Ground Stop Adjuster: A Machine Learning Approach to Improve Air Traffic Management Initiatives

Farzan Masrour Shalmani*[1], Milad Memarzadeh[2], Aida Sharif Rohani[2], and Krishna Kalyanam[2]
[1]*Universities Space Research Association (USRA), NASA Ames Research Center, Moffett Field, CA 94035, USA*
[2]*NASA Ames Research Center, Moffett Field, CA 94035, USA*

Traffic Management Initiatives (TMIs) play a crucial role in balancing demand and capacity within the U.S. National Airspace System (NAS). In current practice, traffic management coordinators (TMCs) determine and issue TMIs and recent research has explored the use of machine learning tools to aid the TMCs. However, most studies have primarily focused on a particular type of TMI, i.e., Ground Delay Programs (GDPs) due to their higher rate of occurrence and longer duration. This study investigates a machine learning approach for monitoring and adjusting a different type of TMI, i.e., Ground Stop (GS), aiming to assist human decision-makers with accurate, consistent, and timely recommendations. Using data from three major airports in the New York metroplex, we evaluated models that predict GS parameters, such as duration and scope. Our results demonstrate that using data from all airports in the NY metroplex and increasing feature granularity improve the prediction accuracy of the ML models.

I. Introduction

Collaborating closely with commercial air carriers and related organizations, the Federal Aviation Administration (FAA) regulates air traffic and ensures the safety and efficiency of air operations in the US. Air traffic controllers (ATCOs) make strategic decisions, such as delaying, rerouting, or canceling flights, partly based on guidance provided by the FAA's Air Traffic Control System Command Center (ATCSCC). The guidance includes, among other things, control measures known as Traffic Management Initiatives (TMIs) designed to mitigate adverse conditions (e.g., bad weather). Two types of terminal TMIs that are routinely used are Ground Delay Programs (GDPs) and Ground Stops (GSs). In a GDP, flights destined for an airport facing thunderstorm activity experience delays at their origin airports. This proactive approach minimizes the risk of routing aircraft through hazardous weather conditions and also replaces potential (fuel burning) airborne delays with ground delays. In a GS, a temporary restriction is imposed on the departure or arrival of aircraft at a specific airport or within a designated airspace. Although other TMIs (e.g., airspace flow program, miles-in-trail) are also implemented as part of (air) traffic flow management in the NAS, the focus of this work is on terminal TMIs and in particular, Ground Stops.

Since TMIs, by design, results in flight delays or cancellations, it is crucial to put in place the correct set of program parameters, such as scope and duration. For example, when a GS is extended beyond what is necessary, it imposes unnecessary restrictions on departing flights, leading to long delays and potential disruption to the broader traffic flow. This situation could occur due to an inaccurate prediction of the GS' duration based on rapidly changing weather conditions. On the other hand, if a GS ends prematurely before the capacity constraints at the destination airport or enroute sectors are resolved, it can result in congestion and airborne holding. The challenge lies in precisely aligning the termination of the GS with the resolution of capacity constraints, avoiding unnecessary ground delays while also preventing airborne holding due to premature termination. Incorrectly specifying the parameters of a GS can exacerbate these issues, increasing delays, causing higher traffic volumes, and placing additional workload on ATCOs. Note that this issue is complicated by the uncertainty in weather prediction and traffic volume (i.e., demand-capacity balancing).

To address this complex issue, we propose the integration of Machine Learning (ML) models in the Traffic Flow Management (TFM) pipeline. In current operations, decisions are made by human experts based on extensive training, historical patterns, available traffic and weather data. Historical data can indicate the likely impact of various TMIs. ML models properly trained on this historical data can offer valuable insights and aid human decision-making. With the FAA increasingly exploring advanced analytics, ML has emerged as a focal point for enhancing TFM within the NAS. As a first step, this study aims to provide TMCs with decision-making support for the issuance and adjustment of TMIs.

*Corresponding author: farzan.masrourshalmani@nasa.gov

Ground Stops are reactive measures, often issued for short durations of about an hour with potential extensions, and rely on real-time factors like severe weather, equipment failures, and traffic imbalances. While prior research has largely focused on GDPs, GS decisions involve unique challenges, such as their unpredictable nature, shorter durations, and more immediate operational impact. This study addresses these gaps by introducing a ML-based approach to predict and adjust GS parameters, such as duration and geographical scope. We fuse diverse data sources, including traffic, weather conditions, and historical GS data, to identify influential factors and improve decision-making accuracy. Unlike prior studies that often narrowly focus on GDPs or treat TMI prediction as a binary classification problem, our work emphasizes the GS-specific nuances and provides actionable insights for human operators. In our approach, the GS model is designed as a multi-class classification problem to predict the status and scope of GSs.

The remainder of this paper is structured as follows. In Section II, we provide a literature review on the application of ML to TMIs. Section III presents our proposed framework, detailing its design, components, and methodologies for predicting and managing Ground Stops. In Section IV , we describe the experimental setup, including data sources, evaluation metrics, and baseline comparisons, followed by a discussion of the results. Finally, Section V concludes the paper, summarizing key findings and outlining potential directions for future work.

II. Literature Review

The prediction and monitoring of TMIs using ML has gained significant attention due to its potential to improve air traffic flow. TMIs, such as GDPs and GSs, are critical for managing air traffic disruptions caused by weather or capacity constraints. Predicting TMI occurrence and adjusting parameters like duration and scope are essential tasks. Several ML frameworks have been developed to address these challenges, particularly focusing on GDPs due to their higher rate of occurrence in the NAS. Early research on TMI prediction primarily concentrated on GDPs. Relying on weather forecasts, several statistical methods were used to produce probabilistic capacity profiles which in conjunction with deterministic models provided insights into the GDP planning process [1–4]. The downside of using deterministic models is their reliance on fixed inputs and predetermined rules, which lack the ability to account for the inherent uncertainty and variability present in real-world scenarios. In a separate series of studies, researchers aimed to predict GDPs and GSs using various supervised learning methods, including Decision Trees, Naive Bayes, Support Vector Machines, Random Forests, and boosting ensembles. These studies analyzed the influence of weather conditions and arrival demand on TMI incidents [5–10].

