JetLagged: Predicting Flight Delays in the United States

Vanessa Eichensehr
veichens@vt.edu
Virginia Tech
USA

Nhan Hoang
nhan21@vt.edu
Virginia Tech
USA

Jackson Livanec
jlivanec@vt.edu
Virginia Tech
USA

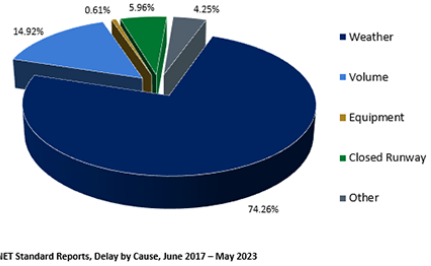
ABSTRACT

Air travel is a fundamental type of transportation that connects the entire world. As the aviation industry continues to grow, flight delays are becoming more frequent. The primary objective of this research is to develop a machine learning model to predict potential flight delays given real-time weather data. To train the model, the historical flights On-Time Performance dataset of 31 large hub airports was collected through Airline Data Inc. This service provides processed airline data sourced from the Department of Transportation. In addition, the historical weather data was collected from Meteostat, an open-source platform that stores historical meteorological data from the National Oceanic and Atmospheric Administration (NOAA). Real-time weather forecast data is accessed through the NOAA API. After processing and balancing, the flight data was modeled as a network graph, and several network centrality and temporal features were engineered. After further data exploration, multiple algorithms were evaluated, including Decision Tree, Logistic Regression, Random Forest, XGBoost, One-Class SVM, and Neural Network. Random Forest performed the best, with an accuracy and F1-score of 0.89. Random Forest and XGBoost models are utilized in an interactive dashboard that predicts flight delays in real time. This work highlights the potential for a machine learning tool that can help air traffic controllers predict delays, minimizing the disturbance it causes to customers and airlines.

1 INTRODUCTION

Currently, travel by air is the fastest and most popular mode of transportation over long distances. Whether a passenger is traveling for tourism, business, or family, air transit plays a critical role in providing a fast, safe, and reliable service for quickly moving large volumes of people. Unfortunately, air travel is subject to potential delays from air carriers due to runway issues and weather conditions. In 2023, the United States Bureau of Transportation Services reported that 20.13% of commercial flights were delayed [15]. Flight delays are sensitive to a variety of changes in weather, ranging from gusts of wind to precipitation to humidity. According to the latest update from the Federal Aviation Administration's Next Generation Air Transportation System, the largest cause of air traffic delay is weather, which accounts for roughly 74% of delays greater than 15 minutes [7]. The remaining larger delay categories include air traffic volume, equipment, and closed runways, as seen in Figure 1.

Figure 1: Causes of air traffic delay by FAA [7]



To address this issue, MITRE has been working closely with FAA to provide tools that can help air traffic controllers make informed decisions on air traffic in real-time. In 2001, MITRE developed a model to visualize how weather delays at an airport can propagate delays throughout the air system [17]. Then, in 2013, MITRE developed another tool that integrates weather data into Air Traffic Management to aid the air traffic controllers in real-time [9]. However, a large portion of the decision-making is still in the hands of humans. Although these existing tools can be helpful, they are built based on hard-coded parameters. Meanwhile, weather conditions and flight operations are dynamic and do not always fit neatly into the rule-based system. Currently, the FAA has not adopted new initiatives involving machine learning approaches. However, several researchers have already taken steps to determine whether machine learning can be helpful for predicting flight delays.

The goal of this study is two-fold

(1) To supplement the current body of work on predicting flight delays using feature engineering and the latest machine learning techniques and
(2) To integrate real-time weather data and create a dashboard where a user can enter an airport and receive delay predictions for the next 100 hours. This research seeks to enhance the decision-making process for air traffic controllers and prepare passengers before they arrive at the airport.

The remainder of this paper is organized as follows. Section 2 surveys related research. Section 3 outlines the methodology, feature engineering techniques, and algorithm analysis. Section 4 describes the experimental setup and presents the results. Section 5 presents our dashboard tool. Section 6 discusses the implications of our findings, and Section 7 concludes the paper.

