

Application of Machine Learning to the Analysis and Prediction of the Coincidence of Ground Delay Programs and Ground Stops

Eugene Mangortey*, Marc-Henri Bleu Laine*, Tejas G. Puranik†, Olivia J. Pinon‡, and Dimitri N. Mavris§
Georgia Institute of Technology, Atlanta, GA, 30332

Traffic Management Initiatives such as Ground Delay Programs and Ground Stops are implemented by traffic management personnel to control air traffic volume to constrained airports when traffic demand is projected to exceed the airports' acceptance rate due to conditions such as inclement weather, volume constraints, etc. Ground Delay Programs are issued for lengthy periods of time and aircraft are assigned departure times later than scheduled. Ground Stops on the other hand, are issued for short periods of time and aircraft are not permitted to land at the constrained airport. Occasionally, Ground Stops are issued during an ongoing Ground Delay Program, and vice versa, which hinders the efficient planning and implementation of these Traffic Management Initiatives. This research proposes a methodology to help stakeholders better capture the impact of the coincidence of weather related Ground Delay Programs and Ground Stops, and potentially help reduce the number and duration of such coincidences. This is achieved by leveraging Machine Learning techniques to predict their coincidence at a given hour, predict which Traffic Management Initiative would precede the other during their coincidence, and identify key predictors that cause their coincidence. The Random Forests Machine Learning algorithm was identified as the best suited algorithm for predicting the coincidence of weather-related Ground Delay Programs and Ground Stops, as well as the Traffic Management Initiative that would precede the other during their coincidence.

I. Nomenclature

<i>API</i>	=	Application Program Interface
<i>ASOS</i>	=	Automated Surface Observing Systems
<i>ASPM</i>	=	Aviation System Performance Metrics
<i>ATC</i>	=	Air Traffic Controllers
<i>CASSIE</i>	=	Computing Analytics and Shared Services Integrated Environment
<i>CSV</i>	=	Comma-Separated Value
<i>EDCT</i>	=	Expected Departure Clearance Times
<i>FIXM</i>	=	Flight Information Exchange Model
<i>FN</i>	=	False Negative
<i>FP</i>	=	False Positive
<i>GDP</i>	=	Ground Delay Program
<i>GS</i>	=	Ground Stop
<i>LGA</i>	=	LaGuardia Airport
<i>NAS</i>	=	National Airspace System
<i>SMOTE</i>	=	Synthetic Minority Over-sampling Technique
<i>TFMS</i>	=	Traffic Flow Management System
<i>TMI</i>	=	Traffic Management Initiative
<i>TN</i>	=	True Negative
<i>TP</i>	=	True Positive

*Graduate Research Assistant, Aerospace Systems Design Laboratory, School of Aerospace Engineering, AIAA Student Member

†Research Engineer II, Aerospace Systems Design Laboratory, School of Aerospace Engineering, AIAA Member

‡Senior Research Engineer, Aerospace Systems Design Laboratory, School of Aerospace Engineering, AIAA Senior Member

§Regents Professor for Advanced Systems Analysis, School of Aerospace Engineering, AIAA Fellow

II. Introduction

Air 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 [3–7]. 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)[8]

	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

A. Ground Delay Programs (GDP)

Ground Delay Programs are implemented whenever an airport is constrained by inclement weather, volume constraints, etc. over a long period of time [4, 5, 9]. Figure 1 provides an overview of projected air traffic demand and airport capacity at an airport, prior to and after the implementation of a Ground Delay Program. From Figure 1a, it can be seen that projected air traffic demand exceeds airport capacity between 17:30 and 22:30 due to constraints at the airport. Consequently, a Ground Delay Program is implemented to ensure that air traffic demand matches airport capacity, as seen in Figure 1b. During this time, all flights scheduled to arrive at the constrained airport are issued Expected Departure Clearance Times (EDCT), which are updated whenever conditions change. EDCT is the runway release time (“Wheels Off”) assigned to aircraft due to Traffic Management Initiatives that require holding aircraft on the ground at the departure airport [10]. Figure 2 provides an overview of steps taken to plan a Ground Delay Program at a constrained airport.

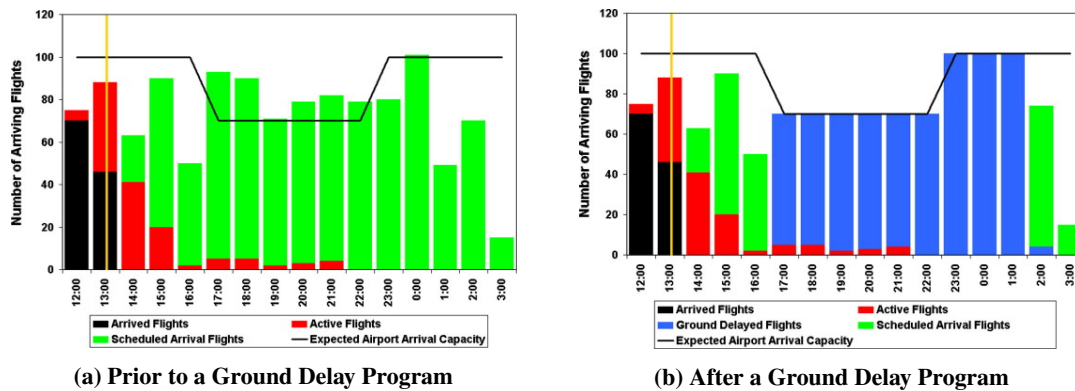


Fig. 1 Projected air traffic demand and airport capacity, prior to and after a Ground Delay Program [1]

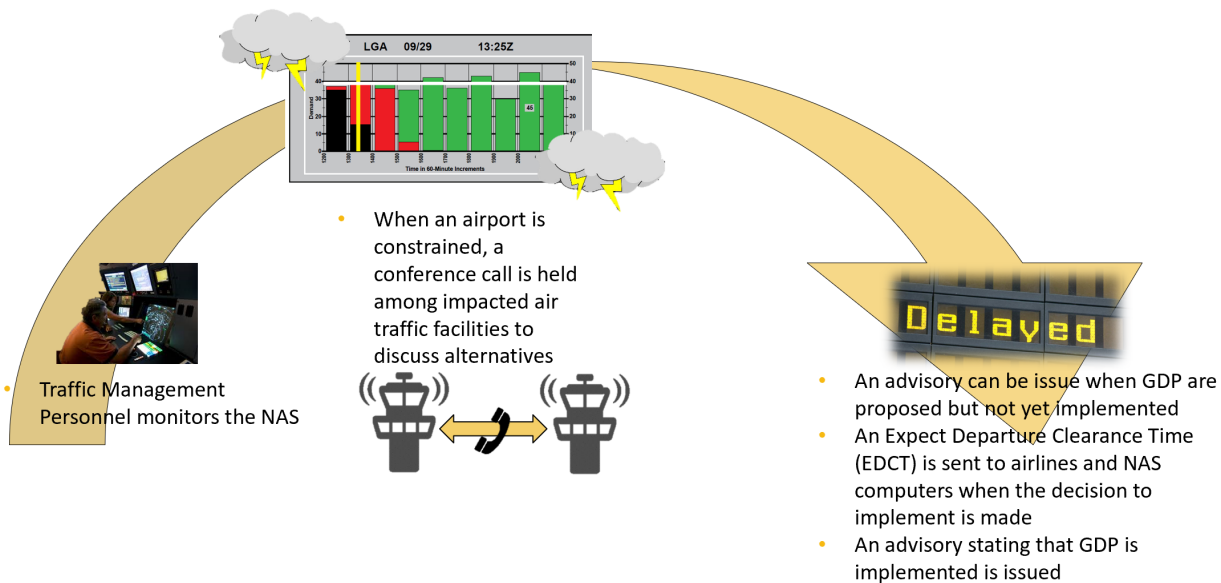


Fig. 2 Overview of steps taken to plan a Ground Delay Program [2]

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 [9]. 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.