A limited number of studies focused on predicting the parameters of TMIs, specifically addressing their duration and extent. In one such study that focused on optimizing the TMI parameters at San Francisco International Airport (SFO), the authors utilized a probabilistic forecast of fog [11]. They simulated various capacity scenarios based on the (fog) burn-off forecasts, and selected GDP parameters that minimized airborne and overall ground delays. However, this approach exclusively emphasizes stratus (fog) burn-off as the primary determinant of GDPs and GSs, neglecting other influential factors such as severe weather events, runway closures, demand/capacity imbalances, and other important variables. Jones et al. [12] introduced reinforcement learning-based methods to recommend GDP parameters like program rates and scope under uncertainty, which improved decision-making in real-time. Additionally, Buxi and Hansen [1] integrated weather forecasts into models to optimize GDP arrival rates, highlighting the role of accurate weather predictions in TMI parameter adjustment.

In addition to supervised methods, unsupervised approaches have been employed to find similar historical TMI events, aiding in decision-making for air traffic management. Kuhn et al. [13] introduced a methodology to identify comparable days in the context of TMIs by using a distance metric based on feature importance and clustering techniques. Similarly, Estes et al. [14] utilized the K-center clustering method to summarize large TMI datasets into a smaller set of representative days, offering interpretable insights for decision-makers. These methods allow for the discovery of patterns in historical data, which can be leveraged for better predictions and adjustments of TMI parameters in real-time.

III. The GS Adjuster Framework

The objective of our "GS Adjuster" framework is to deliver reliable, consistent and expedited recommendations for the progression, adjustment, and termination of GSs along with its various parameters. The two main components of the framework are data preparation/processing and GS adjuster (ML) model. Each of these components will be discussed in more detail in the following subsections.

A. Data Preparation and Processing

The data preparation and processing component is a critical part of the pipeline, ensuring that raw data is cleaned, organized, and transformed into a format suitable for use in the predictive model of the GS Adjuster. The input data consists of traffic data, TMI records, actual and forecast weather data, cancellations and holdings, as well as runway closures. This component consists of three key steps: feature engineering, dataset consolidation, and conversion to ML-ready formats. The first step in the data preparation pipeline is feature engineering, which is essential for ensuring the quality and consistency of the input data. This process involves parsing text data, handling missing values, and applying various data engineering techniques such as normalization and vectorization. To ensure uniformity, the data is converted into an hourly format, which facilitates better analysis and model training. This transformation is vital for ensuring that the predictive models can accurately identify the necessity and parameters of GSs.

Once feature engineering is completed, the next step is dataset consolidation. In this step, all the data sources—traffic data, weather data, and TMI records—are merged into a single dataset. This is done by using date, time, and airports as the primary keys for merging. This consolidation allows for a unified dataset that can be used to train the model. By merging these datasets, we ensure that all relevant variables are synchronized and can be accurately utilized in downstream predictive tasks.

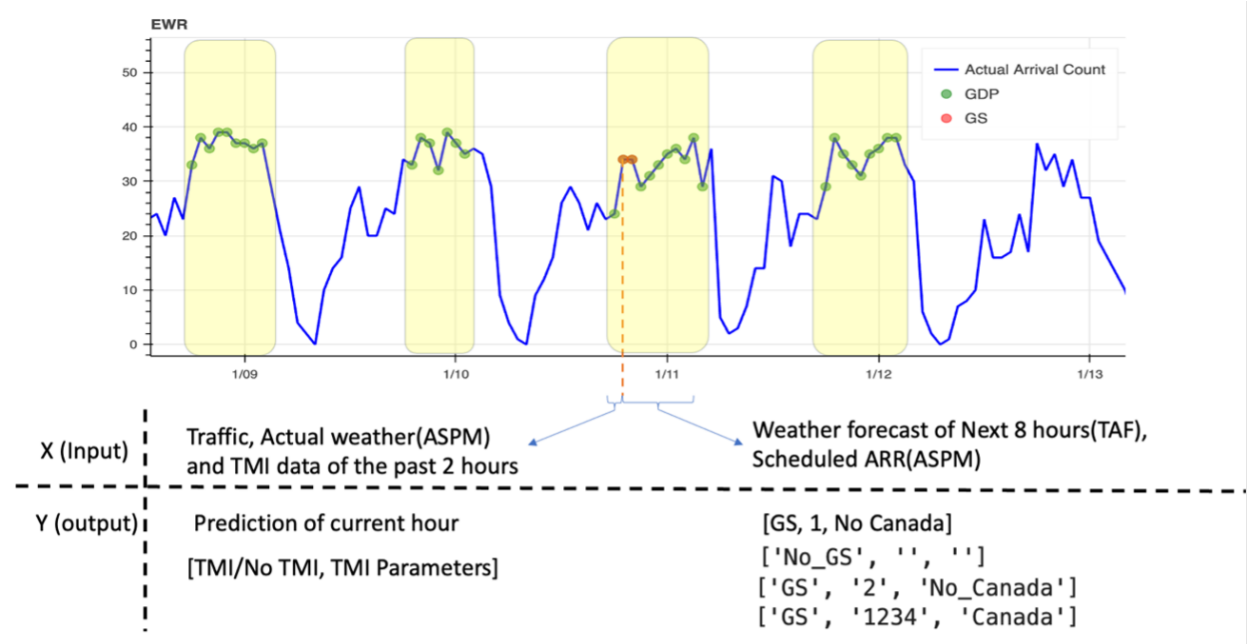

Fig. 1 Our study focuses on time intervals when a TMI program is in place. The yellow boxes indicate the time windows we consider in the training data, while data points outside of these time windows are ignored.

