

Analysis and Prediction Of The Interactions Between Ground Delay Programs and Ground Stops

AE 8900 MAV Special Problems

Presented by: Marc-Henri Bleu-Laine

Advisor: Prof. Dimitri Mavris

December 6th, 2019

Grade received: _____

Advisor's signature: _____

School of Aerospace Engineering

Georgia Institute of Technology

Honor Code Statement

I certify that I have abided by the honor code of the Georgia Institute of Technology and followed the collaboration guidelines as specified in the project description for this assignment.

Signed: _____

Analysis and Prediction Of The Interactions Between Ground Delay Programs and Ground Stops

Executive Summary

Traffic Management Initiatives (TMI) are implemented by Traffic Flow Management (TFM) personnel when a congestion at an airport occurs. Congested airports face an imbalance between the traffic demand and their capacity. This imbalance usually finds its causes to be the weather, traffic volume, runway incidents, and others. Even though TMIs are used as a mean to manage traffic and regulate excess demand or lower acceptance rate at an airport, they have a downside. Delays in the National Airspace System (NAS) impose stress on air traffic controller, passenger, and the economy. Overall delays also have a high price tag, as they resulted in a total cost of \$28.6 Billions in 2018 alone. During any given day, any TMI can be implemented, and this leaves room for TMIs to interact with each other. TMI interactions are common, and unfortunately can result in prolonged delays. The interactions also reduce the ability of predicting individual TMIs and have an effect on their parameters, such as the duration of the TMI. Current TMI tools do not take interactions into consideration due to the complexity of programming them out within the tools. Therefore focusing on the TMI interactions is as important as investigating individual TMIs. Previous work only focused on reducing delays and their duration by predicting TMIs such as Ground Delay Programs (GDP), and Ground Stops (GS). Ground Delay Programs are terminal TMIs that manage the air traffic flow by providing an Expected Departure Clearance Time (EDCT) to the impacted flights, which are flights destined for the congested airport. Ground Stops are more restrictive than Ground Delay Programs as impacted aircraft are forced to remain grounded until the end of the Ground Stop or until better conditions appear. Occasionally, Ground Stops are issued during an ongoing Ground Delay Program, and vice versa. Previous work did not include the interactions of these two TMIs and the literature does not provide any analyses of the models created for the prediction of Ground Stops or Ground Delay Programs. The model performances were evaluated but the reasons behind the model choices were not given. The focus of this work aims at 1) predicting the coincidences of GDP and GS and the type of coincidence (whether there is GS was preceding a GDP, or vice versa, when coincidence occurs), and 2) analyze the created prediction models.

To achieve these objectives, an inherited methodology from literature was modified and used. The first step of the methodology is to identify and acquire data. The dataset for this research came from two sources, The Traffic Flow Management System (TFMS), and the Automated Surface Observing System (ASOS). The TFMS dataset provides initial flight plan messages, amended flight plan messages, departure and arrival time notifications, flight cancellation messages, boundary crossing messages, track position reports (TFMS Flight), and also traffic flow management initiatives such as Ground Stops, Reroutes, Airspace Flow Programs, etc (TFMS Flow). Data from January 2017 to December 2017 was used. The second step of the methodology is to determine an airport of interest. LGA airport was therefore selected as it had more instances of coincidences than other airports. The third and fourth steps involved parsing the data of each database, fusing it using the month, day, and time, and cleaning it by removing rows with missing values. The fifth step involved the development of the models. More specifically, a neural network, a random forest model and a boosting ensemble model were selected as base models. 80% of the fused and cleaned dataset was used for training and validation. Using this subset, a 3-fold cross-validation was used to train and evaluate each of the base models with different hyperparameter settings, resulting in the training and validation of a total of 34 models. Prior to training, the Synthetic Minority Over-sampling Technique (SMOTE) algorithm was used to deal with a highly imbalanced training set. The fifth step was concluded by the selection of the best model for each of the two prediction cases. The last step of the methodology involved evaluating the best models on previously unforeseen data representing the other 20% of the original dataset. The Random forest model was found to be the best model for the prediction of the coincidence, and the the prediction of the GS preceding GDP, and vice versa, when coincidence occurs. The model achieved a Kappa Statistic of 0.962, and 0.856 for the coincidence of GS and GDP, and the GS preceding GDP, and vice versa, when coincidence occurs case, respectively.

The model analyses were then used to identify key predictors and their impact on both the coincidence of Ground Stops and Ground Delay Programs and the precedence of Ground Delay Program before Ground Stop, or vice versa,

when coincidence occurs. The analyses were also helpful to understand the reasons behind the model choices. The results revealed that the top predictors for predicting the coincidence were thunderstorms, low ceilings and pressure altimeter. It showed that the probability of coincidence increased by 9% and 4.5% whenever thunderstorms and low ceilings were present, respectively. The top predictors for predicting which Traffic Management Initiative will precede the other were thunderstorms, hour of day (midnight) and low ceilings. In particular, the likelihood of a Ground Stop preceding a Ground Delay Program when coincidence occurs is much higher with thunderstorms and at midnight, compared to a Ground Delay Program preceding a Ground Stop. Furthermore, the analyses also uncovered, through the usage of surrogate trees, the logical decisions that the models took to make the predictions.

It is anticipated that the methodology and results could provide a better understanding of coincidence to stakeholders, especially if more data (non weather-related) was to be included, and the work expanded to more airports. It would improve current TMI tools prediction capabilities and improve planning and implementation of Traffic Management Initiatives.

Analysis and Prediction Of The Interactions Between Ground Delay Programs and Ground Stops

Marc-Henri Bleu-Laine*
mhbl3@gatech.edu

Traffic Management Initiatives, Machine Learning, Decision Making

Traffic management initiatives (TMI) such as Ground Delay Programs (GDP) and Ground Stops (GS) are commonly used by traffic managers when congestion at an airport causes an imbalance between the demand and the capacity for that airport. The implementation of these TMIs causes delays and the delays create stress on traffic management personnel, airlines, passengers, and on the economy. Furthermore, when the TMIs interact with each other, their negative impacts can be enhanced. This research focuses on the coincidences of Ground Delay Programs and Ground stops and proposes a methodology that could help reduce their number and impact by using machine learning techniques to predict the coincidence of Ground Stops and Ground Delay Programs intersections at a given hour, predict which Traffic Management Initiative would precede the other during their coincidence, and analyze the created models. The Results identified the Random Forest models as the best performing one for both the coincidence of Ground Stops and Ground Delay Program, and the precedence of Ground Stop before Ground Delay Program, or vice versa, during their coincidences. Important predictors were found to be the presence of a thunderstorm, the presence of a low cloud ceiling, the altimeter pressure and the time (midnight). Lastly, built decision trees used as surrogate of the random forest models revealed the reasoning behind the algorithm's classification choices.

I. Motivation

A^{IR} Traffic Controllers (ATC) continually monitor demand and capacity at airports [1, 2]. Inclement weather, Runway-related incidents, equipment failures, and volume constraints often cause air traffic demand to exceed airport capacity. Whenever this occurs, traffic management personnel implement Traffic Management Initiatives (TMI) to balance demand and airport capacity [3, 4]. However, their implementation often leads to delays which sometimes propagate throughout the National Airspace System and are costly to airlines and passengers, as seen in Table 1. Consequently, efforts are being pursued by stakeholders in the aviation industry to improve the planning and implementation of Traffic Management Initiatives as a means to reduce delays, and their impacts. However, as with any other process, the planning and implementation of Traffic Management Initiatives continually faces challenges that need to be addressed. One of these challenges is the coincidence of two Traffic Management Initiatives (TMI): Ground Delay Programs and Ground Stops. The coincidence of the two TMIs often occurs due to rapid changes in conditions which leaves traffic management personnel with limited time to plan and implement initiatives.

Table 1 Total Cost of Delay in the United States (\$Billions) [5]

	2012	2013	2014	2015	2016	2017	2018
Airlines	5.7	6.0	5.8	5.8	5.6	6.4	6.4
Passengers	9.7	11.0	10.5	13.3	13.3	14.8	16.1
Lost Demand	1.3	1.4	1.4	1.8	1.8	2.0	2.1
Indirect	2.5	2.7	2.6	3.1	3.0	3.4	3.6
Total	19.2	21.1	20.3	24.0	23.7	26.6	28.6

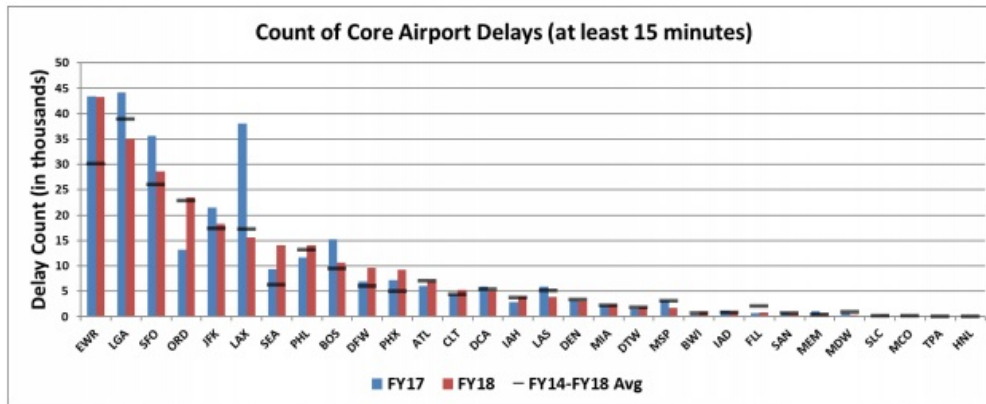


Fig. 1 Delays at The 30 Major U.S. Airports[5]

II. Background

Traffic Management Initiatives (TMI) are programs and tools that Traffic Flow Management (TFM) Personnel use to manage traffic. There are two types of Traffic Management Initiatives[6]:

- **Enroute** TMIs: They are used to manage traffic issue in enroute segment of a flight
- **Terminal** TMIs: they are used to regulate excess or lower acceptance rate an airport

Within each of these two high level categories there exist multiple types of TMIs. However, this focuses on only two types of terminal TMIs, which are Ground Delay Programs (GDP), and Ground Stops (GS). The subsequent subsections provide the definitions of the two TMIs along with scenarios in which they would be implemented by TFM personnel.