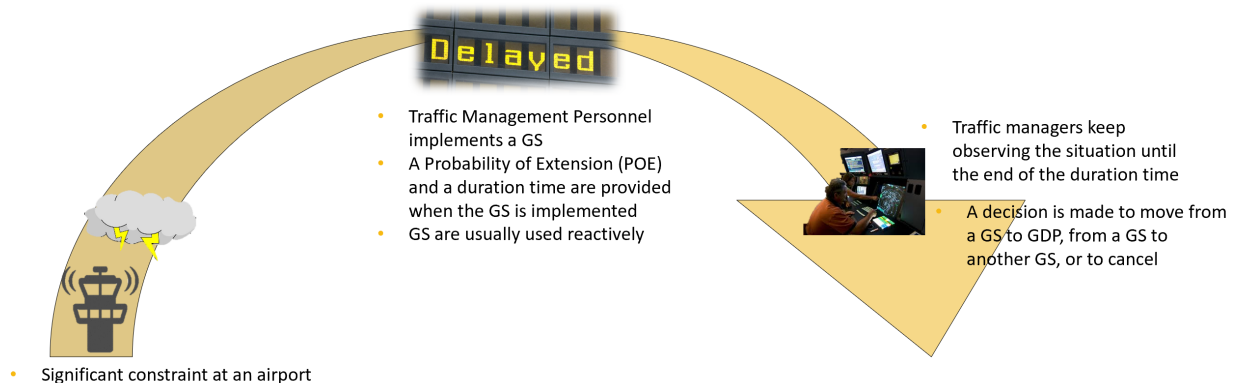


Fig. 3 Overview of steps taken to plan a Ground Stop [2]

III. Background

The coincidence of Ground Delay Programs and Ground Stops poses a challenge to the efficient planning and implementation of these Traffic Management Initiatives. The coincidence occurs when Ground Delay Programs are introduced during the implementation of Ground Stops, and vice versa. This is usually as a result of sudden changes in conditions at the constrained airport. This section discusses previous work related to Ground Delay Programs and Ground Stops, and outlines research gaps and the objectives of this work.

A. Previous Work related to Ground Delay Programs and Ground Stops

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. The model calculated the optimal Ground Delay Program file time, which was estimated to significantly reduce the delay times. Results showed that, when compared to 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 [11].

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% [12].

Wang generated a classification model using Ensemble Bagging Decision Trees to map historical airport weather forecast, scheduled traffic, and other airport conditions to implemented 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 [13]. In addition, Wang also determined that the coincidence of Traffic Management Initiatives affects the implementation of Ground Delay Programs.

Mangortey et al. developed prediction models to predict the occurrence of Ground Delay Programs and Ground Stops. This was achieved by fusing TMI data with airport data, and benchmarking Machine learning algorithms to identify the best suited ones for the tasks at hand [5, 14].

B. Research Gap and Objective

The implementation of a Ground Stop during an ongoing Ground Delay Program, and vice versa, hinders the efficient planning and implementation of these Traffic Management Initiatives. Thus, accurately predicting their coincidence may be beneficial to stakeholders. However, across the surveyed applications in the literature, it appears that previous work has focused on improving the implementation of Ground Delay Programs and Ground Stops, and no work has been conducted to analyze and predict the coincidence of weather-related Ground Delay Programs and Ground Stops. Consequently, the objective of this research is three-fold:

- 1) Predict the coincidence of weather-related Ground Delay Programs and Ground Stops
- 2) Predict whether a Ground Delay Program will precede a Ground Stop, or vice versa, when coincidence occurs
- 3) Identify factors that influence the coincidence of weather-related Ground Delay Programs and Ground Stops so as to help stakeholders better understand this phenomenon

Figure 4 shows that LaGuardia Airport (LGA) had the highest number of coinciding Ground Delay Programs and Ground Stops across multiple U.S. airports from January to August 2017. Consequently, this work was carried out using LGA data. Sections III, IV, and V discuss the methodology used, the results obtained, and provide some concluding remarks and avenues for future work, respectively.

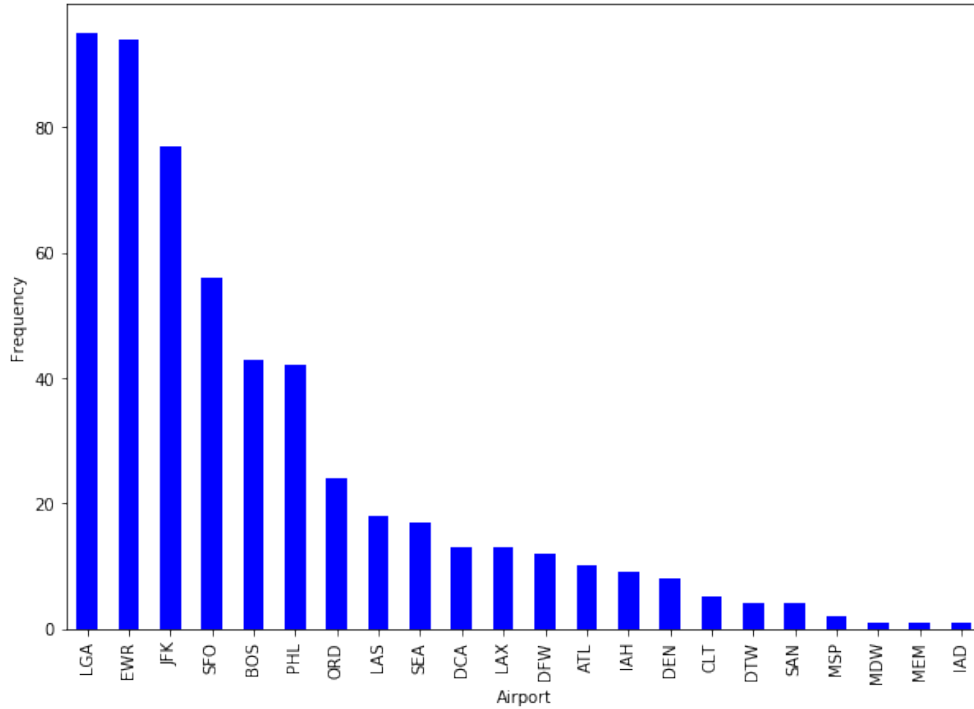


Fig. 4 Coincidence of GDP and GS at Multiple Airports from January - August 2017

IV. Methodology

Figure 5 provides an overview of the methodology used in this work which is discussed in detail in this section.



