical applications.

In this study, Kernel-SHAP is applied to the trained model to compute feature attributions for individual predictions. It does so by perturbing the input features and observing changes in the model output, fitting a local linear surrogate model around each instance. The outputs include SHAP values, which are additive and locally accurate—ensuring that the sum of all feature attributions equals the predicted value. Visualizations, such as force plots, summary plots, and beeswarm plots, are used to communicate the features that contribute positively or negatively to a prediction, and the extent of consistency of these contributions across instances. Kernel-SHAP boasts the benefit of theoretical guarantees, such as consistency and local accuracy, and can be used for a wide range of black-box models, from tree ensembles to support vector machines to neural networks. It does presume feature independence and may entail high computational overhead, especially in high-dimensional settings. Despite these shortcomings, Kernel-SHAP provides comprehensible, intuitive, and faithful explanations of model behavior, thus being a good complement to model-specific methods such as LRP.

## IV. Results

In this section, we provide the outcomes of applying bias mitigation and explainability techniques to the RCA tool and discuss their results.

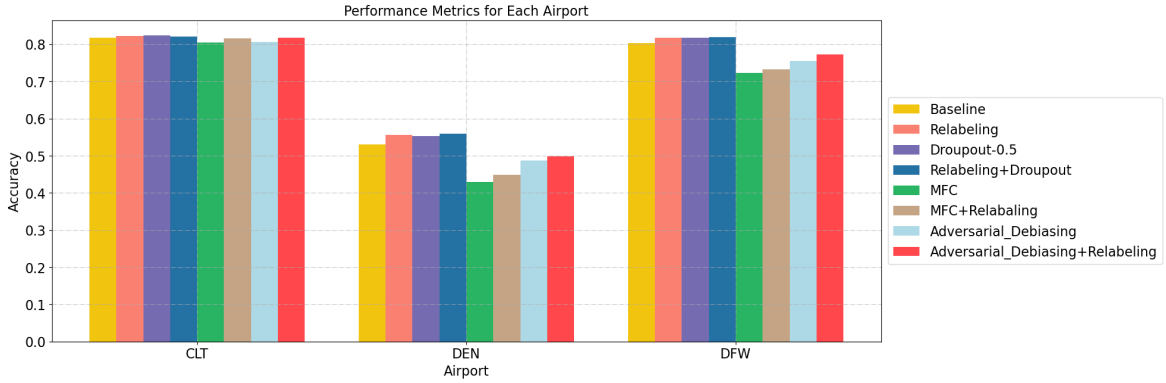


Fig. 2: Performance comparison of bias mitigation techniques across airports.

### A. Bias mitigation

To accurately estimate the performance of bias mitigation approaches, we split the entire data into 80% train and 2% test, and apply each of the approaches, on the training set only, keeping the test set an actual representative of the unseen data. The general goal of bias reduction in the case of the RCA tool is to minimize and counter the negative bias, where the performance is significantly lower than the expectation. For each instance exhibiting negative bias, we compute the magnitude of the bias as the difference between the actual performance of the baseline model, $F1^*$ , the lower bound of expected performance (see Algorithm 1 in [20]), $\mu - 2\sigma$, i.e., negative bias $= \min(\mu - 2\sigma - F1^*, 0)$. We add these differences for a total bias score and note the indices of the affected data points. After applying the bias mitigation techniques, we recompute the lower bound difference of $F1$ at both the previously identified locations and at any new locations with negative bias that may occur. The total summed difference after mitigation yields a residual bias measure. To get a visual sense of the impact of bias reduction, we show a bar plot comparing the total negative bias observed before and after applying the debiasing method. The comparison depicts the effectiveness of different methods in decreasing the incidence and magnitude of a feature-wise negative bias throughout the data.

Figure 1 shows the existing negative bias for each feature in three CLT, DEN, and DFW airports. The values closer to zero represent less detected negative bias and the bars missing in the plots means the zero value for the specific feature and technique. At CLT, where the airport has the simplest runway configuration of the three airports studied, there is a strong negative bias under conditions of changing winds. This result is not surprising, as wind is one of the primary operational variables that influence runway configuration selection. DEN—with the most complex runway layout—has significant bias across a more comprehensive set of features. For instance, hour of the day and arrival demand are highly biased at DEN. Similarly, the visualization results for DFW are highly biased as well, in addition to the wind condition. Among the tested mitigation strategies, the most effective approach to reduce bias is a combination of relabeling (as pre-processing) and both regularization and the MFC (as in-processing).