A. Ground Delay Programs (GDP)

Ground Delay Programs (GDP) is a traffic management procedure that is implemented to manage demand and capacity at an airport. Flights affected by GDP are delayed a GDP at their departure airport and given an Expected Departure Clearance Time (EDCT). It is the runway release time ("wheel up") assigned to an aircraft. The EDCT regulates the arrival of the flights at the impacted airport [2]. This way arrival demands are kept to a manageable level. Since the National Airspace System (NAS) is continuously monitored, the EDCT are updated when the congestion gets better or worse. GDPs are usually implemented due to weather or other causes that reduce the airport capacity. Fig 2 shows a scenario in which GDP can be implemented [2].

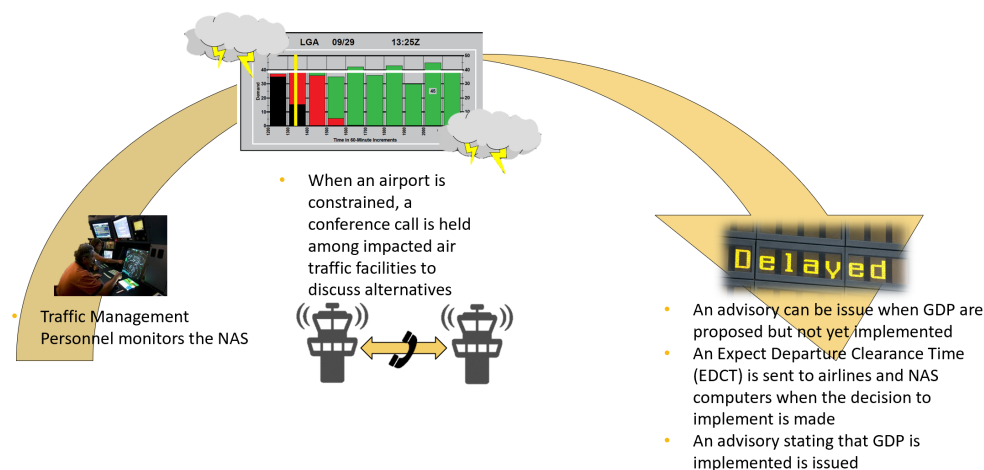


Fig. 2 Ground Delay Program Scenario [2]

In this scenario, Traffic Management Personnel continuously monitors the NAS. When a concern is raised about the

traffic situation at an airport, alternatives and potential scenarios are discussed between the Air Traffic Control System Command Center (ATCSCC) and the system users at the impacted airport. When a GDP is proposed an advisory is issued, even if it has not been implemented yet. The ECDT is sent to airlines and the NAS computers when the decision to implement the GDP is made, at this time another advisory is issued.

B. Ground Stops (GS)

Ground Stops are implemented whenever an airport is constrained over a short period of time, which can be caused by inclement weather, volume constraints, runway-related incidents, equipment failures, etc [?]. Unlike during the implementation of Ground Delay Programs, aircraft are not allowed to land at constrained airports during Ground Stops. Thus, en-route flights are kept in airborne holding patterns or are diverted, while flights that are yet to depart are grounded until the Ground Stop is terminated. This significantly impacts airports and flight operations, sometimes across the entire National Airspace System (NAS). Figure 3 provides an overview of the steps taken to plan a Ground Stop at a constrained airport. In particular, it shows that traffic management personnel provide stakeholders with the duration and the probability of extending a Ground Stop. It also shows that at the end of its duration, a decision is made to either terminate the Ground Stop, implement another Ground Stop, or implement a Ground Delay Program.

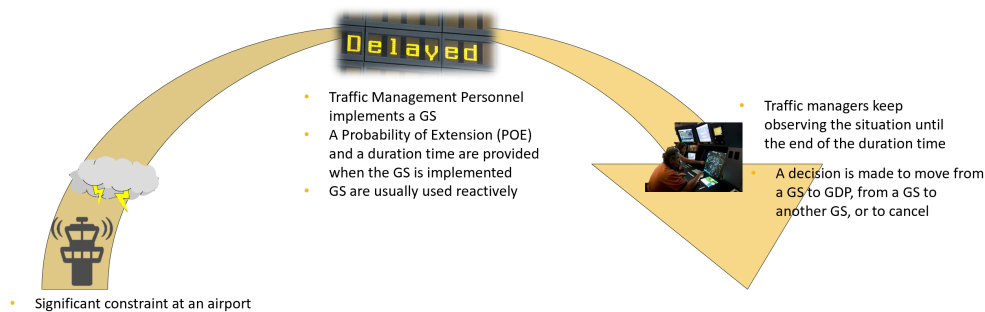


Fig. 3 Ground Stop Scenario [2]

C. Ground Delay Programs and Ground Stops Interactions

Interactions between GDP and GS can happen in different ways. For instance, as a flight is impacted by a GDP, weather degradation and others can cause GS to be issued. The GS would have higher priority over the GDP. Once the GS is released, if the GDP was still active, it would get a new EDCT, prolonging the delay. In another case, a GS could be implemented to allow Traffic Management to assess the situation of the National Airspace System. If the situation worsens, these conditions might create a demand and capacity imbalance, resulting in a GDP right after the GS.

D. Machine Learning

Machine learning can be defined as a set of methods that automatically detect patterns in data [7]. Machine learning can be divided into three subsets:

1) Supervised Learning

- Use training data to learn a mapping from input x to a target y
- if y is a categorical or nominal variable, it is called a classification problem
- if y is a real value, it is a regression problem

2) Unsupervised Learning:

- Knowledge discovery task
- No target specification
- The problem is not well defined since we are not told what patterns to look for

3) Reinforcement Learning:

- Less commonly used
- Useful to learn how to act when given occasional reward or punishment signals

This research will tackle binary and multi-class classification problems.

III. Literature Review

Jixin proposed the development of a framework to optimize key parameters of Ground Delay Programs such as file time, end time, and distance, using a genetic algorithm [8]. The model calculated the optimal Ground Delay Program file time, which was estimated to significantly reduce the delay times. Results showed that, in comparison with actual Ground Delay Programs that occurred, the proposed framework reduced the total delay time, unnecessary ground delay, and unnecessary ground delay flights by 14.7%, 50.8%, and 48.3%, respectively.

TMI interactions affects the parameters of GDP [9], which neither the literature nor Jixin's model captures. Wang generated a classification model using Ensemble Bagging Decision Trees to map historical airport weather forecast, schedule traffic, and other airport conditions to implement Ground Stop and Ground Delay Program operations. The model yielded an 85% overall classification accuracy when predicting Ground Stop only days and a 71% accuracy when predicting Ground Delay Program only days [9].

Mangortey et al. [4] uses machine learning to predict the occurrence of weather and volume related Ground Delay programs. Three datasets are fused together, the Traffic Flow Management System (TFMS), the Aviation System Performance Metrics (ASPM), and the Automated Surface Observing Systems (ASOS). A boosting ensemble model was the best at predicting the Ground Delay Programs, reaching a kappa statistic of 0.68.

Smith used weather data provided by the National Climatic Data Center (NCDC). Aircraft Arrival Rates data collected from the Aviation System Performance metrics (ASPM), and delay data obtained from the Bureau of Transportation Statistics website to build a predictive tool for air traffic delays predictions [10]. Smith used Support Vector Machines (SVM) to predict future airport capacity, and use it to derive Ground Delay Programs, and their duration. He then integrated these predictions within a decision making tool that allowed for better planning to reduce the effect of weather on arrival flow.

Avijit et al. developed an optimization algorithm to assign flight departure delays under probabilistic airport capacity. The algorithm dynamically adapted to weather forecasts by revising, when necessary, departure delays. San Francisco International Airport served as a use case. The algorithm was applied to assign departure delays to flights scheduled to arrive during the fog clearance time. Weather forecasts were obtained from an ensemble forecast system for predicting fog burn-off time developed by the National Weather Service (NWS) and MIT Lincoln Labs. Experimental results indicated that overall delays at San Francisco International Airport could be reduced by up to 25

IV. Methodology

The approach used is similar to the one used by Mangortey et al. [4]. It is composed of 6 successive steps presented in fig. 4. The subsections below will describe in details on how each step was implemented for this research.



Fig. 4 Overview of Proposed Methodology[4]

A. Data Identification and Acquisition

The following datasets containing Ground Delay Program and Ground Stop data, as well as weather data from January to August 2017 were identified and used for this research:

1. Traffic Flow Management System (TFMS)

Air traffic management personnel use the Traffic Flow Management System (TFMS) to implement traffic flow management initiatives. These are implemented to ensure that constrained areas in the National Airspace System (NAS) remain safe [11]. TFMS is composed of two components: TFMS Flight and TFMS Flow. TFMS Flight provides initial flight plan messages, amended flight plan messages, departure and arrival time notifications, flight cancellation messages, boundary crossing messages, and track position reports. TFMS Flow on the other hand, provides data on traffic flow management initiatives such as Ground Stops, Reroutes, Airspace Flow Programs, etc [11]. This data was obtained from the Federal Aviation Administration's (FAA) Computing Analytics and Shared Services Integrated

Environment (CASSIE). CASSIE brings FAA divisions, partners, and stakeholders together in a shared services environment consisting of Big Data, computing power, and analytical tools [3].

2. Automated Surface Observing Systems (ASOS)

The Automated Surface Observing Systems (ASOS) dataset provides weather conditions, which are widely used by meteorologists, climatologists, hydrologists, and aviation weather experts [12, 13]. In particular, this data provides a summary of airport weather conditions such as the date and time that the conditions were recorded as well as weather attributes such as ambient temperature, sea level pressure, visibility, wind speed, wind direction, wind gusts, dew point temperature, precipitation accumulation, cloud height and amount, etc. The ASOS data used for this research was obtained online in csv format [14].