Fig. 5 Overview of methodology

A. Identify datasets

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 [15]. 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 [15]. 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 [7].

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 [16, 17]. 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 [18].

B. Parse datasets

This section discusses steps taken to parse the Traffic Flow Management System (TFMS) dataset into a format suitable for analytical purposes.

1. Traffic Flow Management System (TFMS)

The Traffic Flow Management System (TFMS) dataset is stored in Flight Information Exchange Model (FIXM) [19] format, which is widely used for storing and transmitting aviation data. These datasets are stored as hourly files containing advisories generated during that hour and need to be parsed from FIXM format to csv format. FIXM files have schema files, which dictate the structure of the files and should be parsed using their respective schema to ensure that all required fields are extracted in their correct format. This is done using a Python [20] parser developed by Mangortey et al. [4–6], which follows the process highlighted in Figure 6 and is described below:

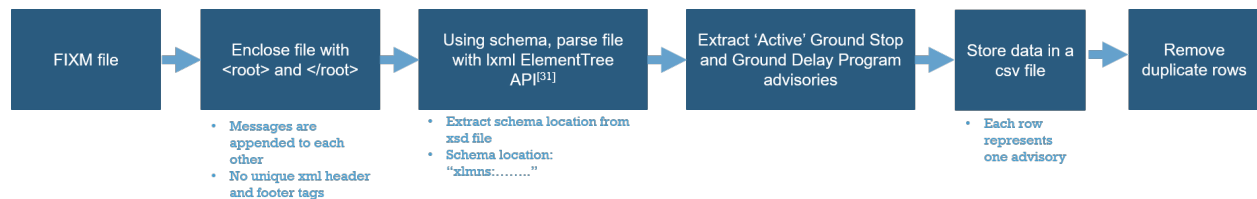


Fig. 6 FIXM to JSON conversion process

- 1) Since the dataset is comprised of advisories generated within the hour, there is no way to distinguish between the beginning of the file and the end of the file. Thus, it is important to enclose each file with a header and footer such as <root > and <\root > respectively to ensure that each file has unique starting and end points.
- 2) Extract the schema location from the xsd file. The schema location is typically of the format "xlmns:....."
- 3) Parse the FIXM file using the ElementTree [21] Application Program Interface (API)
- 4) Extract "Active" weather-related Ground Stop and Ground Delay Program advisories for the selected airport of interest
- 5) Store each advisory as a row in a csv file

Duplicate rows were then removed, and the data was analyzed to ensure that the dataset was parsed correctly. Parameters extracted for this work include the start and end dates and times of Ground Stops and Ground Delay Programs, and the detailed cause of the TMIs (thunderstorms, wind, etc.).

2. Automated Surface Observing Systems (ASOS)

The following parameters were extracted from the Automated Surface Observing Systems database in csv format:

- 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)

C. Clean datasets

The next step in the methodology focuses on identifying inconsistent and/or missing data and cleaning the datasets.

1. Traffic Flow Management System (TFMS)

- 1) The first step of the data cleaning process involves analyzing the data to ensure that fields are in their appropriate formats and do not contain any missing values or non-alphanumeric characters
- 2) The next step involves removing duplicate Ground Stop and Ground Delay Program advisories. Duplicate advisories exist because TFMS occasionally stores the same advisory multiple times
- 3) The duration and scope of an ongoing Ground Stop or Ground Delay Program may be modified whenever conditions change. This leads to overlapping advisories which is inaccurate. In order to address this inconsistency, the end time of the initial advisory is set as the start time of the new advisory as seen in Figure 7. In particular it shows that the start time of advisory number **0117** is prior to the end time of advisory number **0071**. Thus, the end time of advisory number **0071** is set as the start time of advisory number **0117**.

Advisory Number	Start Time	End Time
0071	2017-04-20T16:00:00Z	2017-04-21T03:00:00Z
0117	2017-04-20T18:15:00Z	2017-04-21T03:00:00Z

Advisory Number	Start Time	End Time
0071	2017-04-20T16:00:00Z	2017-04-20T18:15:00Z
0117	2017-04-20T18:15:00Z	2017-04-21T03:00:00Z

Fig. 7 Updating the end dates and times of an updated active advisory

2. Automated Surface Observing Systems (ASOS)

ASOS data is recorded in five minute intervals, and was analyzed to ensure that fields were in their appropriate formats and to identify any fields with missing values. In particular, rows containing missing values were deleted since the Machine Learning algorithms used for this work do not support data with missing values. In addition, fields such as cloud coverage type and altitude, ice accretion, and peak wind gust and direction were not used for this research as over 80% of these fields contained missing values.


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 [22]. 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 [23–25], 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. Develop prediction 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

Python and open-source Machine Learning libraries such as Scikit-learn[26] and Keras/Tensorflow[27] were leveraged for these tasks. Figure 9 provides an overview of this process. First, the fused data was randomly partitioned into two sets: training-validation and testing. 80% of the data was assigned to the training-validation set, which was used to generate and tune the models and 20% of the data was assigned to the test set, which was used to generate predictions for evaluations. The Neural Networks [28], Random Forests [29] and Boosting Ensemble algorithms [30] Machine Learning algorithms were benchmarked for this research based on their performance in related work [7].