The results indicate that some of the bias inherent in the RCA tool may be due to over dependence on certain characteristics or group membership, which exert an unequal influence on the model decision-making process. The effect of each of the methods on overall performance of the model is shown in Fig. 2, where the variation in performance between different airports and approaches is captured. It is noteworthy that the use of relabeling and regularization did not only diminish the negative bias of the RCA tool but also preserved or even enhanced the model's overall performance. In contrast, the use of MFC, although good at bias mitigation, caused the predictive accuracy to plummet drastically, which we expect as the trade-off between proper balancing of fairness and performance goals.

### B. Explainability

*1) Layer-wise Relevance Propagation :* Figure 3 contains three heatmap-style visualizations, representing the LRP results for a fully connected neural network used in RCA with two hidden layers. These visualizations show how input features contribute to the output of the network for each airport. The relevance values range from low (purple) to high (yellow), as indicated by the color bar on the right. Most of the nodes are purple or blue, indicating low relevance, with a few yellow or orange nodes showing higher relevance. These visualizations provide insight into the interpretability of the network and also in which input features contribute to the predictions at each airport.

From the LRP visualizations, recognizable patterns are evident for all three airports (CLT, DEN, DFW), illustrating how the various input features contribute to the network's predictions across the hidden layers. The different shades of blue to yellow identify the most intense connections and nodes utilized in the prediction process for each airport.

*2) Kernel-SHAP:* The kernelExpaliner method can provide both the instance level (single instance) and the overall model explanation. The overall model evaluation was already discussed in [22]. The force plot of a local explanation of the predictions visualizes the contribution of each input feature

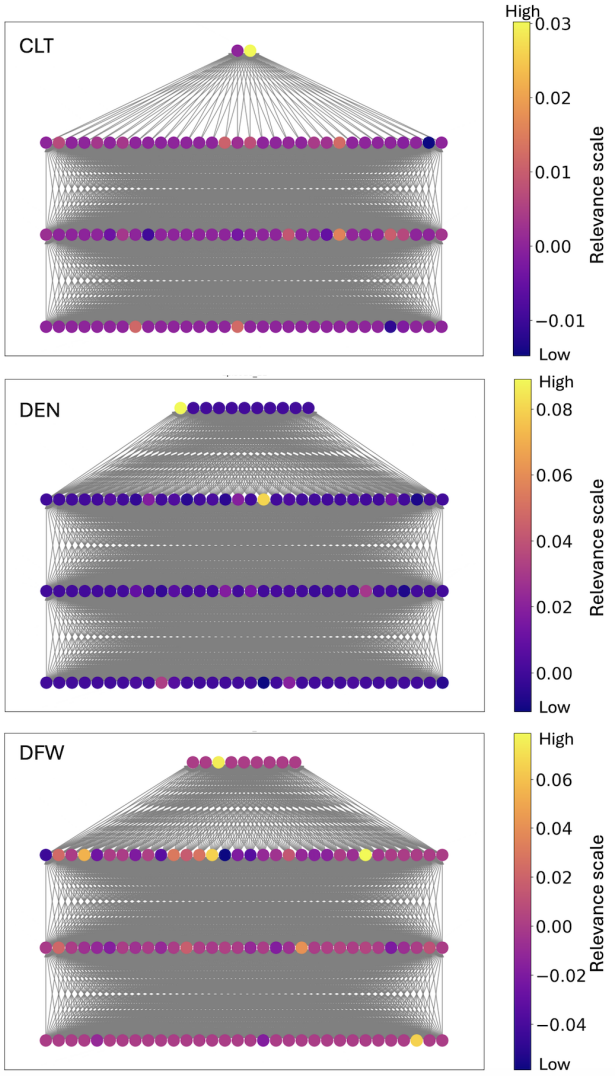


Fig. 3: Layer-wise score for each airport at a single instant during the training process. The bottom line of nodes are the inputs. The first 24 nodes are the hour of day feature. The next two are the arrival and departure rates, following by two wind components, cloud ceiling, visibility, and meteorological conditions. The top line of nodes is the decision nodes, which the number of the nodes are equal to the number of runway configurations for each airport.

relative to a baseline value. The baseline, often simply called the expected value, represents the average prediction output of the model across the training data. The contribution of each feature is communicated using a SHAP value, which quantifies the extent to which the specific feature moves the prediction away from the baseline. Features that raise the predicted value are represented by positive contributions (red color), and those that lower it are represented by negative contributions (blue color). The total of all SHAP values for an instance, added to the baseline, yields the model's final prediction for the instance. Additive decomposition in this manner facilitates the interpretation of single predictions, picking out the features that have the most influence on a model's decision and promoting transparency in high-stakes applications.