B. Identification of Airport of Interest

As previously mentioned, the TFMS was data used to narrow down which airport should be investigated. The airport selection is important as the airport with a higher number of coincidence occurs will help create better machine learning models. As seen on fig. 5, LGA airport had the most interactions, therefore it was chosen. The criteria used to decide on the number of interactions at a given airport were the following:

- Number of GDPs with preceding GS
- Number of GDPs with internal GS
- Number of GS with following GDP
- Number of GS internal to GDP

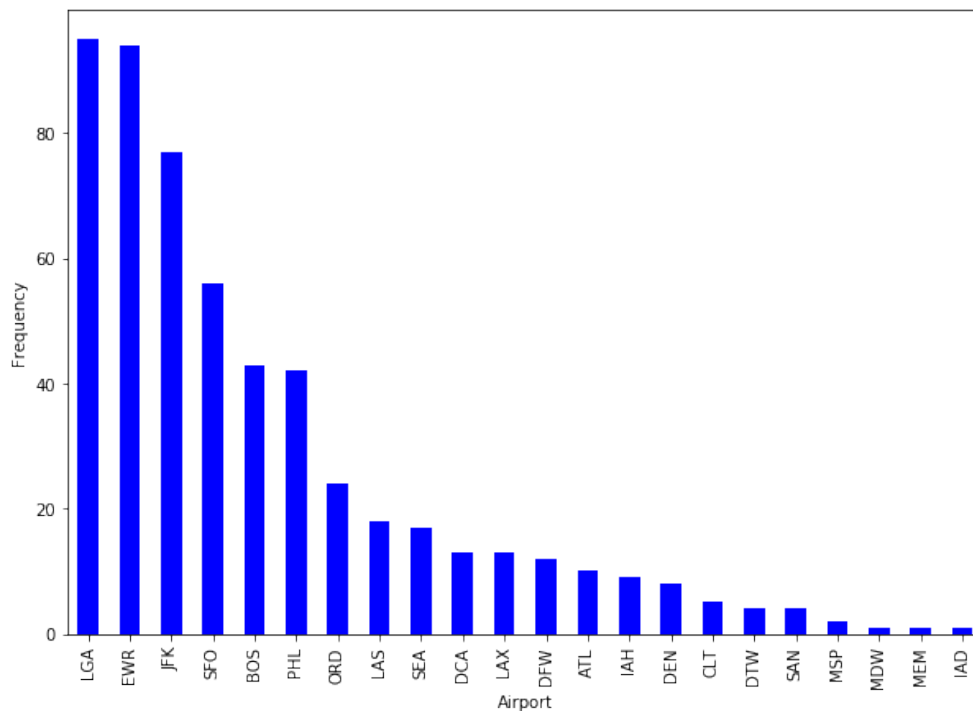


Fig. 5 Frequency of Interactions at Multiple Airports from January 2017-August 2017

C. Process Data

The data process is explained in the following paragraphs, they describe how the raw data was manipulated.

1. Traffic Flow Management System (TFMS)

The TFMS data are stored in the Flight Information Exchange Model (FIXM) format [15]. In order to perform any analytical work on the data, there is a need to convert it from the FIXM format to a CSV file. A Python parsing script was developed by Mangortey et al. for this task[4], and was reused to parse the FIXM data. The script follows the process in fig. 6. The output of the parser should be checked for duplicate rows, and data integrity. The next step after parsing is done, is to ensure adequate start and end times of the Ground Delay Programs and the Ground Stops. For each of these TMI, when updates are made, a new advisory is sent along with its start and end times. When this happens, there are overlapping advisories for the same TMI, which is inadequate. In order to fix this, the end time of the first advisory is set to be the start time of the second one, this process is highlighted in fig.7.

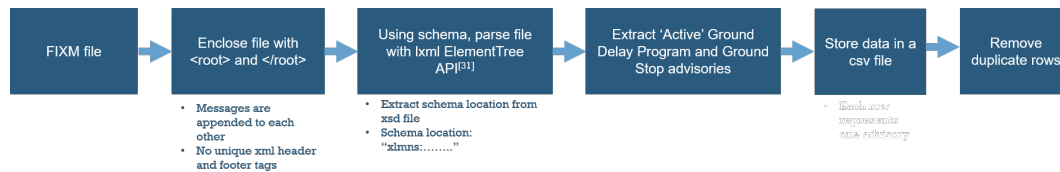


Fig. 6 Parser Steps [4]



Fig. 7 Advisory Start and End Times Fixing Example

2. Automated Surface Observing Systems (ASOS)

CSV files can be directly obtained from the ASOS database. Unfortunately, ASOS data contains a lot of missing values and features leading to a lot of them being removed from this analysis. The following parameters were extracted:


- Date and time
- Air Temperature (Fahrenheit)
- Dew Point Temperature (Fahrenheit)
- Relative Humidity (%)
- Wind Direction (Degrees)
- Wind Speed (Knots)
- Precipitation Accumulation (Inches)
- Pressure Altimeter (Inches)
- Visibility (Miles)
- Wind Gusts (Knots)
- Cloud Coverage Type
- Cloud Altitude (Feet)

D. Fuse Data

The next step in the methodology focuses on fusing the datasets by date and time. Data Fusion is a method of data analysis involving the combination of data from multiple sources to obtain more consistent information than that obtained from a single data source [16]. For this research, this was achieved by:

- 1) Fusing Ground Delay Program and Ground Stop data with weather conditions to generate non-coincident cases
- 2) Identifying coincident Ground Delay Program and Ground Stop advisories for the selected airport, and fusing with weather conditions to generate coincident cases
- 3) Including weather conditions from days without Ground Delay Programs or Ground Stops to generate additional non-coincident cases

Some Machine Learning techniques require numerical data rather than categorical data. Thus, after fusing the datasets, there was a need to encode categorical data into numerical data. This was done using One-Hot Encoding, where each unique categorical parameter was converted into a binary parameter [17? ?], as seen in Figure 8, where four binary variables were created from the four categories (dates).



Date	Delays
1/1/2015	34
1/2/2015	5
1/3/2015	26
1/4/2015	8

1/1/2015	1/2/2015	1/3/2015	1/4/2015	Delays
1	0	0	0	34
0	1	0	0	5
0	0	1	0	26
0	0	0	1	8

Fig. 8 One-Hot Encoding Process

The non-encoded variables used for this research were Pressure Altimeter (inches), Wind direction (degrees), Dew point temperature (Fahrenheit), Temperature (Fahrenheit), Precipitation (inches), Visibility (miles), Wind gust (knots) and Wind speed (knots). Encoded variables were the month of year, hour of day and details of the cause of the TMI (thunderstorms, wind, etc.).

E. Generate, and Test Models

This subsection discusses the steps taken to develop, tune and test prediction models for the following tasks:

- 1) Predicting the coincidence of Ground Stops and Ground Delay Programs
- 2) Predicting whether a Ground Delay Program will precede a Ground Stop, or vice versa, when coincidence occurs

Three base models were created using Python and open-source machine learning libraries such as Scikit-learn[18] and Keras/Tensflow[19]. These models were a Random Forest, an Ada Boosting Ensemble, and a Neural Network. Two of the models were selected because of their high performances in [4] and the last one (Neural Network) was added to the benchmarking exercise since it is commonly used in machine learning.

1. Training and Testing

Prediction of Ground Stop and Ground Delay Program Coincidence The data was split into two subsets with a common 80-20 ratio [20], corresponding training-validation and testing sets respectively. Given that there are more normal days in the dataset, there is a heavy class imbalance. There exists methods to deal with class imbalance during the training. The Synthetic Minority Over-sampling technique (SMOTE) algorithm is one of them, and was used to create synthetic data of the minority class for the training-validation subset. The SMOTE algorithm works by randomly selecting a k-nearest-neighbor of each member of the minority class. Implementing the SMOTE algorithm instead of naively oversampling the minority class ensures that overfitting is avoided. However, the SMOTE algorithm cannot be used for very large datasets [21]. Fig. 11 shows the class proportions before and after using SMOTE.

After dealing with the imbalance of the data, a 3 fold stratified cross-validation was used. This is useful to see how the created model performs given unseen datasets. The stratified cross-validation helps keeping the proportion of the classes in the split consistent. Cross-validation works by alternatively training on different portions of the dataset, and testing on the portion that was held out and not included in the training, as shown in fig. 9. Each base model with varying hyperparameter settings were trained and validated this, such that the optimal model can be determined. For this research a three-fold cross-validation was used to limit the number of computations needed to train each algorithm.

Ground Delay Program precedence before Ground Stop, or vice versa, when coincidence occurs Similarly, the same process was used (fig. 10) but the problem was changed from a binary classification problem to a multi-class one. Indeed, the targets for this case were normal days, GDP preceding GS when a coincidence occurs, and GS preceding GDP when a coincidence occurs. The SMOTE algorithm was again used to balance the dataset. Fig. 12 shows histograms of the training-validation and the test sets. Because of the inherent nature of a multi-class problem, a minority class is balanced by creating as many samples as needed to reach the same number of instances as the rest of

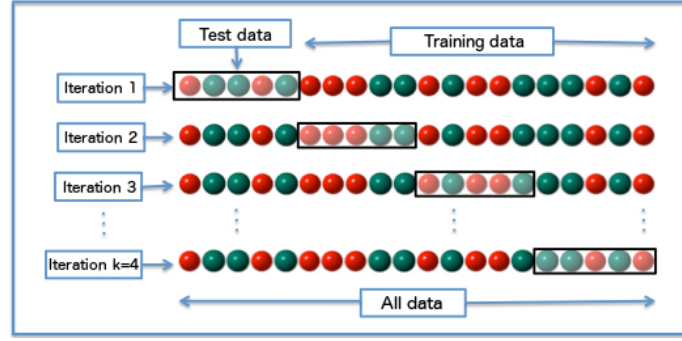


Fig. 9 K Fold Cross-Validation

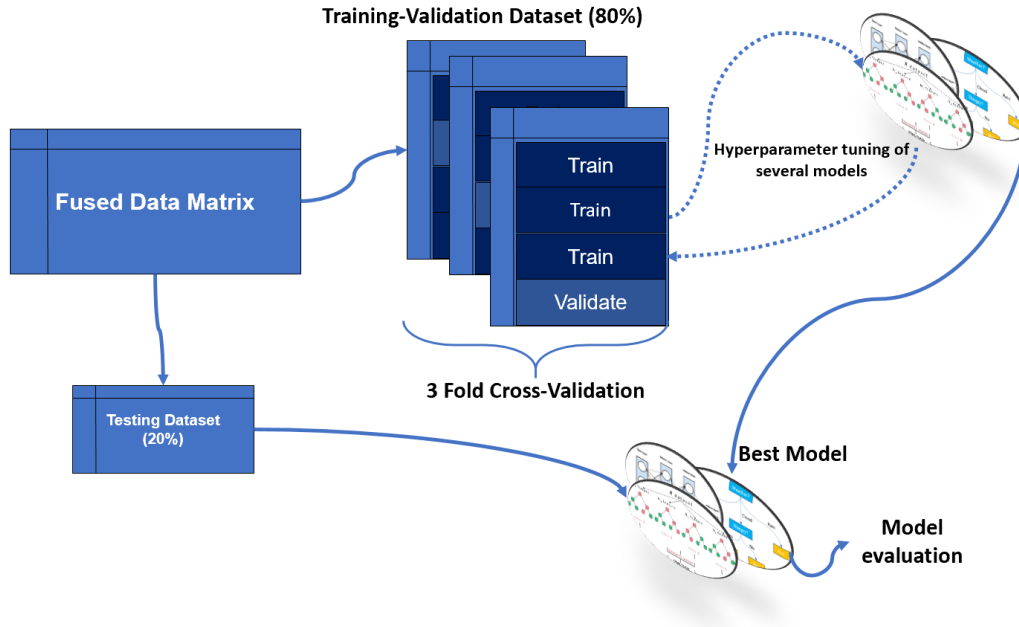


Fig. 10 Overview of model generation, validation, and testing process

the data, which includes the majority class and the other minority classes, if any. This was done repeatedly for both minority classes,