The final step is converting the consolidated data into a format suitable for the GS adjuster model. In this framework, each data instance summarizes ten hours of data. Specifically, the data loader for the GS model prepares the input and output as follows:

- Removing Extra Intervals: since GSs do not occur frequently, one challenge is creating a balanced training dataset. To address this, the training data is adjusted by removing "normal" time steps where no significant event occurred. Specifically, we retain time intervals where there was a TMI (either GS or GDP) in place, or where a TMI ended exactly in the preceding interval. This creates dynamic time windows around TMIs for each airport, improving the relevance and balance of the training dataset. Fig.1 illustrates an example of these time windows for Newark Liberty International (EWR) airport.
- Generating Multiclass Data for Each Airport: after removing extra intervals, the data loader generates the following for each airport:
  - X: An $n \times d$ matrix where n is the number of data points, and d is the number of features. Each row encapsulates ten hours of data surrounding a specific timestep. This includes actual traffic, weather, and TMI data from the two hours preceding the timestep, as well as weather forecasts and scheduled traffic for

the following eight hours. The row labeled "X" in the table at the bottom of Fig.1 provides an example of this structure.

- Y: The output for each timestep consists of three dimensions. The first dimension is a binary decision depicting whether a GS should be implemented for the following hour or terminated. The second dimension defines the scope of the GS within the NAS, while the third dimension determines whether the GS affects flights from Canada. One of the challenges with TMI modeling is the sparsity of TMI events, particularly regarding its scope. To address the challenge of scope in the GS model output, we implemented grouping. With 20 Air Route Traffic Control Centers (ARTCCs) in the NAS, we utilized historical data to group them into 4 categories. In particular, we summarized our historical data in a graph format where nodes represent centers, and link weights are defined based on the co-occurrence of centers in the scope parameter of TMIs. By identifying strongly connected components in this graph, we were able to partition the centers into four groups. Fig. 2 illustrates the 20 centers in the NAS, with the color coding representing the four groups we defined. Consequently, the output Y for the NAS scope dimension will indicate the groups impacted by the GS for the given timestep. For Canada, we further simplified the process by setting the model to simply decide whether flights from Canada should or should not be included in the GS. The row labeled "Y" in Fig.1 provides an example of these outputs.
- Time: A one-dimensional vector that records the timesteps used to create each row in the X and Y matrices.

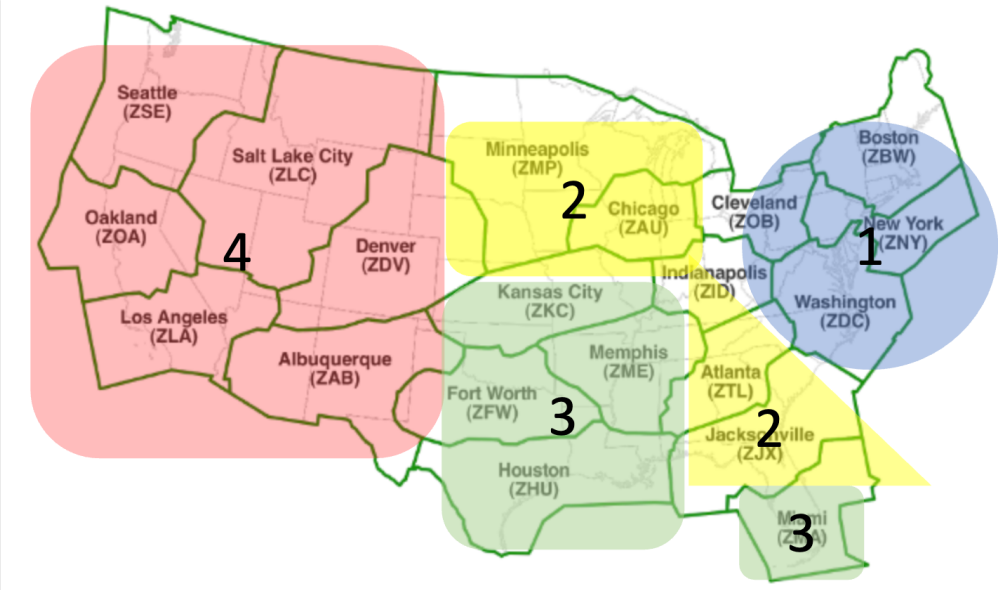

Fig. 2 The NAS centers, alongside the color-coded grouping. The GS scope is defined based on a list of centers that should be included.

B. GS Adjuster Model

As mentioned earlier, our model is designed as a multi-class classification model, focusing on predicting the status and scope of GSs on an hourly basis. GSs are typically unplanned and issued for an hour with the possibility of extension. By leveraging historical data from the past two hours and forecasted conditions for the upcoming hours, the model aligns with the operational realities of GS management, providing timely and accurate predictions to support data-driven decision-making.

Human decision-making for GSs follows a hierarchical process: first determining whether a GS is needed, and if so, defining its scope. Based on this, we explored two model structures for the GS prediction: a hierarchical structure [15] and an independent structure. The hierarchical structure organizes the problem into a class hierarchy—often a tree or a Directed Acyclic Graph (DAG)—and accounts for the dependency of earlier decisions on subsequent ones. In this approach, we use the local classifier per level method, which involves training a multi-class classifier for each level of the hierarchy. Instead of assigning each data instance to a single class directly, the hierarchical model arranges the classes

into a structured hierarchy, as illustrated in Fig.3. Alternatively, in the independent structure, dependencies between decision dimensions are not considered. Separate multi-class classifiers are trained independently for each dimension of the output. In both model structures, there is one classifier to determine the need for a GS and two additional classifiers to predict GS parameters.