In Fig. 4, the single instances for each airport are explained in a way that reflects the significant effect on the decision of the RCA tool. The red color indicates the feature value pushing the prediction higher, and the blue feature pushes it lower. To be more specific, for example, in CLT, the baseline value $-0.061$ suggests that the model is not likely to choose the current configuration in this instance. The east wind component is the most significant factor that negatively influences the configuration decision. The model appears to steer away from the configuration due to this feature. The Cloud Ceiling has a positive impact, while visual meteorological condition exerts a minor negative influence on the decision. All other features, including hour of day, are neutral and insignificantly affect the model's decision.

## V. Conclusion

This paper presents a comprehensive framework for integrating fairness and interpretability into a machine learning-based runway configuration decision-support system. By applying responsible AI principles to the RCA tool, we demonstrate the practical application of bias mitigation and explainability methods in a safety-critical aviation context. Our contributions include the adaptation of Meta Fair Classifier to multi-class settings, the implementation of a feature-wise adversarial debiasing approach without requiring sensitive attribute specification, and the combination of in- and pre-processing methods for improved bias mitigation. In addition, the use of both LRP and Kernel-SHAP provides complementary interpretability perspectives, making the decision-making process more transparent. Empirical evaluation across three diverse airports shows that these techniques can reduce group-level disparities and enhance model interpretability with minimal performance degradation. The findings underscore the feasibility and necessity of adopting RAI frameworks to build fairer, more accountable AI systems in aviation. Future work will focus on real-time adaptation and further generalization of the proposed methods to other decision-making modules within ATM systems.

## Acknowledgment

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

## References

[1] M. Memarzadeh, T. G. Puranik, K. M. Kalyanam, and W. Ryan, "Airport runway configuration management with offline model-free reinforcement learning," in *AIAA SciTech 2023 Forum*, 2023, p. 0504.

[2] S. Nethi, M. Memarzadeh, and K. M. Kalyanam, "Optimization of runway configurations with forecast-augmented offline reinforcement learning," in *AIAA SCITECH 2024 Forum*, 2024, p. 0533.

[3] M. Memarzadeh and K. Kalyanam, "Runway configuration assistance: Offline reinforcement learning method for air traffic management," *Journal of Aerospace Information Systems*, pp. 1–13, 2024.

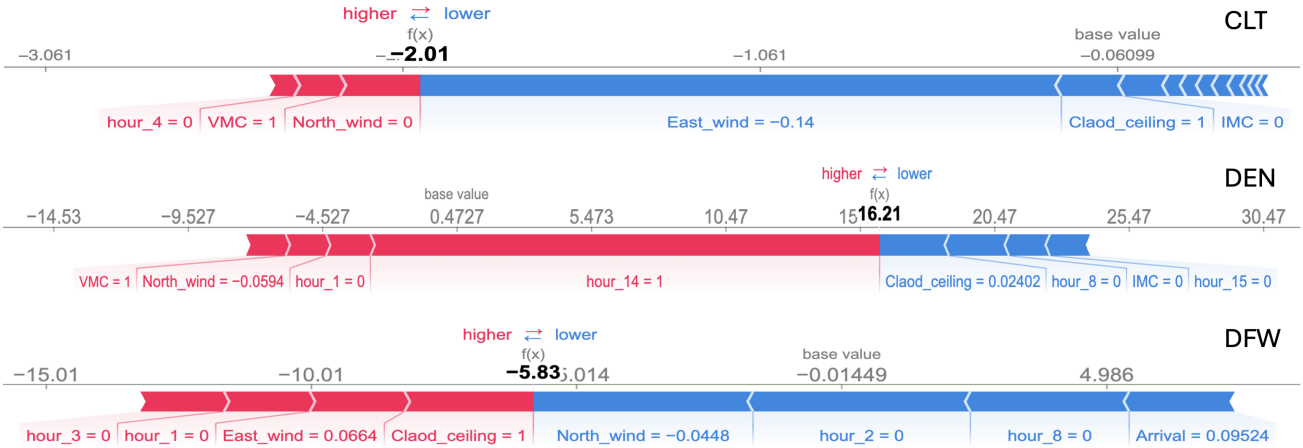


Fig. 4: The local instance explanation and feature effects on the decision of the RCA tool. For this instance, the runway configuration selections are North/North, South-East/South-East, and North/North-West (Arrival/Departure) for CLT, DEN, and DFW, respectfully.