F. Model Evaluations

Classification models are commonly evaluated using confusion matrices. A symbolic confusing matrix is shown in table 2. The confusion matrix essentially just shows the number of correct and incorrect predictions for each class. Using this matrix, other common metrics used for this work are described below:

1. Accuracy

This refers to the ratio of the number of true positives and negatives, to the total number of predictions. Accuracy varies from 0 to 1 and is specified as [22]:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

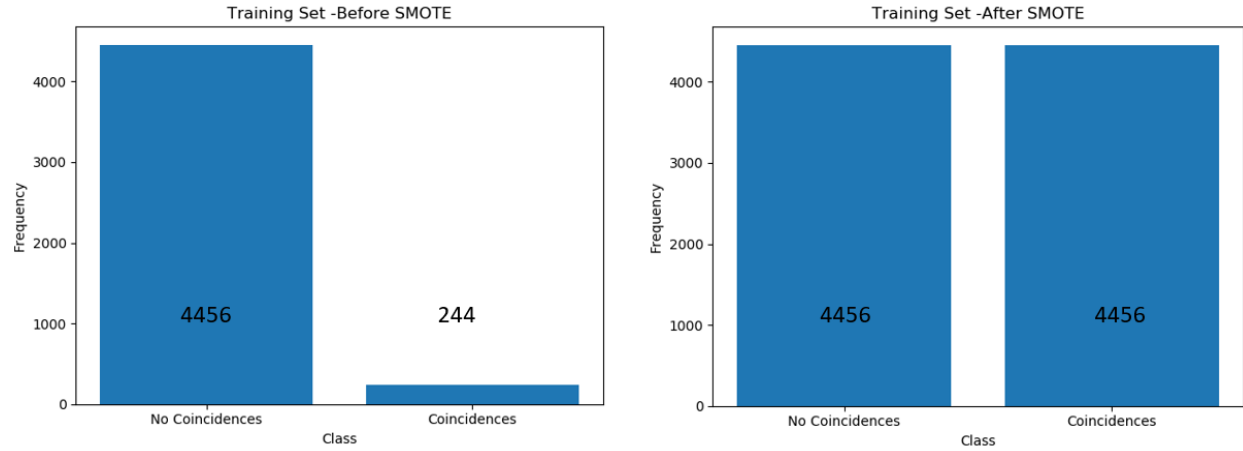


Fig. 11 Distribution of classes for predicting the coincidence of GDP and GS

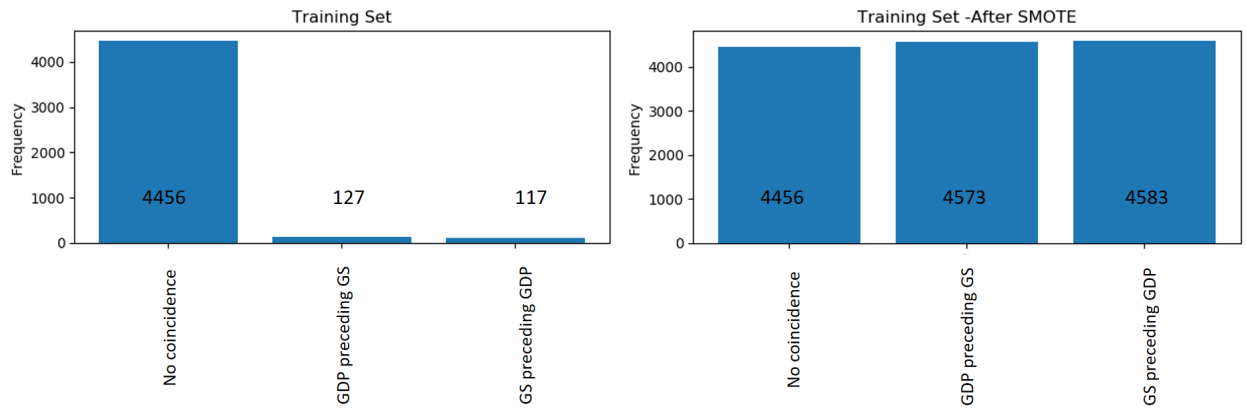


Fig. 12 Distribution of classes for predicting the precedence of GDP before GS, or vice versa, when coincidence occurs

2. Sensitivity

This refers to the proportion of true positives that were correctly classified. Sensitivity varies from 0 to 1 and is specified as [22]:

$$Sensitivity = \frac{TP}{TP + FN}$$

3. Specificity

This refers to the proportion of negative examples that were correctly classified. Specificity varies from 0 to 1 and is specified as [22]:

$$Specificity = \frac{TN}{FP + TN}$$

4. Kappa Statistic

A model might have high accuracy because it correctly predicts the most frequent class, particularly when the dataset is imbalanced. Kappa Statistic adjusts accuracy by accounting for the probability of a correct prediction by chance alone, and is appropriate for imbalanced datasets. Kappa Statistic is specified below, where P_0 is the observed value and P_E is the expected value [23]. It is specified as:

$$K = \frac{P_0 - P_E}{1 - P_E}$$

5. Balanced Accuracy

A model might have high accuracy because it correctly predicts the most frequent class, particularly when the dataset is imbalanced. Balanced accuracy adjusts accuracy by calculating the average of accurate predictions in each class [24] and is specified as:

$$\text{Balanced Accuracy} = \frac{\text{Sensitivity} + \text{Specificity}}{2}$$

The True Positive (TP) refers to the correct classification of the class of interest, and true Negative (TN) refers to the correct classification of the class that is not of interest. The false Positive (FP) refers to the incorrect classification of the class of interest, and the false Negative (FN) refers to the incorrect classification of the class that is not of interest [22].

Table 2 Confusion Matrix

	Actual True	Actual False
Predicted True	TP	FN
Predicted False	FP	TN

V. Analysis

This section describes the analyses performed. We first start with a brief but useful explanatory data analysis then an explanation of how a surrogate tree model and partial dependence plots can be used to turn the trained models into white boxes.

A. Explanatory Data Analysis

The fused dataset contained information from the TFMS and ASOS databases. Table 3 shows the original number of instances for each class. With no surprises, the number of normal hours is the majority, while the second largest number of hours is associated with GDPs. Interestingly enough, there are more instances of coincidence of GS and GDP than GS alone. This suggest that GS are more commonly used with other TMIs.

Table 3 Number of Instances For Each Class

Normal	GDP	GS	Coincidence	Total
4765	725	85	300	5875

One of the categorical feature in the dataset provide the reason why the TMI was implemented, fig 13 shows the distribution of the causes. Low cloud ceilings are unanimously the main meteorological reason why GDP and GS were implemented for this period of time, it is followed by low visibility. Snow and ice seems to be a third reason why GDPs were implemented but it does not make the top 3 for the GSs.

The times at which the GDPs, GSs, and their coincidences happened was also investigated. From fig. 14, we see that coincidences seems to happen more frequently late during the day or early in the morning. We can also see that the

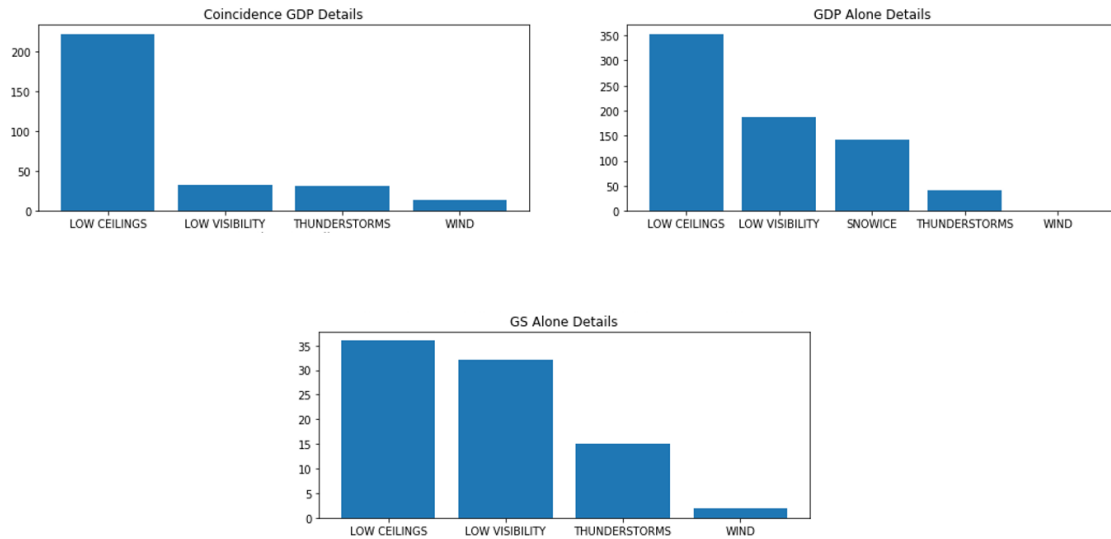


Fig. 13 Causes of Ground Delay Programs, Ground Stops and their coincidence at LGA airport between January and August 2017

time of occurrence of the GDP vary much more than the time of occurrence of the GS. Ground Stops occur more in the afternoon and early evening. As previously mentioned, the data for this work ranges from January 2017 to August of the same year. A trend discovered in the dataset was the fact that GSs happened mainly in the summer time, resulting in more frequent coincidences around that time as well, while GDPs were observed across all months. This can be seen on fig. 15.