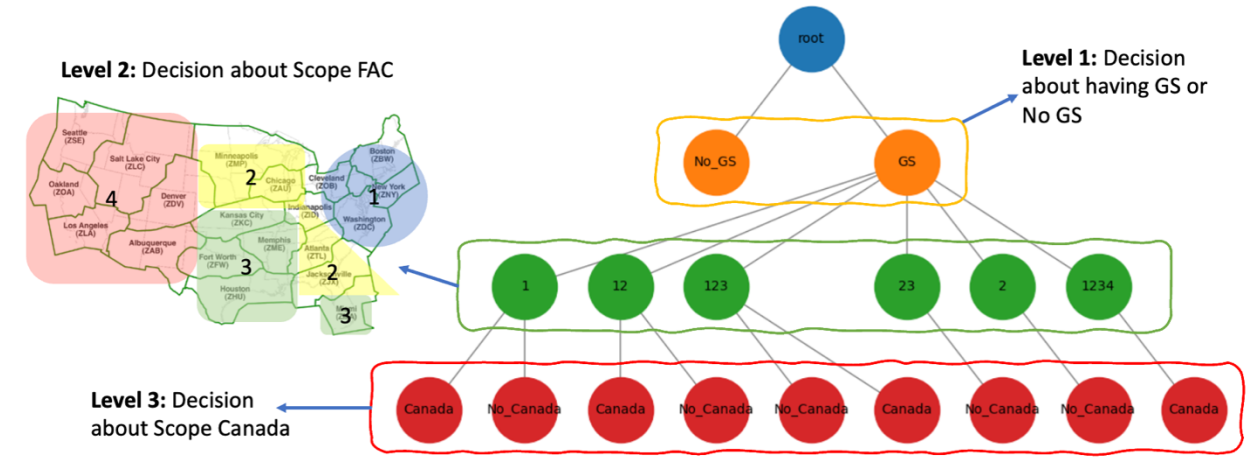

Fig. 3 Hierarchical model structure

IV. Experiments

In this section, we present the experimental setup used to evaluate the performance of the proposed model for GS. We describe the data used, the evaluation metrics, and the baseline models for comparison. Following this, we present the results of our experiments, highlighting the effectiveness of our approach in predicting GS parameters. Finally, we perform a feature importance analysis to understand the contribution of different input features to the model's predictions.

A. Experimental Setup

1. Datasets

We study all three major airports in the New York metroplex — LaGuardia (LGA), John F. Kennedy International (JFK), and Newark Liberty International (EWR). We fuse traffic, weather and other relevant aviation data from 2017 to 2019 to train and validate the ML models. The data sources we used include (1) Terminal Aerodrome Forecast (TAF), which provides meteorological forecasts specific to each airport and is issued four times a day for predefined time periods; (2) the TMI dataset, which includes all GSs and GDPs along with their respective parameters; (3) the Aviation System Performance Metrics (ASPM) dataset, which contains traffic-related data such as aircraft delays and arrival and departure rates; (4) Notices to Airmen (NOTAMs) that are used to extract runway closure data and manage inter-dependencies between nearby terminals; and (5) flight cancellation data and the number of airborne holdings caused by TMIs.

Since the data is sequential in nature, the data loader splits the data according to time to prevent any potential data leakage. We used 2017 and 2018 data for training, the first six months of 2019 for validation, and the last six months of 2019 for testing. Table 1 summarizes aggregate statistics about the training, validation, and testing data used for the GS models. The table documents the effect of limiting data to the time steps when there was actually a TMI in place or when a TMI had just terminated. This resulted in a more balanced distribution of the GS class (GS positive class) versus "No GS" (GS negative class), which might help the training process. While JFK and LGA follow very similar distributions, with 40% and 42% GS positive class respectively, EWR has proportionally fewer GS incidents at 28%.

Table 1 The summary of statistics for GS Data.

Airport	# Train	#Validation	#Test	GS Positive Class%	GS Negative Class%
JFK	2435	335	390	40%	60%
LGA	3635	1022	1100	42%	58%
EWR	5324	1349	1317	28%	72%

2. Evaluation Metrics

Conventional measures like precision, recall, F1 score, and accuracy are not entirely suitable for Hierarchical Classification (HC) due to the relationships among the classes. Given the complexity of our models and the hierarchical nature of the GS model, summarizing evaluation metrics across different layers of hierarchy poses unique challenges. Furthermore, it is crucial to understand how well classical metrics such as the F1 score, precision, and recall align with the ultimate objectives of the GS adjuster model's prediction. While a simple baseline such as repeating the last hour's data might perform relatively well on average accuracy, it fails to capture critical factors like the termination of GSs or changes in their scope.

To address these challenges, we developed the following specialized evaluation metrics:

- Terminated F1: this metric evaluates how well the model captures the ending intervals of a GS. It is crucial for understanding the model's ability to detect when a GS should cease, which is a significant aspect of operational efficiency.
- Started F1: this metric assesses the model's performance in identifying the starting intervals of a GS. Accurate detection of GS initiation is essential for timely and effective implementation of TMIs.
- Continued F1: this metric measures the model's performance during intervals with no changes, including:
  - Intervals with no GS in place.
  - GS intervals where parameters remain unchanged. This evaluation helps in understanding the model's consistency and stability over periods with no significant changes.
- Scope F1: this metric evaluates the model's effectiveness in capturing the parameters of a GS, including its scope and impact on different regions. This is vital for assessing the model's precision in specifying the geographical and operational scope of a GS.