[4] L. E. Celis, L. Huang, V. Keswani, and N. K. Vishnoi, "Classification with fairness constraints: A meta-algorithm with provable guarantees," in *Proceedings of the conference on fairness, accountability, and transparency*, 2019, pp. 319–328.

[5] B. H. Zhang, B. Lemoine, and M. Mitchell, "Mitigating unwanted biases with adversarial learning," in *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, 2018, pp. 335–340.

[6] G. Montavon, A. Binder, S. Lapuschkin, W. Samek, and K.-R. Müller, "Layer-wise relevance propagation: an overview," *Explainable AI: interpreting, explaining and visualizing deep learning*, pp. 193–209, 2019.

[7] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *Advances in neural information processing systems*, vol. 30, 2017.

[8] R. Schwartz, R. Schwartz, A. Vassilev, K. Greene, L. Perine, A. Burt, and P. Hall, *Towards a standard for identifying and managing bias in artificial intelligence*. US Department of Commerce, National Institute of Standards and Technology …, 2022, vol. 3.

[9] E. Ferrara, "Fairness and bias in artificial intelligence: A brief survey of sources, impacts, and mitigation strategies," *Sci*, vol. 6, no. 1, p. 3, 2023.

[10] L. Halawi, M. Miller, and S. Holley, "Fostering trust in artificial intelligence in commercial aviation: an exploratory study." *Issues in Information Systems*, vol. 25, no. 2, pp. 397–407, 2024.

[11] A. P. Saraf, K. Chan, M. Popish, J. Browder, and J. Schade, "Explainable artificial intelligence for aviation safety applications," in *AIAA Aviation 2020 Forum*, 2020, p. 2881.

[12] Y. Xie, N. Pongsakornsathien, A. Gardi, and R. Sabatini, "Explanation of machine-learning solutions in air-traffic management," *Aerospace*, vol. 8, no. 8, p. 224, 2021.

[13] L. H. Nazer, R. Zatarah, S. Waldrip, J. X. C. Ke, M. Moukheiber, A. K. Khanna, R. S. Hicklen, L. Moukheiber, D. Moukheiber, H. Ma *et al.*, "Bias in artificial intelligence algorithms and recommendations for mitigation," *PLOS digital health*, vol. 2, no. 6, p. e0000278, 2023.

[14] F. Kamiran and T. Calders, "Data preprocessing techniques for classification without discrimination," *Knowledge and information systems*, vol. 33, no. 1, pp. 1–33, 2012.

[15] T. Kamishima, S. Akaho, H. Asoh, and J. Sakuma, "Fairness-aware classifier with prejudice remover regularizer," in *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2012, Bristol, UK, September 24-28, 2012. Proceedings, Part II 23*. Springer, 2012, pp. 35–50.

[16] V. Iosifidis, B. Fetahu, and E. Ntoutsi, "Fae: A fairness-aware ensemble framework," in *2019 IEEE international conference on big data (big data)*. IEEE, 2019, pp. 1375–1380.

[17] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should i trust you?" explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 1135–1144.

[18] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: visual explanations from deep networks via gradient-based localization," *International journal of computer vision*, vol. 128, pp. 336–359, 2020.

[19] A. Kumar, A. Zhou, G. Tucker, and S. Levine, "Conservative q-learning for offline reinforcement learning," *Advances in neural information processing systems*, vol. 33, pp. 1179–1191, 2020.

[20] M. Memarzadeh, Z. Wang, F. Masrour Shalmani, P. Razzaghi, and K. M. Kalyanam, "Responsible ai for air traffic management: Application to runway configuration assistance tool," in *US-Europe Air Transportation Research Development Symposium*, 2025, pp. in–press.

[21] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[22] N. Ebensperger, P. Razzaghi, M. Memarzadeh, P. Wei, and K. M. Kalyanam, "Enhancing runway configuration assistant model: The role of explainable ai for model interpretability," in *AIAA AVIATION 2022 Forum*, 2025, pp. in–press.