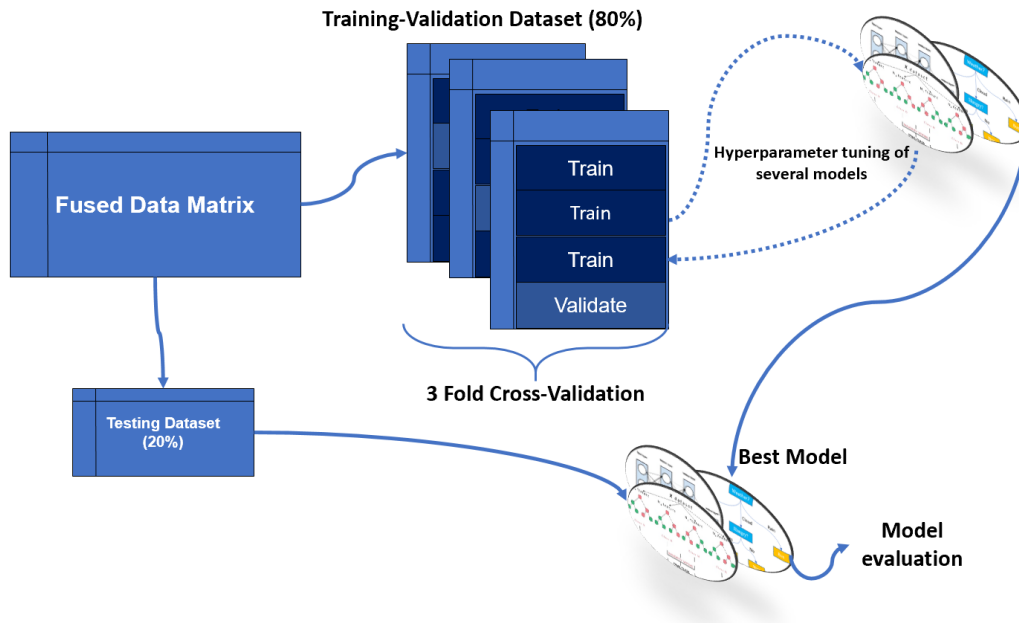


Fig. 9 Overview of model generation, validation, and testing process

As mentioned previously, 80% of the fused data was used to train and tune the prediction models. However, the datasets were heavily imbalanced, as the non-coincident cases greatly outnumbered the coincident cases. Imbalanced datasets often lead to poorly performing prediction models [5, 7]. Thus, the Synthetic Minority Over-sampling Technique

(SMOTE) algorithm [31] was leveraged to produce balanced training-validation sets. The SMOTE algorithm balances a dataset by increasing the minority class. This is achieved 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 over-fitting is avoided. However, the SMOTE algorithm cannot be used for very large datasets.

K-fold validation was then implemented to validate and tune hyperparameters of the different algorithms. During a k-fold cross-validation, a subset of the training-validation dataset is randomly held-out. The rest of the dataset is used to train the models, while varying combinations of algorithm hyperparameters. After training, the models are then validated using the previously held-out dataset. This process is repeated k times and the average performance across all folds is assessed to identify the optimal combination(s) of hyperparameters of the algorithms [29]. For this research a three-fold cross-validation was used to limit the number of computations needed to train each algorithm.

F. Evaluation of models

Evaluating the performance of prediction models is an important step as it informs as to how the model will perform on future data. Prediction models can be evaluated using results obtained from a confusion matrix, which categorizes predictions according to whether they match the actual value, as seen in Table 2.

Table 2 Confusion Matrix

	Actual: No	Actual: Yes
Predicted: No	True Negative (TN)	False Negative (FN)
Predicted: Yes	False Positive (FP)	True Positive (TP)

True Positive (TP) refers to the correct classification of the class of interest. True Negative (TN) refers to the correct classification of the class that is not of interest. False Positive (FP) refers to the incorrect classification of the class of interest. False Negative (FN) refers to the incorrect classification of the class that is not of interest [29]. The following performance metrics were then computed to assess model performance:

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 [29]:

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

2. Sensitivity

This refers to the proportion of true positives that were correctly classified. Sensitivity varies from 0 to 1 and is specified as [29]:

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

3. Specificity

This refers to the proportion of negative examples that were correctly classified. Specificity varies from 0 to 1 and is specified as [29]:

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

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 [32]. 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 [33] and is specified as:

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

V. Results and Discussion

Table 3 provides a summary of the distribution of classes in the fused dataset used for this research. In particular, it shows that the dataset is heavily imbalanced as the number of GDP and GS cases, as well as cases without GDP and GS greatly outnumber cases with coinciding GDP and GS. This justifies the use of the SMOTE technique to create a balanced training-validation set,

Table 3 Number of Instances For Each Class

Normal	GDP only	GS only	Coincidence	Total
4765	725	85	300	5875

Further analysis of the data revealed that low ceilings and low visibility were the most frequent causes of Ground Delay Programs, Ground Stops and their coincidence, as seen in Figure 10. Figure 11 shows the distribution of the occurrence of Ground Delay Programs, Ground Stops and their coincidence across all hours of the day. It shows that their coincidence was higher later in the day or early in the morning. It also shows that the occurrence of Ground Delay Programs varies across the day compared to Ground Stops which were predominant in the afternoon and early evening. Figure 12 shows that Ground Stops and their coincidence with Ground Delay Programs were more frequent over the summer, while Ground Delay Programs were distributed across all months.

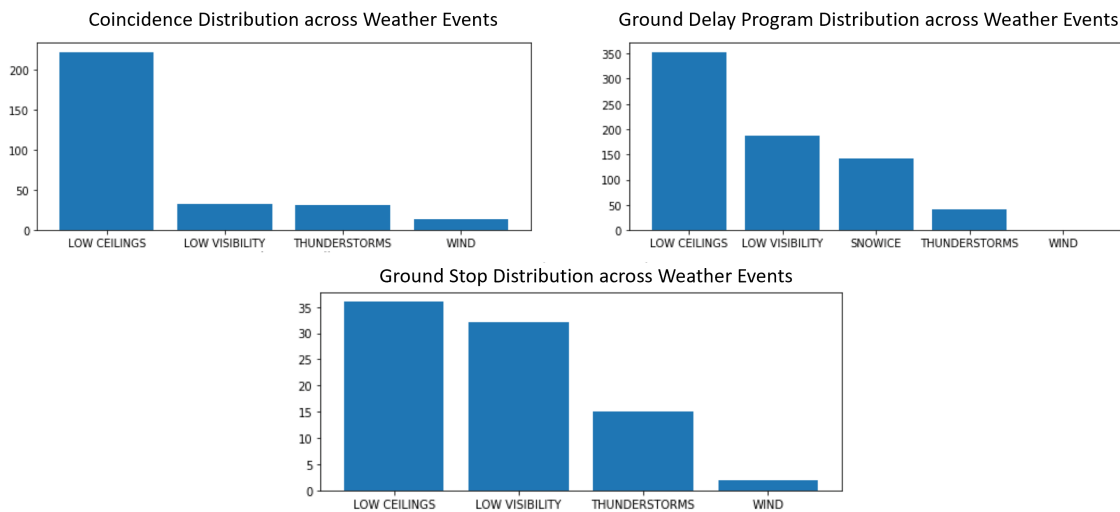


Fig. 10 Causes of Ground Delay Programs, Ground Stops and their coincidence at LGA airport between January and August 2017