2 RELATED WORK

Numerous studies have explored machine learning techniques for predicting flight delays. Published in 2016, a study by Choi et al.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

predicted weather-induced airline delays using machine learning algorithms [4]. The researchers combined US domestic flight data for the 45 largest airports and weather data from 2005 to 2015. Then, they used the SMOTE sampling technique to generate data to balance their dataset synthetically. They used the following features: Quarter of the Year, Month, Day of Week, Departure Time, Arrival Time, Wind direction angle, Wind speed rate, visibility, precipitation, snow depth, snow accumulation, and five other coded descriptors for the weather conditions. After normalizing their data, the authors then trained and tested four machine learning models: Decision Trees (77.92% accurate), Random Forest (81.37% accurate), AdaBoost (83.21% accurate), and KNN (61.69% accurate) [4].

Published in 2024, a study by Ajayi et al. used network centrality measures to enhance flight delay predictions from machine learning models [1]. The researchers first constructed a network model in which airports served as nodes and flight routes as edges. The three network centrality measures at each node are degree centrality, betweenness centrality, and closeness centrality. Once the scores were computed for all airports, the dataset was augmented by adding the network centrality measures for the departure and arrival airports for a total of six additional features. The researchers tested and trained three models: Random Forest (86.2% accurate), Gradient Boosting (85.8% accurate), and CatBoost (85.6% accurate) [1].

Pamplona et al. published research in 2018 predicting air traffic delays between Sao Paulo and Rio de Janeiro with a supervised neural network [16]. Aiming to predict delays only in January 2017, the authors built an artificial neural network with four hidden layers that had an average accuracy of 91.3%. Their model suggested that the day of the week, block hour, and route have a great influence on the flight delay but failed to consider meteorological factors [16].

In 2022, Shao et al. predicted flight delays with spatio-temporal trajectory convolutional network and airport situational maps [18]. This vision-based solution was intended to achieve high forecasting accuracy for just one airport, Los Angeles International Airport. The authors showcase end-to-end deep learning architecture, known as TrajCNN, to capture spatial and temporal information from the situational awareness map, which shows statuses from airports across the country. Though it only makes predictions for one airport, the proposed framework obtained good results (around 18 minutes mean error) for predicting flight departure delay [18].

3 METHODOLOGY

In this study, a delayed flight is defined as a flight where the actual departure time is greater than 15 minutes than the scheduled departure time. This is the industry standard definition of a flight delay. This definition does not include canceled flights. Although this distinction reduces the size of the training dataset, this trade-off is necessary to balance long-term model accuracy.

3.1 Feature Engineering

Flight data was collected and preprocessed as described in Section 4.1. The flight dataset was enriched with several engineered fields that fall into three main categories: weather, network, and temporal features. The resulting dataset includes the following eight weather features that overlapped from the historical dataset and real-time weather API:

- Elevation
- Temperature
- Dew Point
- Relative Humidity
- Chance of Precipitation
- Wind Direction
- Wind Speed
- Wind Gust Speed

Firstly, elevation was selected because it impacts the severity of temperature, pressure, and precipitation. For example, high-elevation airports are prone to greater snow accumulation, which may cause delays. Additionally, high-elevation airports, when in combination with high temperatures, lower the air density, which can decrease aircraft performance. Specifically, decreasing the climb rates and increasing takeoff distance. Temperature was selected because it influences atmospheric density and is closely tied to other meteorological factors such as wind patterns and precipitation [3, 5].

Next, the dew point was selected because it is an indicator of the effects of humidity and can lead to changes in density altitudes. Low dew points suggest dry conditions that can enable efficient flight operations during takeoff, while high dew points are associated with increased density altitude and lower air density [5].

Relative humidity was selected because it is closely coupled with precipitation potential. Precipitation chance predicts the likelihood of rain, snow, or sleet directly disrupting flights. Wind direction is important in relation to runway direction because crosswinds can deem a runway unsafe for takeoff. Wind speed and gust speed inform aircraft stability, turbulence, and the feasibility of takeoff and landing. Airports will often employ operational limits for wind speeds [11].

The various weather data points are all meteorologically interconnected, allowing missing data to be back-filled using industry-standard conversions [13].

After the weather data was collected, the data was plotted as a network graph, with airports as nodes and flights as edges [Figure 2]. Using the network graph, the degree centrality, betweenness centrality, and closeness were studied. Degree centrality measures the number of direct connections a node (e.g., airport) has in a network. For an unweighted graph, it is defined as:

$$C_D(v) = \frac{\deg(v)}{N-1}$$

where $\deg(v)$ is the degree of node v, and N is the total number of nodes in the network. In this research, the total degree was used, but future work could consider indegree and outdegree separately. Airports with a high degree of centrality act as regional hubs with many direct connections. As a result, delays are spread to the other connected airports.