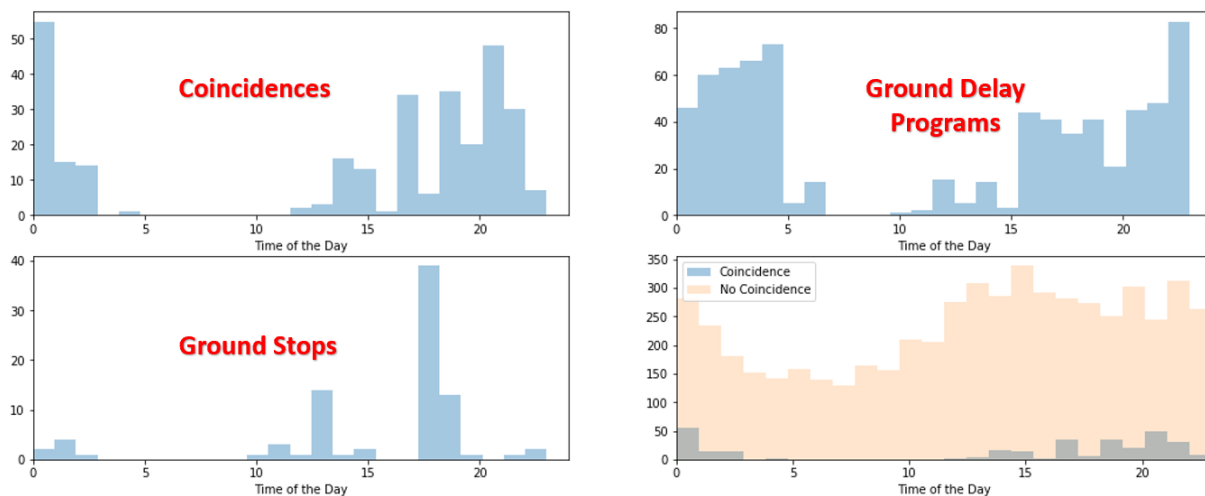


Fig. 14 Hourly distribution of the occurrence of Ground Delay Programs, Ground Stops and their coincidence at LGA airport between January and August 2017

When Ground stops are implemented, probabilities of extension are provided. It was interesting, but expected, to see that there are more coincidence cases when the probability of extension of the Ground Stops was high. This suggested that every GSs with a high probability of extension ended up coinciding with GDPs. The histograms capturing this information are presented in fig. 16.

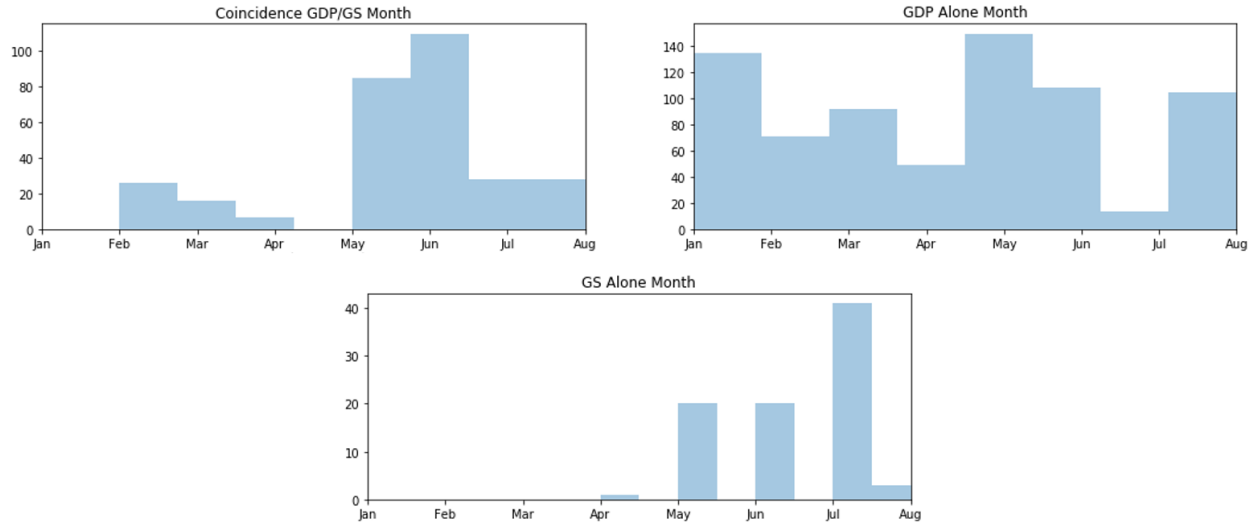


Fig. 15 Monthly distribution of the occurrence of Ground Delay Programs, Ground Stops and their coincidence at LGA airport between January and August 2017

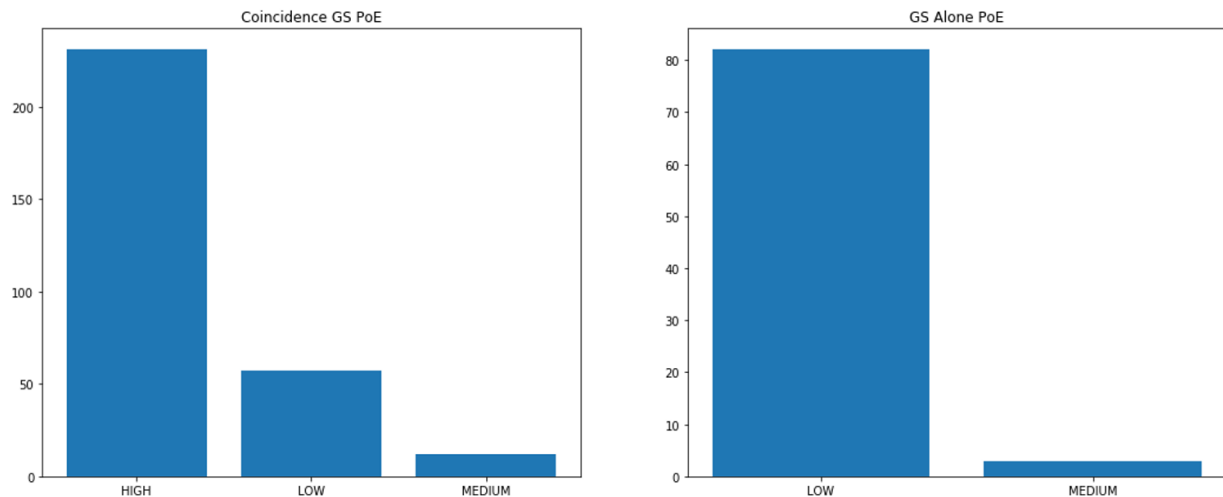


Fig. 16 Probability of Extension For Coincidences and Ground Stops Alone

B. Tree Surrogate Model

A surrogate model tree can be used to turn the complex black box model that was used to make the prediction into a white box one[25]. Therefore, this surrogate tries to find underlying relationships between the original input features, and the predicted output target of the black box model. The surrogate tree allows for a global interpretation of the black-box model. It does not try to find a relationship between the original inputs and the actual output, this is an important distinction to make. The tree was created using the Scikit-learn library. Once it is trained, it is possible to create an image representing the structure of the tree similar to fig. 17.

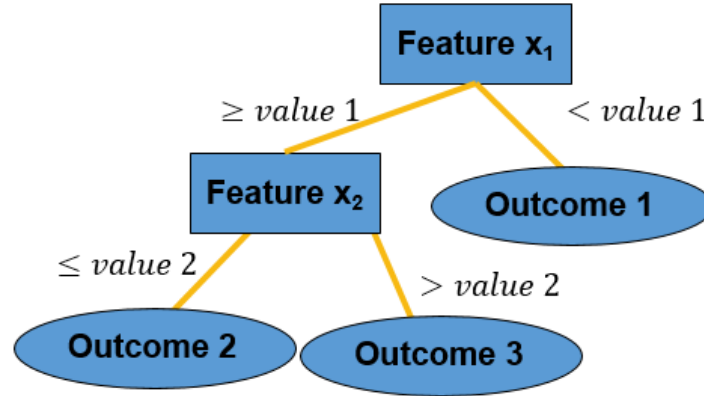


Fig. 17 Decision Tree Example

C. Partial Dependence Plots

Partial dependence plots (PDP) are used because they help with describing the marginal impact of a given feature on a model prediction, while holding other features constant [26]. It is useful to use PDP to understand the relationships between a feature and a target (linear, monotonic, or more complex). We can assess the variability of our target given small changes in a selected features. The Skater [27] library will be use to create the PDP, it is an open source unified framework to enable model interpretation for all forms of model, with the goal of making machine learning models more interpretable for real world use-cases. The library will be used to investigate the top 5 features of the best model, and look at their PDPs. In practice, the PDPs are found by repeatably setting all the instances of a feature of interest to a single value, and a prediction using this modified dataset. The changes in the target feature can then be observed with respect to the feature of interest. The features for which PDPs were created are the ones that had the most importance.

VI. Results & Discussion

This section presents the obtained results for each model for each of the cases along with the the analysis that has been performed.

A. Model Performances

1. Prediction of coincidence between Ground Stops and Ground Delay Programs

A grid search was performed for each of the base models (neural network, random forest, and ADA boosting ensemble). The search explored up to 16 hyperparameter combinations for the random forest model and as low as 6 hyperparameter combinations for the neural network. A lower number of combinations was chosen for the neural network because of the time needed to train. The hyperparameters that were changed are listed in table 4. The last row also shows the best estimator that was returned for each model, which corresponds to the base model with the best set of hyperparameters. The performances of the best estimator was then evaluated for all the previously mentioned metrics, and summarized in table 5

The models performed relatively well. Tables 5, 6, 7, and 8 show the metrics and the confusion matrix obtained on the testing set, which had 1,119 normal data points and 56 coincidences. Comparisons between the model results, show that the Random Forest model has overall better performances, and is the best model for this case.

2. Ground Delay Program Preceding Ground Stop, or vice versa, when coincidence occurs

Similar to the previous case, a grid search was also performed using the same possible combinations and the best models were returned and evaluated. Tables 10, 11, 12 and 13 show the metrics and confusion matrices obtained from the evaluation on the testing set. The neural network tend to evenly falsely predict normal and GS preceded instances

Table 4 List of hyperparameters for prediction of coincidence between Ground Stops and Ground Delay Programs

	Neural Network	Random Forest	Boosting Ensemble
Grid	Number of layers = [4, 6, 8] Activation functions = [Relu, Elu]	Max Depth = [30, 50, 70, 110] Number of estimators = [100, 200, 500, 1000]	Learning rate = [0.1, 0.001, 0.001] Number of estimators = [20, 50, 100, 200]
Number of Combinations	6	16	12
Best Estimator	Activation function = Relu Number of layers = 8	Max Depth=30 Number of estimator = 100	Learning rate = 0.1 Number of estimators = 100

Table 5 Comparison of Machine Learning algorithms using evaluation metrics

	Neural Network	Random Forest	Boosting Ensemble
Accuracy	0.967	0.997	0.996
Balanced Accuracy	0.949	0.973	0.964
Specificity	0.969	0.999	0.999
Sensitivity	0.929	0.946	0.929
Kappa Statistic	0.710	0.962	0.952

Table 6 Confusion Matrix for Neural Network (Case 1)

	Actual True	Actual False
Predicted True	52	4
Predicted False	35	1084

Table 7 Confusion Matrix for Random Forest (Case 1)

	Actual True	Actual False
Predicted True	53	3
Predicted False	1	1118

Table 8 Confusion Matrix for Boosting Ensemble (Case 1)

	Actual True	Actual False
Predicted True	52	4
Predicted False	1	1118

when there is a coincidence preceded by a GDP, while it falsely predict more normal instances than GDP preceded coincidence when there is a GS preceded one.