By employing these tailored evaluation metrics, we aim to capture the nuanced performance of our models more accurately, ensuring alignment with the broader objectives of GS monitoring. These metrics help us understand not just how well our models perform on average, but also their effectiveness in critical operational scenarios, thereby enabling more informed and effective decision-making in managing the TMIs.

3. Baselines models

For both hierarchical and independent structures, any multi-class classifier could serve as the level-wise classifier. We utilized various state-of-the-art multi-class classifier models, including Random Forest, Decision Trees, K-Nearest Neighbors, MultiLayer Perceptron (MLP), and Logistic Regression, to forecast the duration and scope of the GSs. Additionally, we employed XGBoost for enhanced predictive performance. In the hierarchical structure, we used the HiClass Python library, an open-source tool specifically designed for hierarchical classification that is compatible with scikit-learn*. HiClass efficiently handles hierarchical dependencies between classes, enabling us to model the decision-making process involved in GS issuance and progression.

To ensure the best performance, we applied a thorough hyperparameter tuning process for all classifiers. Each model underwent a random search across its predefined hyperparameter space, sampling 50 configurations. The selection of the optimal model was based on maximizing the mean of the four F1 metrics we introduced above on the validation dataset. For consistency and reproducibility, we used a fixed random seed throughout all experiments.

B. Experimental Results

In this section, we present a series of experiments designed to address three key questions in predicting the status and scope of GSs. First, we evaluate which model structure—hierarchical or independent—yields better performance by testing both structures with several state-of-the-art multi-class classifier models. Second, we investigate the impact

*https://pypi.org/project/hiclass/

of training models at the Metroplex level versus the Airport level to determine the most effective scope for accurate predictions. Third, we assess how feature granularity and the number of input features influence model performance.

1. Model Architecture Comparison

In this section, we evaluate the performance of hierarchical and independent model structures for predicting the status and scope of GSs. We tested each architecture with several state-of-the-art multi-class classifier models on three major airports, reporting the top two models' results across four F1 evaluation metrics: Started, Continued, Terminated, and Scope. Table 2 provides a detailed overview, with the last column showing the average performance across all four metrics. To ensure a fair comparison, input features were kept consistent across all models.

Table 2 Performance comparison of hierarchical and independent model structures on predicting GS status and scope metrics.

Airport	Architecture	Classifier	Started F1	Continued F1	Terminated F1	Scope F1	Average
EWR	Hierarchical	Decision Tree	0.5035	0.8629	0.6933	0.3448	**0.6011**
	Hierarchical	Random Forest	0.5035	0.9231	0.6308	0.3055	0.5907
	Independent	Decision Tree	0.5139	0.8490	0.7152	0.3018	0.5950
	Independent	XGBoost	0.4930	0.8644	0.7075	0.2932	0.5895
JFK	Hierarchical	Decision Tree	0.7397	0.8730	0.7742	0.4384	**0.7063**
	Hierarchical	MLP	0.5846	0.8160	0.6984	0.3374	0.6091
	Independent	XGBoost	0.6269	0.8618	0.8125	0.3523	0.6634
	Independent	Decision Tree	0.6269	0.9134	0.6667	0.3578	0.6412
LGA	Hierarchical	Random Forest	0.6526	0.8439	0.7399	0.3200	**0.6391**
	Hierarchical	MLP	0.6667	0.8317	0.7399	0.3020	0.6351
	Independent	XGBoost	0.6597	0.8239	0.7444	0.3259	0.6385
	Independent	Random Forest	0.6455	0.8372	0.7059	0.3184	0.6268

The hierarchical architecture, on average, achieved the best performance across all airports. For both EWR and JFK, the best-performing models used a Decision Tree as the level-wise classifier, while for LGA, Random Forest yielded the best results. The performance gap between hierarchical and independent architectures was most pronounced at JFK, while at EWR and LGA, the performance difference was minimal when using the best level-wise classifier for each architecture. Tree-based models, including Decision Tree, Random Forest, and XGBoost, consistently outperformed other classifiers across multiple metrics within both hierarchical and independent architectures. The only non-tree-based model to appear among the top two was MLP, which performed well within the hierarchical structure at both JFK and LGA airports.

Predicting the scope of a GS was notably more challenging than predicting its status. The Scope F1 evaluation metric, which averages the F1 scores for predicting which ARTCCs (± Canada) to include in the scope, was consistently lower than the F1 evaluation metrics for status prediction (Started, Continued, and Terminated metrics). This discrepancy likely results from the different levels of task complexity: while GS status prediction is a binary classification (GS or No-GS), scope prediction is a multi-label classification task with over ten possible labels. The best hierarchical model outperformed the independent architecture in Scope prediction at JFK and EWR, with results closely matched at LGA. For the status metrics (Started, Continued, and Terminated), no single model consistently excelled across all three metrics. Trade-offs in performance were observed, with models optimized for predicting the Continued metric often showing lower accuracy in predicting the Terminated metric across all three airports. This trade-off suggests that while hierarchical models may have an advantage for complex multi-label tasks like Scope prediction, there may be compromises in performance across different aspects of GS status prediction, highlighting the need for careful model selection based on specific operational goals.

Overall, this analysis emphasizes the strengths of hierarchical architectures, particularly for complex tasks, and underscores the consistent effectiveness of tree-based models across multiple metrics, offering useful guidance for selecting suitable model architectures for TMI prediction.

2. Airport-level vs. Metroplex-level Training

In this section, we examine the impact of training models at the Metroplex-level, where a single model is trained on data from all three major airports (JFK, EWR, and LGA), compared to Airport-level training, where models are trained individually for each airport. This analysis aims to determine the most effective training scope for accurate GS prediction. We compare the performance of the top three hierarchical and independent models identified in the previous section, assessing their F1 evaluation metrics when trained on each airport separately versus at the Metroplex level. Table 3 presents these results, showing the average between the 4 defined F1 metrics (started, continued, terminated, and scope) for each model at the Metroplex along with the percentage of improvement over models trained individually on each airport shown in parentheses.