Betweenness centrality quantifies the number of times a node acts as a bridge along the shortest flight path between two other nodes. It is defined as:

$$C_B(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

Figure 2: Network of US Domestic Flights 2023

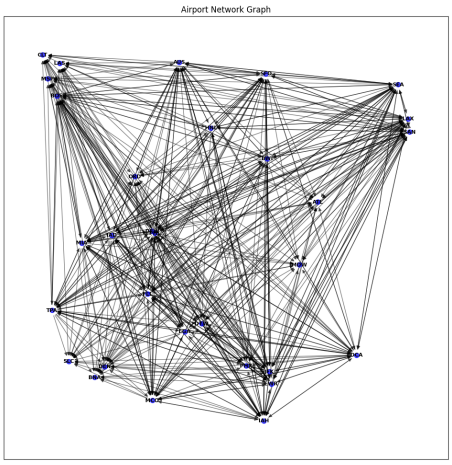
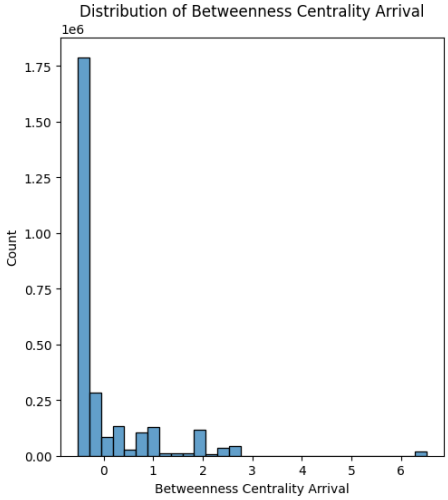


Figure 3: Distribution of Betweenness Centrality Arrivals



where σ_{st} is the total number of shortest paths between nodes s and t, and $\sigma_{st}(v)$ is the number of those paths that pass through node v. Nodes with high betweenness centrality are critical for maintaining flow within the network. The reason is these airports often serve as transit points, making them prone to cascading delays if disrupted. Hence, it can create extreme inconvenience for travel between regions depending on this bridge.

Closeness centrality measures how close a node is to all other nodes in the network. It is defined as:

$$C_C(v) = \frac{N-1}{\sum_{u \neq v} d(v,u)}$$

where $d(v,u)$ is the shortest distance between nodes v and u. Airports with high closeness centrality are well-positioned for efficient travel across the network. Such airports are likely to recover more quickly from logistical disruptions or delays.

In addition to the weather features, the three network centrality measures for the departing airport and arrival airport were appended to the dataset. Some temporal features such as Quarter, Day of Week, Month, and Hour were also added. These categorical features were one-hot encoded to avoid assigning weight to categorical data.

Additionally, temporal features relating to peak travel season were also included. As stated by the OAG, an international travel data provider, August is consistently one of the busiest months for air travel. Specifically, for the first and second weekends of August [14]. A report by Deloitte also noted Americans have peak travel plans during the holiday season of Thanksgiving and Christmas [8]. However, these additional features did not improve model performance, likely due to the strong co-linearity among existing and engineered temporal features described above. Although there is much more travel during certain periods, airlines must also be equipped and prepared to handle seasonality. Therefore, this does not play a large role in determining the likelihood of delay.

3.2 Exploratory Data Analysis

After feature engineering, the exploratory data analysis was performed. Originally, at the 31 major airports in the dataset, 637,232 flights were delayed by 15 minutes or more, while 2,180,679 flights were not delayed. The data was cleaned, balanced, and normalized according to Sections 4.1 and 4.2. After balancing, 1,409,641 flights were delayed, and 1,408,270 flights were on time. The balanced and normalized dataset was used for the following analyses. A total of 2,817,911 data points and 36 features were explored.

First, univariate analysis was performed by plotting the distribution of each of the features in the dataset on its own. Notably, in plotting the distributions of betweenness and degree centrality for both departures and arrivals, a power law/Pareto effect can be observed, where many airports have low betweenness centrality and a few have high betweenness centrality as seen in Figure 3. Also, we observed that more flights occur during the summer months and during the winter holiday season.