The Random Forest performs much better than other models by getting almost all the normal instances right, and by getting most of coincidence classified as a coincidence, though some of them are wrongly classified as the other type of coincidence. When there is a GS preceding a GDP when a coincidence occurs instance, the model falsely evenly

Table 9 List of Hyperparameters for Ground Delay Program Preceding Ground Stop, or vice versa, when coincidence occurs

	Neural Network	Random Forest	Boosting Ensemble
Grid	Number of layers = [4, 6, 8] Activation functions = [Relu, Elu]	Max Depth = [30, 50, 70, 110] Number of estimators = [100, 200, 500, 1000]	Learning rate = [0.1, 0.001, 0.001] Number of estimators = [20, 50, 100, 200]
Number of Combinations	6	16	12
Best Estimator	Activation function = Elu Number of layers = 4	Max Depth=30 Number of estimator = 100	Learning rate = 0.1 Number of estimators = 50

classified it as a normal or GDP preceding a GS when a coincidence occurs.

The Boosting Ensemble behaves has similar performances as the Random Forest except when predicting that a GS preceded a GDP when a coincidence occurs. It seems like the algorithm wrongly predicts more GDP preceded coincidence than normal instances.

Table 10 Metric Comparisons (GDP precedence of GS, and vice versa, when coincidence occurs)

	Neural Network	Random Forest	Boosting Ensemble
Accuracy	0.978	0.987	0.986
Balanced Accuracy	0.760	0.833	0.832
Kappa Statistic	0.746	0.856	0.841
Sensitivity (GDP Preceded)	0.607	0.75	0.714
Sensitivity (GS Preceded)	0.679	0.75	0.786
Specificity	0.995	0.999	0.997

Table 11 Confusion Matrix for Neural Network (Case 2)

	Actual Normal	Actual GDP Preceded	Actual GS Preceded
Predicted Normal	1113	4	2
Predicted GDP Preceded	6	17	5
Predicted GS Preceded	6	3	19

Table 12 Confusion Matrix for Random Forest (Case 2)

	Actual Normal	Actual GDP Preceded	Actual GS Preceded
Predicted Normal	1118	0	1
Predicted GDP Preceded	1	21	6
Predicted GS Preceded	4	3	21

Table 13 Confusion Matrix for Boosting Ensemble (Case 2)

	Actual Normal	Actual GDP Preceded	Actual GS Preceded
Predicted Normal	1116	0	3
Predicted GDP Preceded	2	20	6
Predicted GS Preceded	2	4	22

B. Surrogate Tree Model

1. Prediction of coincidence between Ground Stops and Ground Delay Program

The Random Forest model was selected as the best model for this case as it had the best kappa statistic. A random forest model is a combination of multiple trees, therefore it is useful to make it more interpretable using only one tree. The surrogate tree was trained and evaluated, The kappa statistic of the surrogate for different tree depths are presented in fig. 18. The deeper the tree the better the model but the harder the interpretation becomes. It is worth nothing that the best estimator for this classification problem can be reasonably approximated with a tree of only 3 levels, as seen on Fig. 18 and 19. Every node of the tree is a Boolean expression that checks if the value of a feature is less than or equal to a given value. Each evaluation at a node leads to a path, and the final classification choice is show on the tree leaf. For example in fig. 19, any Traffic Flow Management personnel would be able to know if whether or not the weather situation could lead to a coincidence between a GS and a GDP by following the tree path:

- 1) Weather_THUNDERSTOM is less than zero (According the encoding used, this correspond to the absence of a storm)
- 2) Weather_LOW CEILING is less than zero (This corresponds to the cloud ceiling level not being low)
- 3) Weather_LOW VISIBILITY is less than zero (the visibility is good)
- 4) The instance will than be classified as class 0, which corresponded to no coincidence

This results intuitively makes sense, showing that the model decisions are legitimates. The entropy number in each node shows how pure a node is, the lower the purer. The entropy is quite high for some of the leaves of the tree but this is due to the fact that this surrogate tree is not a perfect representation of the best model but also because the best model still makes some errors during classification. On the figure, the darker colors are correlated with lower entropy while the lighter ones are correlated with higher entropy. This allows a quick visual to see how good of a split a node performed.

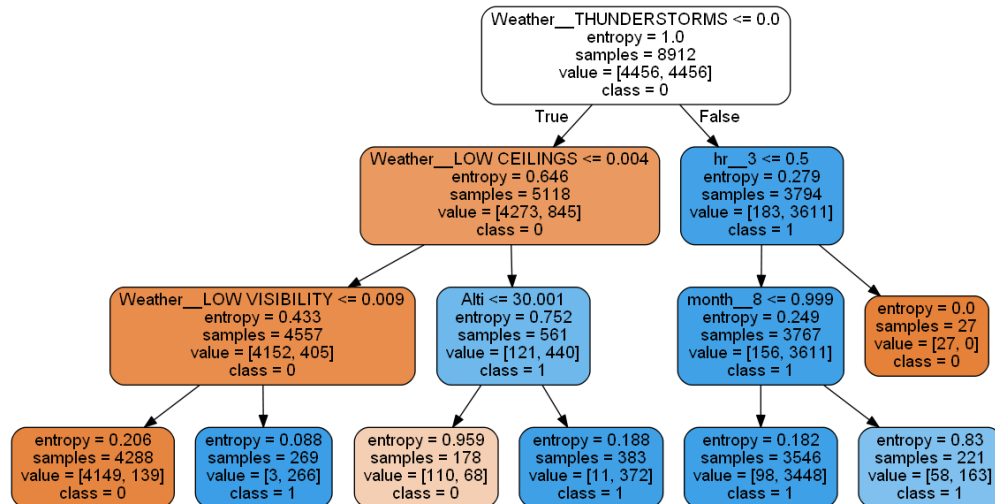


Fig. 18 Decision Tree With a Maximum Depth of 3 (Case 1)

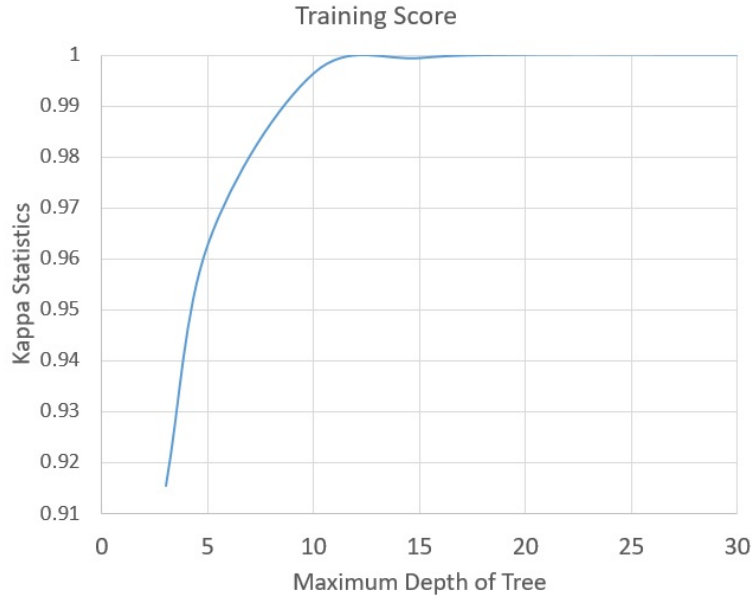


Fig. 19 Kappa Statistic of Surrogate tree for Different Tree Depths

2. Ground Delay Program Preceding Ground Stop, or vice versa, when coincidence occurs

As in the previous case, tree surrogates were created to explain the black-box obtained. During training, the performance the tree were similar to the previous case. The deeper the tree was, the better it was to explain the selected model. Fig. 20 shows a tree of depth 3 that Traffic Management Personal could use. In this case, more colors are available because we have more classes but the same logic is applied in terms of darker and lighter coloring and its correlation with the entropy.

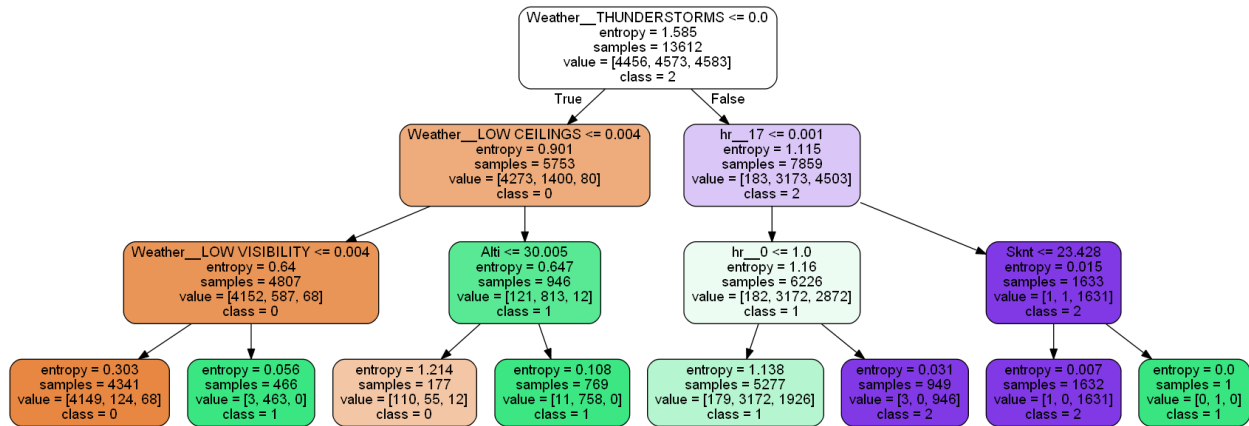


Fig. 20 Decision Tree With a Maximum Depth of 3 (Case 2)

C. Partial Dependence Plots

1. Prediction of coincidence between Ground Stops and Ground Delay Programs

The feature importance of the best model was determine, and is presented in Fig 21. The figure shows that the presence of a thunderstorm, presence of a low cloud ceiling, and the altimeter pressure were the most important features that help the model make decisions. With that knowledge, we can then create PDPs for those predictors.