Table 3 Comparison of Metroplex-level and Airport-level training for top hierarchical models in predicting GS.

Architecture	Classifier	EWR	JFK	LGA
Hierarchical	Decision Tree	**0.6226 (+3.6%)**	**0.6754 (-4.4 %)**	**0.6464 (+3.3%)**
	Random Forest	0.6103 (+3.3%)	0.6307 (+7.7%)	0.6182 (-3.3%)
	MLP	0.5875 (+4.1%)	0.6199 (+1.8%)	0.6168(-2.9%)
Independent	XGBoost	0.5916 (+0.4%)	0.6496 (-2.1%)	0.6329 (-0.9%)
	Decision Tree	0.5894 (-0.9%)	0.6339 (-1.1%)	0.6413 (+4.4%)
	Random Forest	0.613 (+4.1%)	0.6313 (-0.8%)	0.623(-0.6%)

The results reveal that the hierarchical architecture consistently benefits from Metroplex level training, with six out of nine cases showing improved average F1 evaluation metric compared to airport-specific models. This trend suggests that pooling data across airports enhances the hierarchical models' ability to generalize, likely due to the increased variety in training data. Hierarchical Decision Tree remains the best-performing model overall, reinforcing the advantage of hierarchical architectures for GS prediction. The findings suggest that the hierarchical models benefit from the additional training data available at the Metroplex level.

3. Effect of Increasing Feature Granularity

To assess the impact of increasing feature granularity on model performance, we conducted experiments across various feature sets for the top-performing models: Hierarchical Decision Tree (Hierarchical-DT), Independent XGBoost (Independent-XGB), Hierarchical Random Forest (Hierarchical-RF), and Independent Decision Tree (Independent-DT). As shown in Fig.4, the average F1 evaluation metrics for each model improve with an increase in the number of input features, though the rate of improvement varies across models.

The term "granular" is used here to describe both the higher resolution in time dimensions and the inclusion of additional, more detailed features. Feature granularity in this context refers to the level of detail represented in the input features. For example, in the scope of GS, feature granularity can vary based on how NAS centers are represented—either as four grouped categories or using one-hot encoding for all 20 centers. Similarly, for Canadian centers, higher granularity involves more detailed encoding. Another dimension of granularity arises from the temporal resolution of input data: we might use only the scheduled traffic for the current hour or include scheduled traffic for the next eight hours. Additionally, the number of features was increased by incorporating more detailed information. For instance, when considering meteorological conditions, we could limit the input to highly correlated conditions like snow and thunderstorms, or expand it to include a comprehensive set of weather variables from both forecasted and actual data.

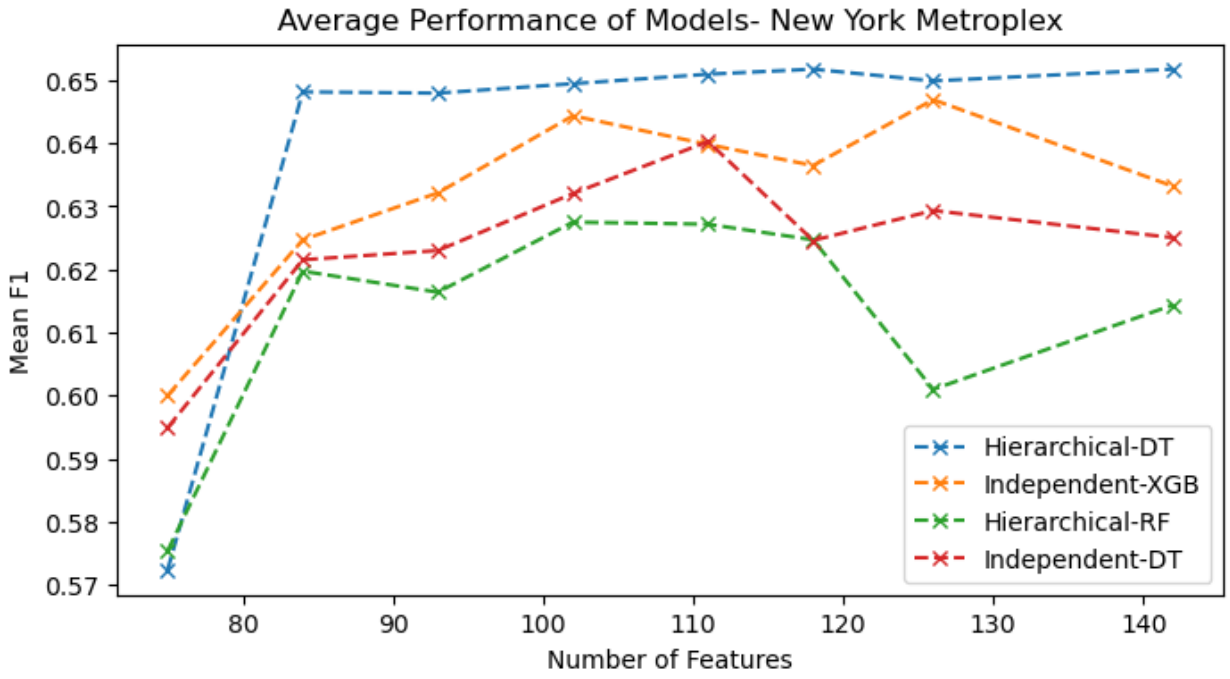

Fig. 4 Impact of feature granularity on average F1 evaluation metric for top-performing models.