Next, bivariate analysis was performed to check the relationships between pairs of features and the target variable. A correlation heat map was created with each of the 36 features and the delay status. The Departure Hour feature had the strongest correlation with Delay at 0.23, followed by Arrival Hour Bucket (0.16), Temperature (0.11), Dew Point (0.10), Precipitation Chance (0.09) and Month 7 (0.08). As expected, higher temperatures were correlated with July and August. Departure and Arrival Hour were highly correlated, and several of the meteorological features were highly correlated, such as wind speed and gusts, temperature, and dew point. The correlation heat map of only the weather features and departure delay is shown in Figure 4.

After creating the correlation matrices, a Random Forest Feature Importance analysis was conducted. The technique identifies the most influential features in a dataset for predicting a target variable (e.g., Departure Delay). It quantifies the contribution of each feature to the predictive power of the model. These feature importance values are then plotted in a bar chart in Figure **??**. The top five most important features are as follows: Arrival Hour Bucket, Closeness Centrality Arrival, Degree Centrality Arrival, Departure Hour Bucket, and Dew Point. In addition to observing each feature individually, the features were also bucketed into their three categories

Figure 4: Correlation Heat Map of Weather Variables and Departure Delay

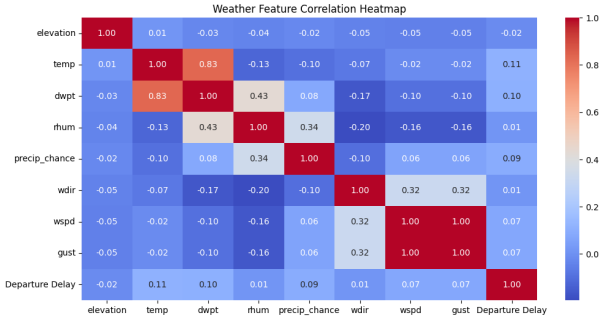


Figure 5: Feature Importance (Random Forest)

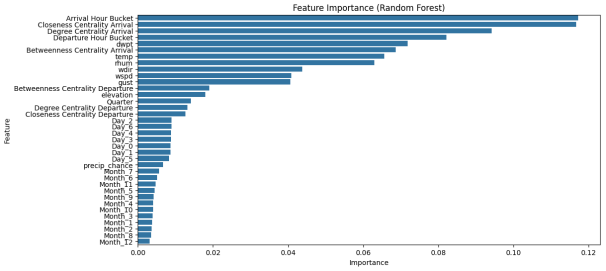
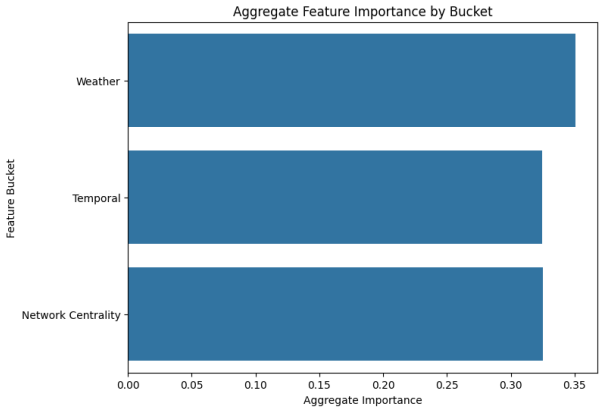


Figure 6: Aggregated Feature Importance



(Network Centrality, Temporal, and Weather) and aggregated to achieve feature importance by bucket in Figure **??**. The combined weather features prove slightly more important than the temporal and network centrality measures. However, each of the three aggregated importance values is similar.

Finally, the relationship between each feature was plotted with respect to our target variable, Departure Delay > 15 Min. The distributions of each of the features were plotted as histograms and box plots with respect to delay status. The most important temporal feature, the Arrival Hour Bucket, is seen in Figure 7. Arrivals that occur later in the day are more likely to be delayed, while arrivals

Figure 7: Distribution of Arrival Hour (Delay vs. Non-delay)