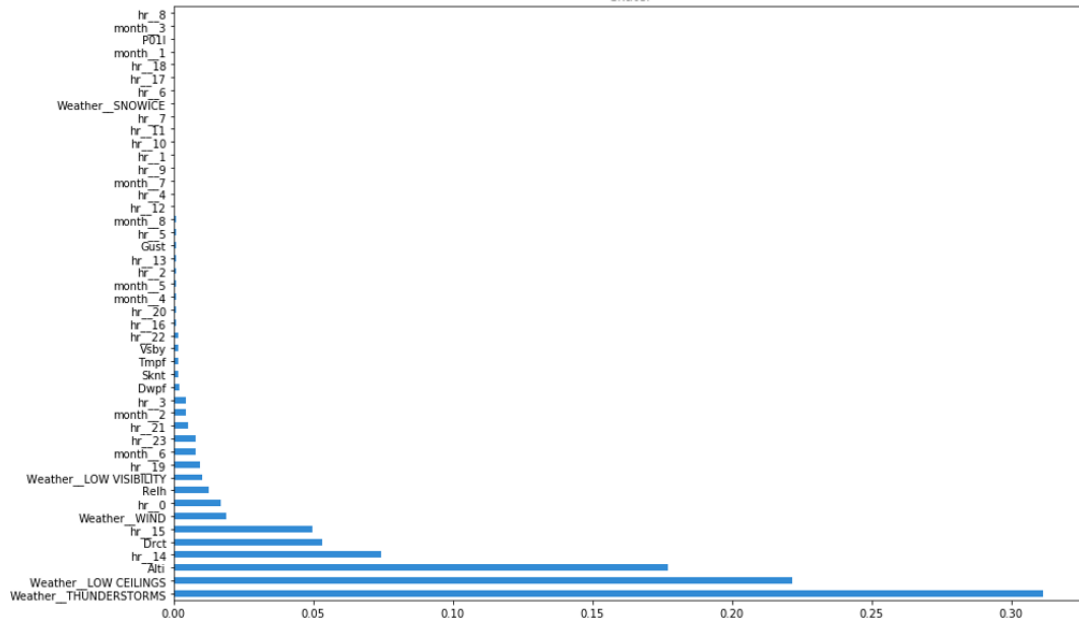


Fig. 21 Feature importance of Random Forests algorithm for predicting the coincidence of GDP and GS

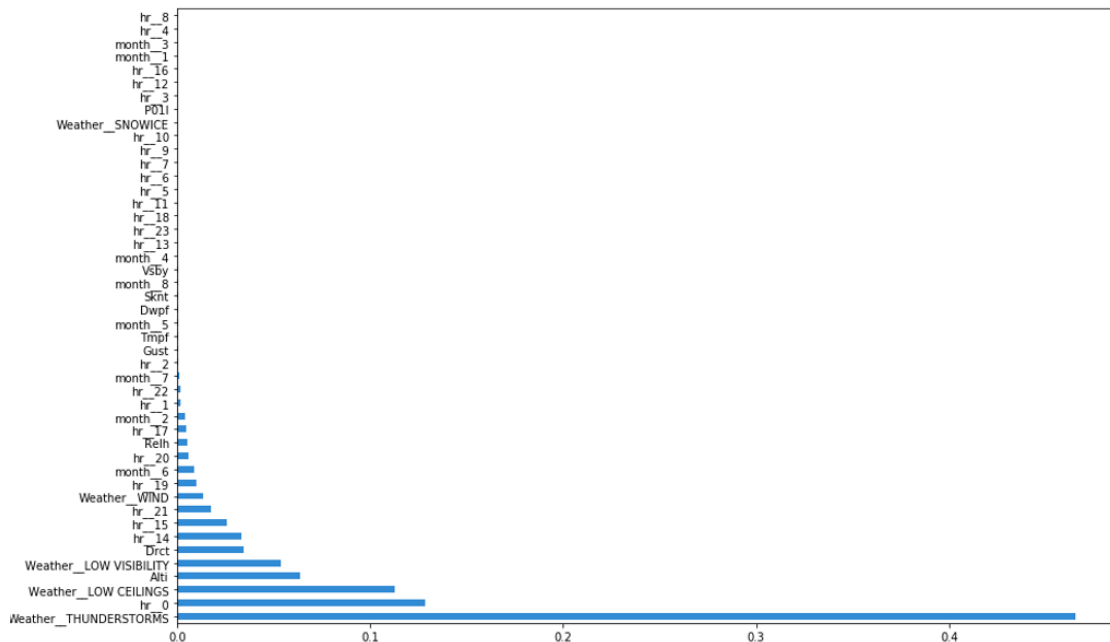


Fig. 22 Feature importance of Random Forests algorithm for predicting the whether a Ground Delay Program will precede a Ground Stop, or vice versa, when coincidence occurs

The partial dependence plots for the thunderstorm and the low ceiling features show that the probability of having a coincidence is greater when either of these feature is true, as seen on Fig. 23 and 24. Indeed, the trained model predicted more coincidences when the the categorical variables were individuality set to true. This results makes sense though, a insight from that is that the overall probability of obtaining a coincidence is low in all cases.

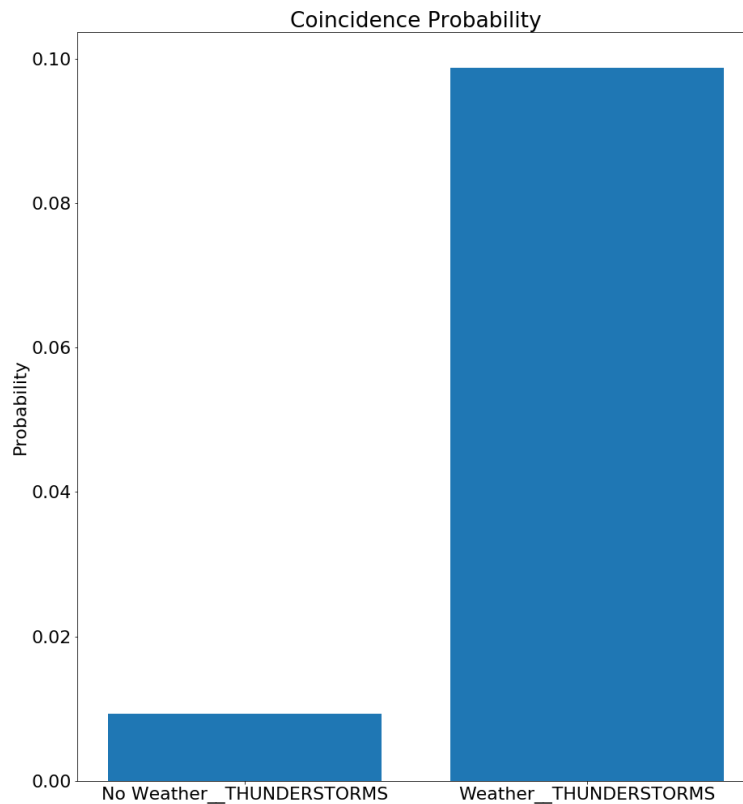


Fig. 23 Partial Dependence Plots for predicting the coincidence of GDP and GS (Thunderstorm)

2. Ground Delay Program Preceding Ground Stop, or vice versa, when coincidence occurs

Only one feature in the top 3 most important feature changed. For this case, the top 3 were the presence or absence of a thunderstorm and low ceiling, along with the time at which the instance takes place. In particular, knowing if the instance happens at midnight was important. This insight can be linked back to the exploration data analysis, which showed that most coincidences happened at midnight. Again the probabilities are small, but this is because there are much higher chances of having a normal instance than a coincidence as seen in table 3. It is worth noting that the presence of a thunderstorm increases the probability of having a GS preceded coincidence more than it increases the probability of having a GDP preceded coincidence, as seen on Fig 25b. When there is a low cloud ceiling, there are slightly more chances of seeing a GDP preceded coincidence though when cloud ceiling is not there is less chances of the GDP preceded one to happen. Finally, there is no change in the probability of having a GDP preceded coincidence whether its midnight or not. The fact that this feature is important is because it plays a better role at increasing the probability of a GS preceded coincidence.

Overall the probability of obtaining a coincidence is smaller than the probability of obtaining a normal instance. However, the partial dependence plots, show that the model is susceptible to the changes in the important features. The PDPs were helpful to make us aware of the coincidences likelihood increments.

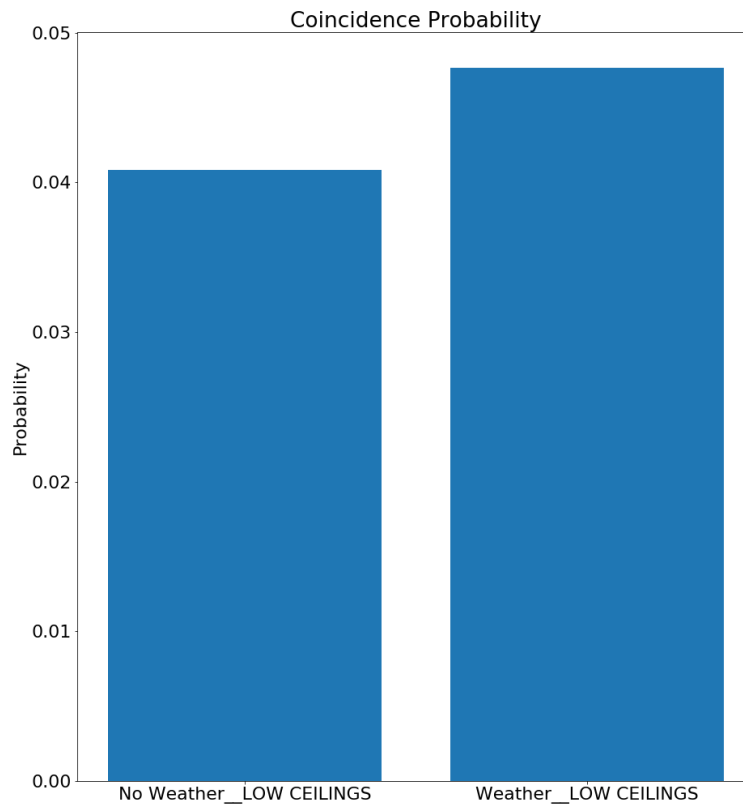
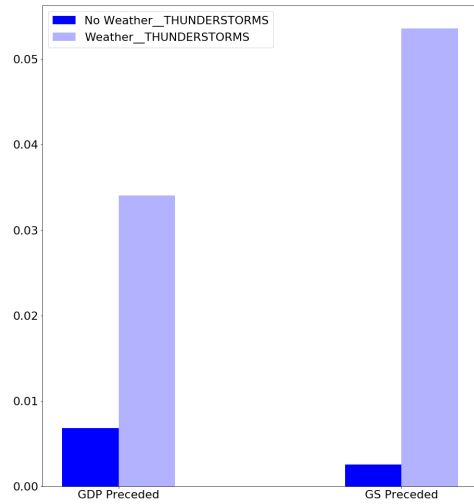