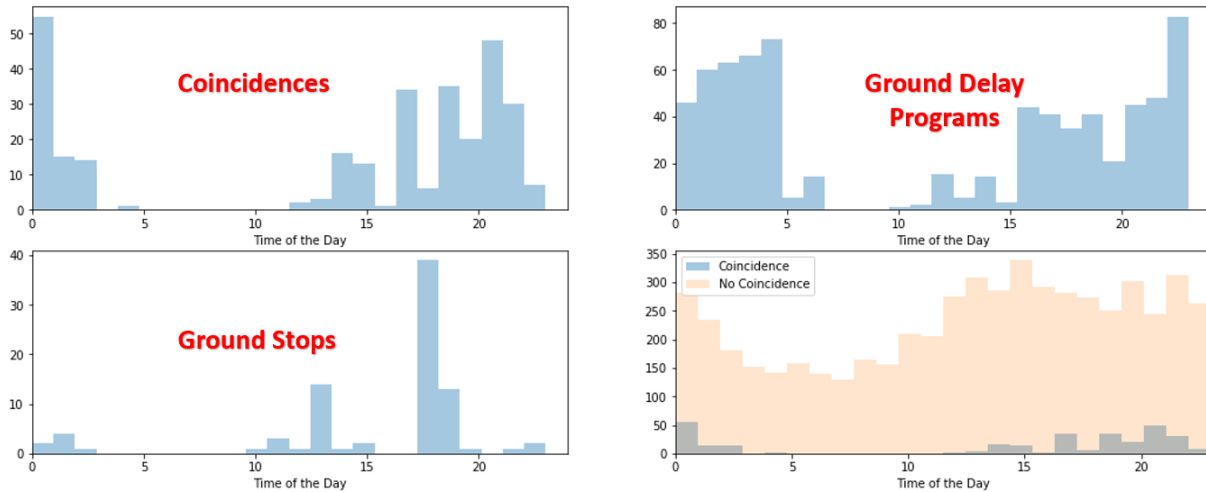


Fig. 11 Hourly distribution of the occurrence of Ground Delay Programs, Ground Stops and their coincidence at LGA airport between January and August 2017

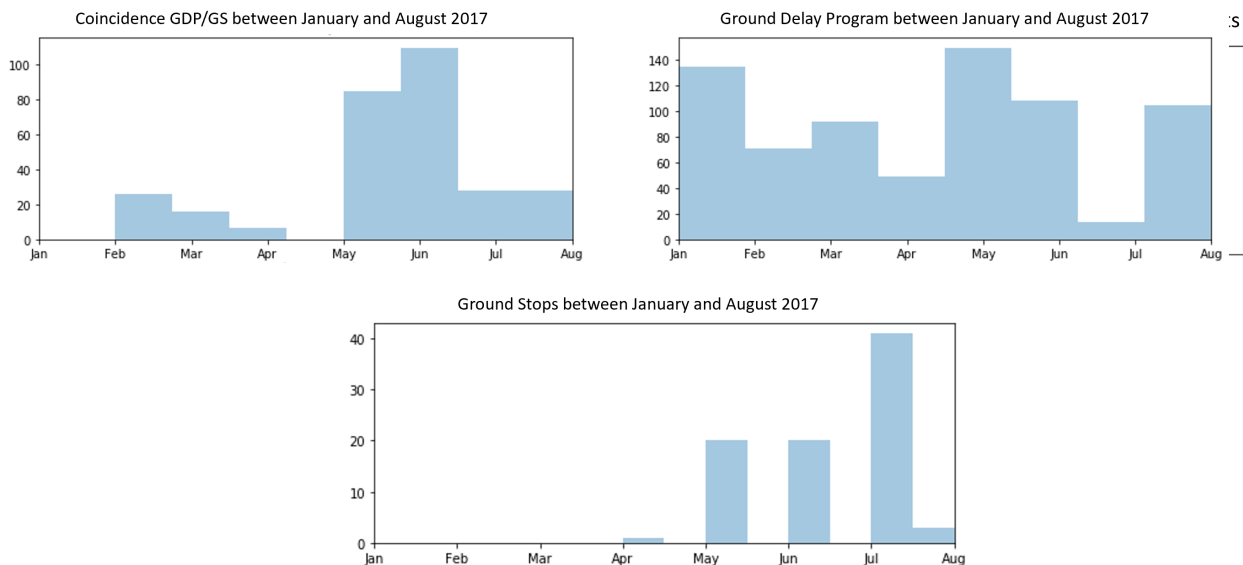


Fig. 12 Monthly distribution of the occurrence of Ground Delay Programs, Ground Stops and their coincidence at LGA airport between January and August 2017

The remainder of this section discusses results obtained for the objectives of this research:

- 1) Predict the coincidence of weather-related Ground Delay Programs and Ground Stops
- 2) Predict whether a Ground Delay Program will precede a Ground Stop, and vice versa, when coincidence occurs
- 3) Identify factors that influence the coincidence of weather-related Ground Delay Programs and Ground Stops so as to help stakeholders better understand their coincidence

A. Predicting the coincidence of weather-related Ground Delay Programs and Ground Stops

Figures 13 provides a breakdown of the distribution of classes with the imbalanced training-validation set. It also shows the distribution of the balanced set created with the SMOTE algorithm. The targets of this model are Coincidence or No coincidence.

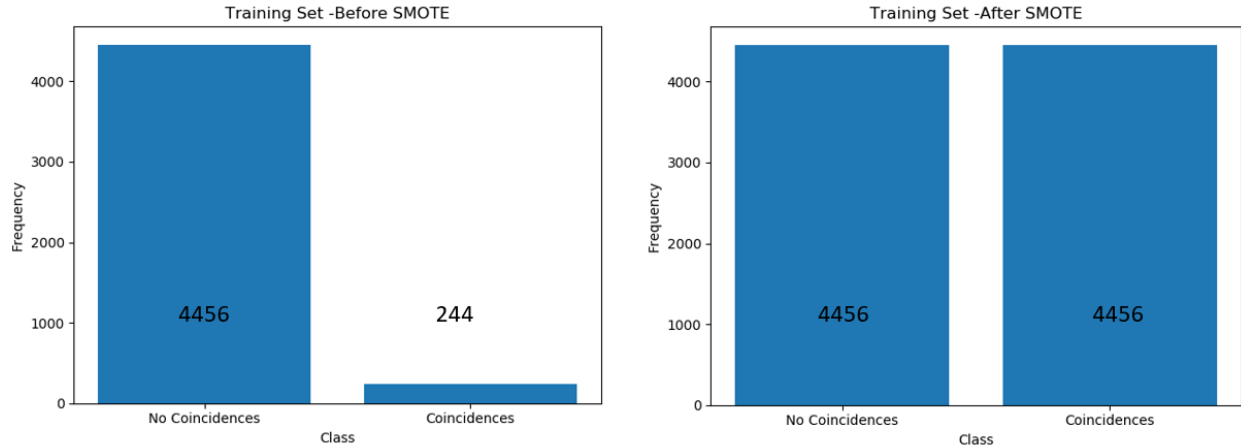


Fig. 13 Distribution of classes for predicting the coincidence of GDP and GS

Grid searches are commonly performed to tune hyperparameters in order to determine the optimal values for a given model. Thus, grid searches were performed on the training-validation set for each of the algorithms (Neural Networks, Random Forests, and Boosting Ensemble). Table 4 provides a summary of the hyperparameters of the algorithms that were tuned with 3-fold cross-validation during the grid search. It also shows the combinations of the hyperparameters that had the best performance for each algorithm. Fewer combinations of hyperparameters were performed for Neural Networks compared to the other algorithms because of the time needed to train the model. The optimal set of hyperparameters were then used to test the prediction models to identify the best suited algorithm for predicting the coincidence of weather-related Ground Delay Programs and Ground Stops.