Initially, at around 80 features, the models show average performance between 57% and 60%, with Hierarchical-DT slightly outperforming the others. With an increase in features to approximately 90, there is a notable improvement in performance across all models, reaching average perforamnce around 62 – 65%. This suggests that adding more features enhances the model's ability to capture relevant patterns in the data, particularly for Hierarchical-DT and Independent-XGB, which show the most significant performance gains. However, as the number of features continues to increase beyond 110, the performance improvements become less consistent. Independent-XGB and Independent-DT show some fluctuations, with performance peaking around 110–120 features and then leveling off or slightly declining. In contrast, Hierarchical-DT maintains relatively high performance throughout, reaching its highest average performance with the most granular feature set.

These results indicate that while adding more granular features generally improves model performance, there is a saturation point beyond which additional features may not yield significant benefits. This pattern is particularly evident in Independent-XGB and Independent-DT, which may be more sensitive to overfitting with increased feature dimensionality. Hierarchical models, especially Hierarchical-DT, appear more robust to this effect, suggesting they can better leverage additional feature granularity without sacrificing performance.

C. Feature Importance

To better understand the factors influencing the predictions of our multi-class classification models for GS, we utilized permutation feature importance analysis. This method provides insight into the relative contribution of each feature by measuring the decrease in model performance when a feature's values are randomly shuffled, disrupting its relationship with the target variable. Permutation feature importance is particularly valuable for interpreting complex models, as it reveals how dependent the model is on each feature for making accurate predictions. If shuffling a feature significantly reduces the model's performance, that feature is deemed to be more influential.

For this analysis, we applied permutation feature importance to two of the best-performing models identified in the previous section: the Hierarchical Decision Tree (Hierarchical-DT) and the Independent XGBoost (Independent-XGB) models. Both the hierarchical and independent architectures involve three separate level-wise classifiers: level 1 determines the GS status, level 2 predicts the scope within the USA, and level 3 focuses on the scope within Canada. We assessed feature importance at each of these levels individually, applying permutation on each feature ten times using the test set of each airport, then calculating and aggregating the level-wise F1 scores for each airport. Fig.5 shows the top five most important features for each model across all levels. This analysis helps identify the features that

significantly influence model predictions across different GS decision stages, offering insights for refining model inputs and improving accuracy.

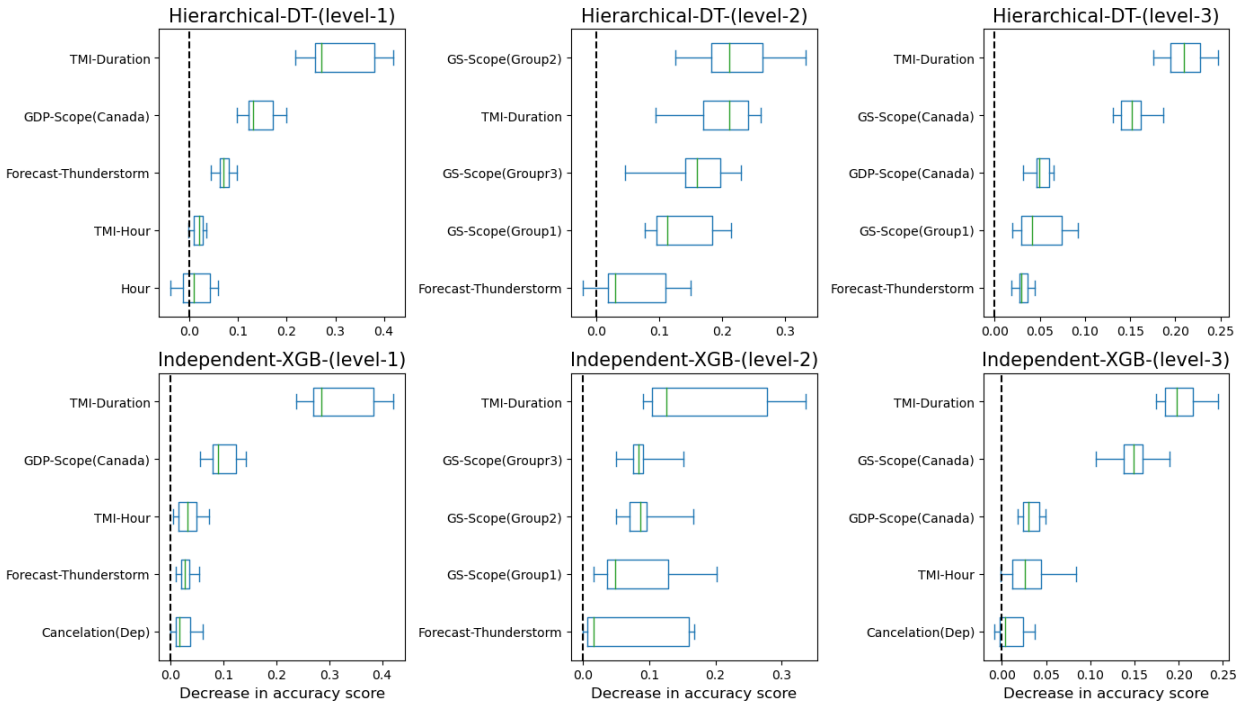

Fig. 5 Top five most important features for each model and decision level, determined using permutation feature importance. The x-axis represents the decrease in accuracy score when each feature is permuted.

The feature importance analysis reveals key insights into the factors that most influence the predictions of our multi-class classification models for GS. Across both the Hierarchical-DT and Independent-XGB models, certain TMI-related features consistently stand out. In level 1, where the models decide on the overall GS status, the most influential features include TMI-related parameters in the past two hours—particularly the initial planned GS duration and the current stage of the GS. Additionally, the presence of a GDP and its scope within Canada, as well as forecasted thunderstorms, play a significant role. While both models prioritize these features, they diverge slightly in their reliance on additional data: Independent-XGB also factors in departure cancellation information, while the Hierarchical-DT considers the hour of the day as an important indicator.