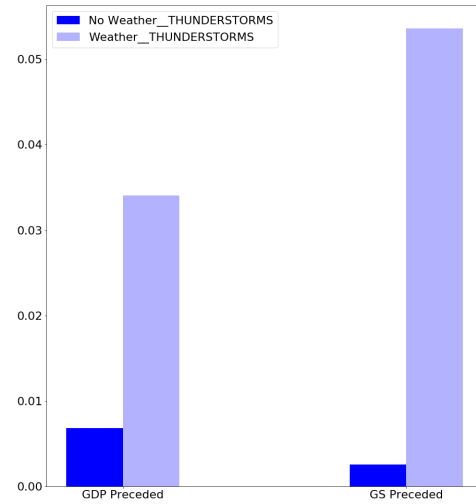
Fig. 24 Partial Dependence Plots for predicting the coincidence of GDP and GS (Low Ceinling)

VII. Conclusion

One of the most common type of delays are delays created by the implementation of Traffic Management Initiatives (TMIs). Traffic Management Initiatives are in place to control air traffic volume to specific airports, where the projected traffic demand is expected to exceed the airport's acceptance rate. These TMIs are commonly triggered by inclement weather, aircraft congestion, closed runways, etc. Ground Delay Programs and Ground Stops are implemented over lengthy and short periods of time, respectively. Occasionally, Ground Delay Programs and Ground Stops coincide, leading to further delays. This research develops and implements a methodology to predict and analyze the coincidence of weather-related Ground Delay Programs and Ground Stops. This work also focuses on predicting whether a Ground Delay Program will precede a Ground Stop, and vice versa, when coincidence occurs. This was achieved by 1) fusing Ground Delay Program and Ground Stop data from the Traffic Flow management System, and weather data from the Automated Surface Observing Systems, and 2) benchmarking Machine Learning algorithms to predict the tasks at hand. The Random Forest algorithm was identified as the best suited algorithm for predicting the coincidence of weather-related Ground Delay Programs and Ground Stops, and which Traffic Management Initiative would precede the other when coincidence occurs. Analysis of the models revealed that the top predictors for predicting the coincidence were thunderstorms, low ceilings and pressure altimeter. Indeed, the probability of coincidence increased to 9% and 4.5% whenever thunderstorms and low ceilings were present, respectively. The top predictors for predicting which TMIs will precede the other were thunderstorms, hour of day (midnight) and low ceilings. In particular, the likelihood of a Ground Stop preceding a Ground Delay Program when coincidence occurs is much higher with thunderstorms and at midnight, compared to a Ground Delay Program preceding a Ground Stop. It is expected that this methodology



(a) PDP for thunderstorms



(b) PDP for low ceilings

Fig. 25 Partial Dependence Plots for thunderstorms and low ceilings from Random Forest algorithm for predicting whether a GDP precedes a GS, or vice versa, when coincidence occurs

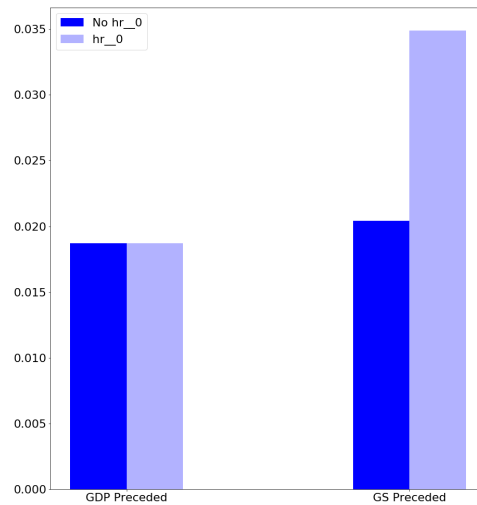


Fig. 26 Partial Dependence Plots for an hour (midnight) from Random Forest algorithm for predicting whether a GDP precedes a GS, or vice versa, when coincidence occurs

can be repeated for other airports and across different days to help stakeholders have a better understanding of this phenomenon.

References

- [1] Manley, B., and Sherry, L., “Analysis of performance and equity in ground delay programs,” *Transportation Research Part C: Emerging Technologies*, Vol. 18, No. 6, 2010, pp. 910 – 920. doi:<https://doi.org/10.1016/j.trc.2010.03.009>, URL <http://www.sciencedirect.com/science/article/pii/S0968090X10000355>, special issue on Transportation Simulation Advances in Air Transportation Research.
- [2] Federal Aviation Administration, “Traffic Flow Management in the National Airspace System,” , October 2009. URL https://www.fly.faa.gov/Products/Training/Traffic_Management_for_Pilots/TFM_in_the_NAS_Booklet_ca10.pdf.
- [3] Mangorthey, E., “PREDICTING THE OCCURENCE OF GROUND DELAY PROGRAMS AND THEIR IMPACT ON AIRPORT AND FLIGHT OPERATIONS,” Ph.D. thesis, Georgia Institute of Technology, May 2019.
- [4] Mangorthey, E., Pinon, O., Puranik, T., and Mavris, D., “Predicting The Occurrence of Weather And Volume Related Ground Delay Programs,” *AIAA AVIATION Forum*, 2019.
- [5] Federal Aviation Administration, “Air Traffic By the Numbers,” , June 2019. URL https://www.faa.gov/air_traffic/by_the_numbers/media/Air_Traffic_by_the_Numbers_2019.pdf.
- [6] National Business Aviation Association, “???”, URL <https://nbaa.org/aircraft-operations/airspace/tfm/tools-used-for-traffic-flow-management/>.
- [7] Murphy, K., *Machine Learning: a Probabilistic Perspective*, 2013.
- [8] Jixin, L., “Optimizing Key Parameters of Ground Delay Program with Uncertain Airport Capacity,” *Journal of Advanced Transportation*, Vol. 2017, No. 6, 2017, p. 19. doi:10.1155/2017/7494213, special issue on Transportation Simulation Advances in Air Transportation Research.
- [9] Wang, Y., “ANALYSIS AND PREDICTION OF WEATHER IMPACTED GROUND STOP OPERATIONS,” *33rd Digital Avionics Systems Conference*, 2014.
- [10] Smith, D. A., “DECISION SUPPORT TOOL FOR PREDICTING AIRCRAFT ARRIVAL RATES FROM WEATHER FORECASTS,” Ph.D. thesis, Georgia Mason University, 2008.
- [11] Federal Aviation Administration, *JAVA MESSAGING SERVICE DESCRIPTION DOCUMENT Traffic Flow Management Data Service (TFMData) Vol. 2.0.5*, Federal Aviation Administration, 2016.
- [12] National Weather Service, “Automated Surface Observing Systems,” , 2019. <https://www.weather.gov/asos/asostech>.
- [13] Guttman, Nathaniel and Baker, Bruce, “Exploratory Analysis of the Difference between Temperature Observations Recorded by ASOS and Conventional Methods,” *Bulletin of American Meteorological Society*, 1996.
- [14] Iowa State University, “ASOS-AWOS-METAR Data Download,” , 2019. <https://mesonet.agron.iastate.edu/request/download.phtml>.
- [15] Lepori, Hubert, “Introduction to FIXM.” , 2017. URL <https://www.icao.int/MID/Documents/2017/SWIMInterregional/8.2IntroductiontoFIXM.pdf>.
- [16] Klein, Lawrence A, *Sensor and data fusion: a tool for information assessment and decision making*, SPIE, 2012.
- [17] Feurer, Matthias and Klein, Aaron and Eggenesperger, Katharina and Springenberg, Jost and Blum, Manuel and Hutter, Frank, “Efficient and Robust Automated Machine Learning,” *Advances in Neural Information Processing Systems* 28, edited by C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Curran Associates, Inc., 2015, pp. 2962–2970. URL <http://papers.nips.cc/paper/5872-efficient-and-robust-automated-machine-learning.pdf>.
- [18] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E., “Scikit-learn: Machine Learning in Python,” *Journal of Machine Learning Research*, Vol. 12, 2011, pp. 2825–2830.
- [19] Chollet, F., et al., “Keras,” <https://keras.io>, 2015.
- [20] Google, 2019. URL <https://developers.google.com/machine-learning/crash-course/training-and-test-sets/splitting-data>.

- [21] Kunert, R., “SMOTE explained for noobs - Synthetic Minority Over-sampling TEchnique line by line,” , ??? URL http://rikunert.com/SMOTE_explained.
- [22] Lantz, Brett, *Machine Learning with R: Discover How to Build Machine Learning Algorithms, Prepare Data, and Dig Deep into Data Prediction Techniques with R*, Packt Publishing, 2015.
- [23] McHugh, M., “Interrater reliability: the kappa statistic,” *Biochem Med (Zagreb)* .;22(3):276–282, 2012.
- [24] Brodersen, K. H., Ong, C. S., Stephan, K. E., and Buhmann, J. M., “The Balanced Accuracy and Its Posterior Distribution,” *2010 20th International Conference on Pattern Recognition*, 2010, pp. 3121–3124. doi:10.1109/ICPR.2010.764.
- [25] Molnar, C., “A Guide for Making Black Box Models Explainable,” , 2019. URL <https://christophm.github.io/interpretable-ml-book/global.html>.
- [26] Sarkar, D., 2019. URL <https://towardsdatascience.com/explainable-artificial-intelligence-part-3-hands-on-machine-learning-model-interpretation-e8ebe5afc608>.
- [27] Kramer, A., and Choudhary, P., “Model Interpretation with Skater,” , September 2018. URL <https://oracle.github.io/Skater/index.html>.