Table 4 List of Hyperparameters

	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, 0.0001] 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

Tables 5, 6 and 7 show the confusion matrices obtained with the testing set for the Neural Network, Random Forests and Boosting Ensemble algorithms, respectively.

Table 5 Confusion Matrix for Neural Network

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

Table 6 Confusion Matrix for Random Forest

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

Table 7 Confusion Matrix for Boosting Ensemble

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

Table 8 provides a comparison of the performance of the three algorithms using the aforementioned evaluation metrics. The **Random Forest** algorithm was identified as the best suited algorithm for predicting the coincidence of Ground Delay Programs and Ground Stops based on this comparison.

Table 8 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

Figure 14 shows the ranking of predictor importance for the prediction model developed with the Random Forest algorithm. In particular, it shows that thunderstorms, low ceilings, pressure altimeter, the fourteenth hour of the day and wind direction are key predictors for this prediction model. A Partial Dependence Plot (PDP) was then used to validate the impacts that the top two key predictors have on the prediction model [34]. This was achieved by varying each predictor while keeping others constant in order to assess its impact on the model's target. The Skater library [35] was leveraged to assess the impacts of the top predictors of the best prediction model (Random Forest) on the prediction of its target (coincidence occurrence). Figure 15 shows the Partial Dependence Plots (PDP) for the two highest key predictors of the Random Forest model: thunderstorms and low ceilings. In particular, it shows that the probability of having a coincidence increased by about 9% and 4.5% whenever thunderstorms and low ceilings were present, respectively. This validates the ranking of thunderstorms as the top predictor followed by low ceilings. Partial Dependence Plots can be created for other predictors to assess their impacts on the prediction model.

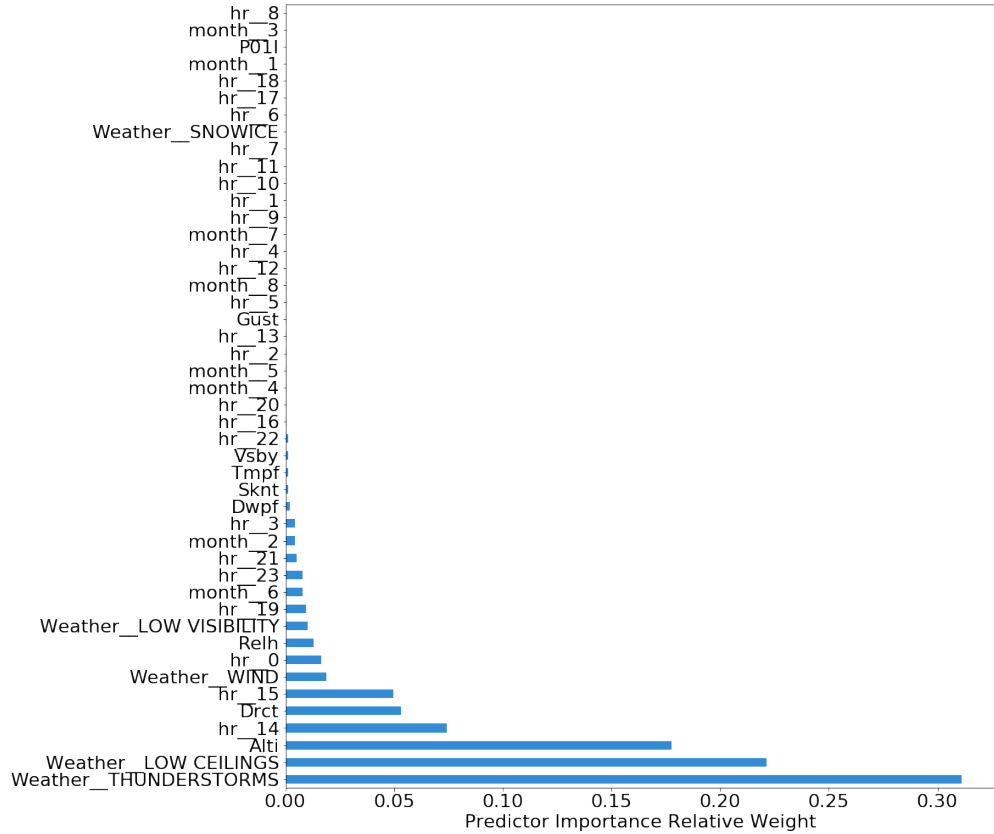
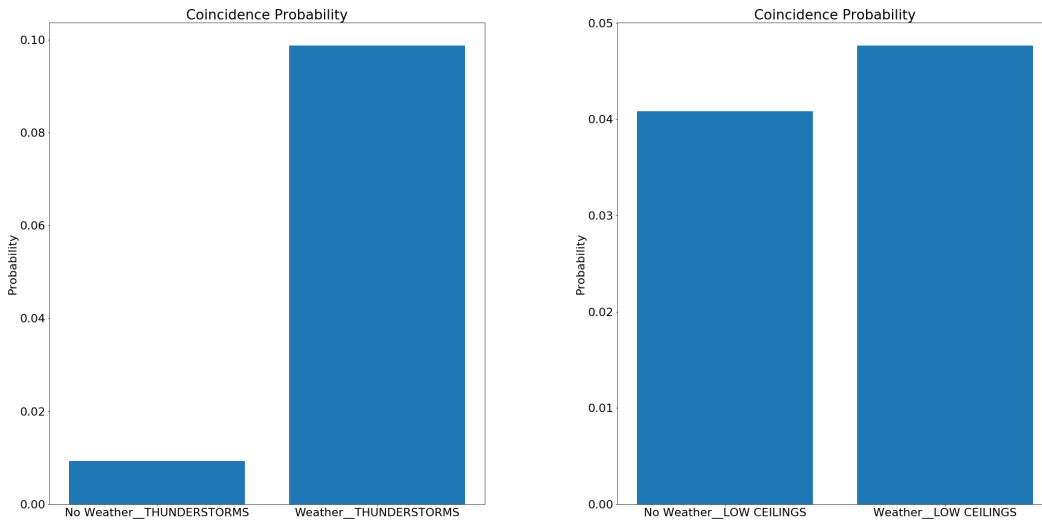


Fig. 14 Feature importance of Random Forests algorithm for predicting the coincidence of GDP and GS



(a) PDP for thunderstorms

(b) PDP for low ceilings

Fig. 15 Partial Dependence Plots for predicting the coincidence of GDP and GS

B. Predicting the precedence of weather-related Ground Delay Programs before Ground Stops, or vice versa, when coincidence occurs

Figures 16 provides a breakdown of the distribution of classes with the imbalanced training-validation set. It also shows the distribution of the balanced set created with the SMOTE algorithm. The targets of this model are No coincidence, GDP preceding GS during their coincidence, and GS preceding GDP during their coincidence.