At level 2, where the models classify the scope within the U.S., both algorithms display similar behavior, primarily relying on recent GS scope within the U.S. as a crucial predictor. This consistency suggests that both hierarchical and independent architectures identify similar patterns when focusing on U.S. scope predictions. Finally, in level 3, which pertains to Canadian scope predictions, the main factors influencing model performance include the TMI duration, the GS and GDP scope within Canada, and again, TMI duration over recent hours. These results underscore the importance of both TMI-specific data and recent activity in informing accurate scope predictions, especially at the more granular Canadian level. Overall, the findings suggest that a combination of TMI parameters, weather information, and contextual features such as cancellations and time of day collectively drive the models' decision-making processes across different levels.

V. Conclusions

This study demonstrates that utilizing a hierarchical classifier is an effective architecture for predicting GS status and scope within the context of TMIs. The tree-based models such as Decision Tree, Random Forest and XGBoost emerged as consistently good performers across multiple evaluation metrics, showcasing robustness in handling complex GS prediction tasks. Training models at the (multi-airport) metroplex-level not only enhances performance but also mitigates the risk of overfitting, particularly important given the atypical nature of TMIs and the limited number of incident

data available. Furthermore, increasing the granularity of input data contributes positively to model performance, although to a lesser extent. Our feature importance analysis reveals that TMI parameters related to the immediate past, particularly those from the past two hours, along with weather forecasts—such as wind conditions and the occurrence of thunderstorms—are critical determinants for accurate predictions. Looking ahead, the next logical step in our work involves the development of a GDP component and its integration into our framework, which will further enhance the decision-making aid for TMCs.

Acknowledgments

The material is partly based upon work supported by the National Aeronautics and Space Administration under Contract Number NNA16BD14C, managed by the Universities Space Research Association (USRA).

References

[1] Buxi, G., and Hansen, M., "Generating day-of-operation probabilistic capacity scenarios from weather forecasts," *Transportation Research Part C: Emerging Technologies*, Vol. 33, 2013, pp. 153–166.

[2] Provan, C. A., Cook, L., and Cunningham, J., "A probabilistic airport capacity model for improved ground delay program planning," *2011 IEEE/AIAA 30th Digital Avionics Systems Conference*, IEEE, 2011, pp. 2B6–1.

[3] Kicinger, R., Krozel, J., Steiner, M., and Pinto, J., "Airport capacity prediction integrating ensemble weather forecasts," *Infotech@ Aerospace 2012*, 2012, p. 2493.

[4] Kicinger, R., Cross, C., Myers, T., Krozel, J., Mauro, C., and Kierstead, D., "Probabilistic airport capacity prediction incorporating the impact of terminal weather," *AIAA Guidance, Navigation, and Control Conference*, 2011, p. 6691.

[5] Liu, Y., Liu, Y., Hansen, M., Pozdnukhov, A., and Zhang, D., "Using machine learning to analyze air traffic management actions: Ground delay program case study," *Transportation Research Part E: Logistics and Transportation Review*, Vol. 131, 2019, pp. 80–95.

[6] Liu, Y., Hansen, M., Zhang, D., Liu, Y., and Pozdnukhov, A., "Modeling Ground Delay Program Incidence using Convective and Local Weather Information," *Proceedings of the Twelfth USA/Europe Air Traffic Management Research and Development Seminar (ATM2017), Seattle, WA, USA*, 2020, pp. 1–8.

[7] Mukherjee, A., Grabbe, S. R., and Sridhar, B., "Predicting ground delay program at an airport based on meteorological conditions," *14th AIAA aviation technology, integration, and operations conference*, 2014, p. 2713.

[8] Mangortey, E., Pinon-Fischer, O. J., Puranik, T. G., and Mavris, D. N., "Predicting The Occurrence of Weather And Volume Related Ground Delay Programs," *AIAA Aviation 2019 Forum*, 2019, p. 3188.

[9] Mangortey, E., Bleu-Laine, M.-H., Puranik, T. G., Pinon Fischer, O., and Mavris, D. N., "Machine learning approach to the analysis of traffic management initiatives," *Journal of Air Transportation*, Vol. 29, No. 2, 2021, pp. 56–68.

[10] Mangortey, E., Puranik, T. G., Pinon-Fischer, O. J., and Mavris, D. N., "Prediction and Analysis of Ground Stops with Machine Learning," *AIAA Scitech 2020 Forum*, 2020, p. 1684.

[11] Cook, L. S., and Wood, B., "A model for determining ground delay program parameters using a probabilistic forecast of stratus clearing," *Air traffic control quarterly*, Vol. 18, No. 1, 2010, pp. 85–108.

[12] Jones, J. C., Ellenbogen, Z., and Glina, Y., "Recommending Strategic Air Traffic Management Initiatives in Convective Weather," *Journal of Air Transportation*, Vol. 31, No. 2, 2023, pp. 45–56.

[13] Kuhn, K. D., "A methodology for identifying similar days in air traffic flow management initiative planning," *Transportation Research Part C: Emerging Technologies*, Vol. 69, 2016, pp. 1–15.

[14] Estes, A., Lovell, D. J., and Ball, M. O., "Unsupervised prototype reduction for data exploration and an application to air traffic management initiatives," *EURO Journal on Transportation and Logistics*, Vol. 8, No. 5, 2019, pp. 467–510.

[15] Silla, C. N., and Freitas, A. A., "A survey of hierarchical classification across different application domains," *Data mining and knowledge discovery*, Vol. 22, 2011, pp. 31–72.