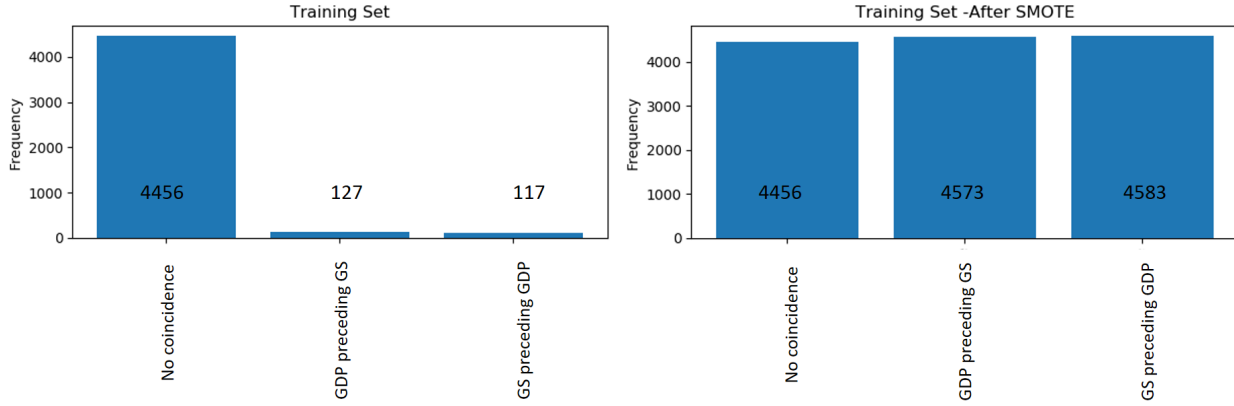


Fig. 16 Distribution of classes for predicting the precedence of GDP before GS, or vice versa, when coincidence occurs

Table 9 provides a summary of hyperparameters that were tuned, as well as the best combinations of hyperparameters for each algorithm. These combinations were then used to develop the prediction models and their performances were evaluated with the testing set.

Table 9 List of Hyperparameters for predicting the precedence of weather-related Ground Delay Programs before Ground Stops, or vice versa

	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, 0.0001] 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

Tables 10, 11 and 12 show the confusion matrices obtained with the testing set for the Neural Network, Random Forests and Boosting Ensemble algorithms, respectively.

Table 10 Confusion Matrix for Neural Network (GDP precedence of GS, and vice versa, when coincidence occurs)

	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 11 Confusion Matrix for Random Forest (GDP precedence of GS, and vice versa, when coincidence occurs)

	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 12 Confusion Matrix for Boosting Ensemble (GDP precedence of GS, and vice versa, when coincidence occurs)

	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

Table 13 provides a comparison of the performance of the three algorithms using evaluation metrics. The **Random Forest** algorithm was also identified as the best suited algorithm for predicting the whether a Ground Delay Program will precede a Ground Stop, or vice versa, when coincidence occurs.

Table 13 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

Figure 17 shows the ranking of predictor importance for the prediction model developed with the Random Forest algorithm. It shows that the presence of thunderstorms, hour of day (midnight), low ceilings, pressure altimeter and low visibility are key predictors for the model. Figure 18 show the Partial Dependence Plots (PDP) for the three highest key predictors of Random Forest model: thunderstorms, hour (midnight) and low ceilings. Figures 18a and 19 show that the likelihood of a Ground Stop preceding a Ground Delay Program when they coincide is much higher with thunderstorms and at midnight, compared to a Ground Delay Program preceding a Ground Stop. Figure 18b also shows that the likelihood of a Ground Delay Program preceding a Ground Stop when they coincide is much higher with low ceilings. Partial Dependence Plots can be created for other predictors to assess their impacts on the prediction model.

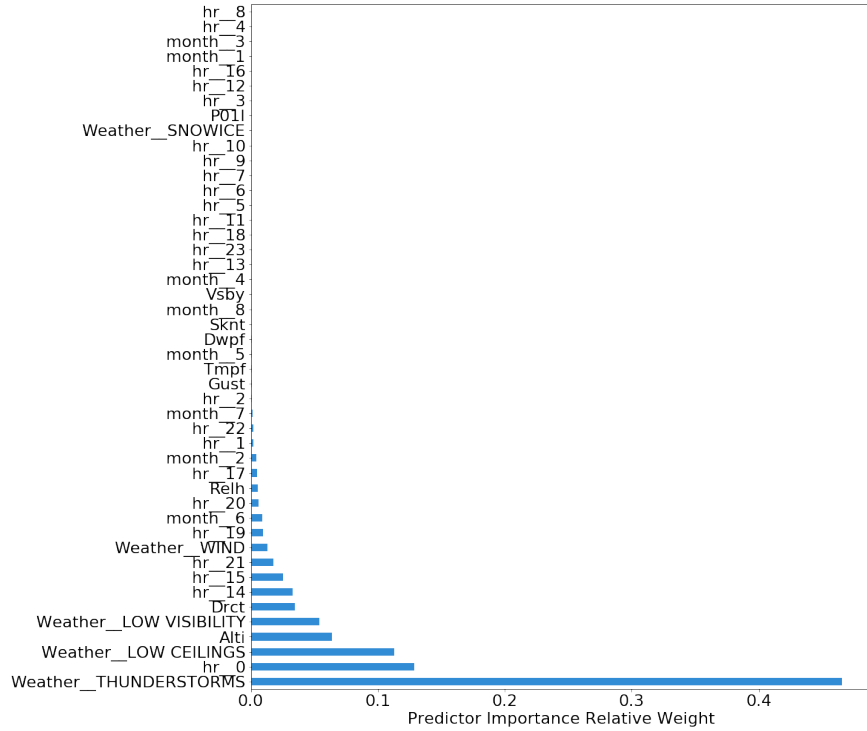
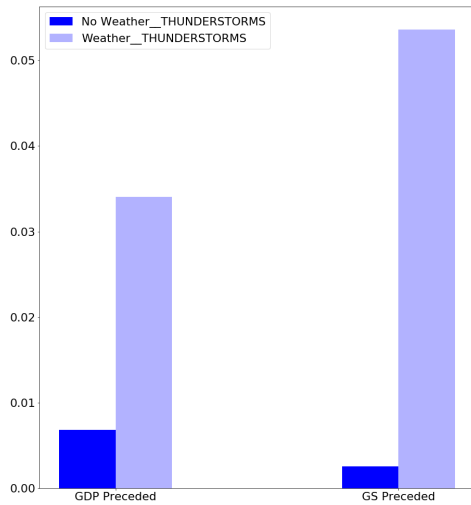
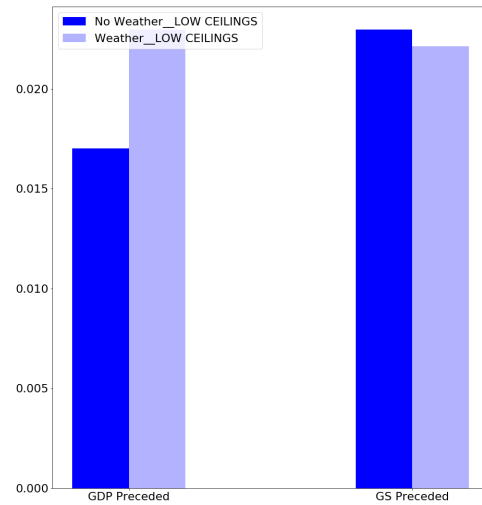


Fig. 17 Feature importance of Random Forests algorithm for predicting the whether a Ground Delay Program will precede a Ground Stop, or vice versa, when coincidence occurs



(a) PDP for thunderstorms



(b) PDP for low ceilings

Fig. 18 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

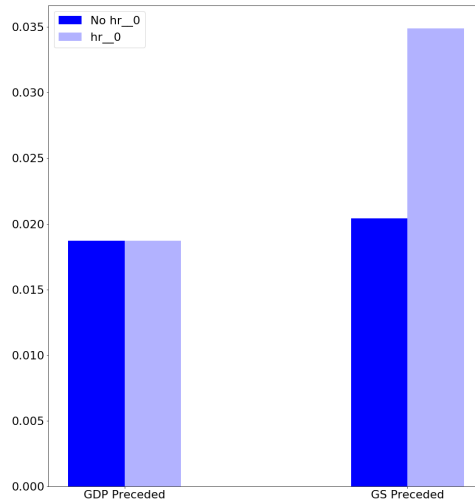


Fig. 19 Partial Dependence Plots for an hour (midnight) from Random Forest algorithm for predicting whether a GDP precedes a GS, or vice versa, when coincidence occurs

VI. 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 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. It is expected that this methodology can be repeated for other airports and across different days to help stakeholders have a better understanding of this phenomenon.

Acknowledgments

The authors wish to acknowledge the support of FAA analysts and researchers for facilitating this research. The views and findings expressed in this document are those of the authors only, and do not represent those of the FAA.

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] Xiong, J., “Revealed Preference of Airlines’ Behavior under Air Traffic Management Initiative,” Ph.D. thesis, University of California, Berkeley, 2010.
- [4] Mangortey, E., Gilleron, J., Dard, G., Pinon, O., and Mavris, D., “Development of a Data Fusion Framework to support the Analysis of Aviation Big Data,” *AIAA Science and Technology Forum (AIAA Scitech)*, 2019.
- [5] Mangortey, E., Pinon, O., Puranik, T., and Mavris, D., “Predicting The Occurrence of Weather And Volume Related Ground Delay Programs,” *AIAA AVIATION Forum*, 2019.
- [6] Dard, G., Mangortey, E., Pinon, O., and Mavris, D., “Application Of Data Fusion And Machine Learning To The Analysis Of The Relevance Of Recommended Flight Reroutes,” *AIAA AVIATION Forum*, 2019.
- [7] Mangortey, 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.
- [8] 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.
- [9] Robyn, D., *Reforming the air traffic control system to promote efficiency and reduce delays*, The Brattle Group, 2007.
- [10] Federal Aviation Administration, “Expected Departure Clearance Times,” , 2017. [http://aspmhelp.faa.gov/index.php/Expect_Departure_Clearance_Times_\(EDCT\)](http://aspmhelp.faa.gov/index.php/Expect_Departure_Clearance_Times_(EDCT)).
- [11] 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.
- [12] Hansen, M., Mukherjee, A., and Grabbe, S., “Ground Delay Program Planning under Uncertainty in Airport Capacity,” *Transportation Planning and Technology*, vol. 35, no. 6, 2012.
- [13] Wang, Y., “Analysis and prediction of weather impacted ground stop operations,” *2014 IEEE/AIAA 33rd Digital Avionics Systems Conference (DASC)*, 2014, pp. 7A2–1–7A2–14. doi:10.1109/DASC.2014.6979510.
- [14] Mangortey, E., Puranik, T., Pinon, O., and Mavris, D., “Prediction and Analysis of Ground Stops with Machine Learning,” *AIAA Science and Technology Forum (AIAA Scitech)*, 2020.
- [15] Federal Aviation Administration, *JAVA MESSAGING SERVICE DESCRIPTION DOCUMENT Traffic Flow Management Data Service (TFMData) Vol. 2.0.5*, Federal Aviation Administration, 2016.
- [16] National Weather Service, “Automated Surface Observing Systems,” , 2019. <https://www.weather.gov/asos/asostech>.
- [17] 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.
- [18] Iowa State University, “ASOS-AWOS-METAR Data Download,” , 2019. <https://mesonet.agron.iastate.edu/request/download.phtml>.
- [19] Lepori, Hubert, “Introduction to FIXM.” , 2017. URL <https://www.icao.int/MID/Documents/2017/SWIMInterregional/8.2IntroductiontoFIXM.pdf>.
- [20] Python.org, “Welcome to Python.org.” , 2018. URL www.python.org/.
- [21] Python Software Foundation, “19.7. Xml.etree.ElementTree - The ElementTree XML API,” , 2018. URL docs.python.org/2/library/xml.etree.elementtree.html.

- [22] Klein, Lawrence A, *Sensor and data fusion: a tool for information assessment and decision making*, SPIE, 2012.
- [23] Duvenaud, David K and Maclaurin, Dougal and Iparraguirre, Jorge and Bombarell, Rafael and Hirzel, Timothy and Aspuru-Guzik, Alan and Adams, Ryan P, “Convolutional Networks on Graphs for Learning Molecular Fingerprints,” *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. 2224–2232. URL <http://papers.nips.cc/paper/5954-convolutional-networks-on-graphs-for-learning-molecular-fingerprints.pdf>.
- [24] Zhang, W., Du, T., and Wang, J., “Deep Learning over Multi-field Categorical Data,” *Advances in Information Retrieval*, edited by N. Ferro, F. Crestani, M.-F. Moens, J. Mothe, F. Silvestri, G. M. Di Nunzio, C. Hauff, and G. Silvello, Springer International Publishing, Cham, 2016, pp. 45–57.
- [25] 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>.
- [26] 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.
- [27] Chollet, F., et al., “Keras,” <https://keras.io>, 2015.
- [28] Hardesty, L., “Explained: Neural Networks,” , April 2017.
- [29] 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.
- [30] Jung, H., “Adaboost for Dummies: Breaking Down the Math (and its Equations) into Simple Terms,” , April 2018. URL <https://towardsdatascience.com/adaboost-for-dummies-breaking-down-the-math-and-its-equations-into-simple-terms-87f439757dcf>.
- [31] Kunert, R., “SMOTE explained for noobs - Synthetic Minority Over-sampling TEchnique line by line,” , 2017. URL http://rikunert.com/SMOTE_explained.
- [32] McHugh, M., “Interrater reliability: the kappa statistic,” *Biochem Med (Zagreb)*. ;22(3):276–282, 2012.
- [33] 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.
- [34] Sarkar, D., 2019. URL <https://towardsdatascience.com/explainable-artificial-intelligence-part-3-hands-on-machine-learning-model-interpretation-e8ebe5afc608>.
- [35] Kramer, A., and Choudhary, P., “Model Interpretation with Skater,” , September 2018. URL <https://oracle.github.io/Skater/index.html>.