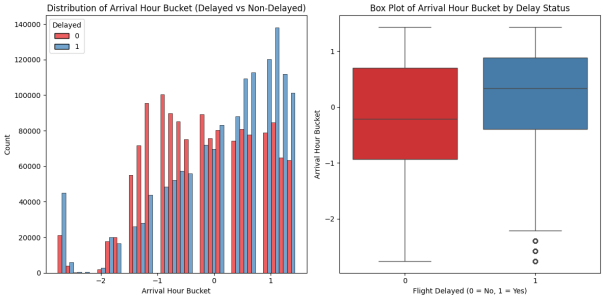
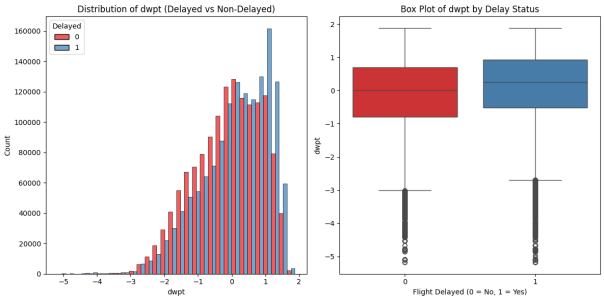


Figure 8: Distribution of Dew Point (Delay vs. Non-delay)



scheduled for the morning are less likely to be delayed. The most important meteorological feature, Dew Point, is shown in Figure 8. Higher dew points seem to cause more delays than lower dew points.

3.3 Feature Selection

During feature selection, we aimed to retain as many features as possible to make use of our extensive dataset. The correlation heat map analysis from Figure 4 shows that every weather feature exhibits a correlation coefficient less than or equal to 0.11. Unfortunately, the correlation coefficient can only measure linear relationships, but a combination of these features may have a significant implicit relationship with flight delay. For example, determining a flight delay status solely based on rain is difficult, yet rain combined with other factors such as wind gusts or congestion at the arrival airport may cause a delay.

As shown in Figures 5–6, the resulted model showed that all features contributed some level of importance. While tuning models, many features were randomly removed, but it also caused lower performance. While matching features in historical and real-time weather data, some helpful weather features from the historical dataset were removed due to the limitation of NOAA API. Because of the large dataset size, overfitting was less of a concern. Due to all of these reasons, we kept all remaining features to allow the models to identify patterns and relationships that did not exist through the simple correlation analysis.

3.4 Algorithm Analysis

The problem is framed as a binary classification task to predict whether a flight will be delayed based on the weather features by the hour. The Decision Tree was selected as the base model to demonstrate a basic classification, as it can capture the basic non-linear relationship. Logistic Regression was also added as an initial model due to its ability to provide probabilistic predictions and its strength in binary classification.

After comparing the statistics, the tree-based approach seems to be a more appropriate fit for the dataset. Hence, this led to the decision to experiment with a bagging solution, Random Forest, to improve the model's accuracy and possibly reduce the overfitting due to the imbalanced dataset between on-time flights and delayed flights, as Random Forest averages out all the predictions from multiple trees. At this stage, XGBoost was also added to the collection due to its computational efficiency. The downside to XGBoost was that it requires extensive hyperparameter tuning for the model to perform optimally.

Due to the nature of the skewed dataset, a one-class Support Vector Machine (SVM) model was also investigated because of its strength in anomaly detection. The delay class was treated as an anomaly when compared to those flights that were on time. Finally, a Neural Network was also implemented with the ambition of modeling the hierarchical relationships between the features. Through the selections of algorithms described above, our main goal was to find a well-balanced model that can interpret the complex relationships between the weather features and possible flight delays.

As the focus is on classification tasks, performance metrics like accuracy, precision, recall, and F1-score were used to evaluate the models. In an imbalanced dataset, accuracy can be a deceptive metric as it disproportionately reflects the performance of the majority class. Meanwhile, when comparing the model's performance in minimizing misclassification of delayed flights versus over-predicting delay, we consider the ability to limit misclassified delayed flights as more important to efficiently control airport operations. Thus, recall is a crucial metric for this particular model evaluation application because it focuses on improving the model to not be afraid to predict delays [19]. F1-score was also prioritized because it quantifies the trade-off between misclassifying and over-predicting delays. At the end of the training, a well-calibrated model will have similar values for recall, precision, and F1 score. A significant disparity between these metrics could indicate issues such as overfitting (high precision, low recall) or underfitting (low scores across all metrics).

3.5 Model Selection

Building on the algorithm analysis, all six of the machine learning algorithms above were evaluated for performance and computational efficiency. Two algorithms, SVM and Neural Network, were eliminated due to their high computation cost. With such a large dataset and limited computation resources, both models would take hours to run and even longer to tune. The Neural Network model was originally added due to its ability to model complex patterns and popularity within the industry. However, its performance was poor and proved difficult to tune. Hence, SVM and Neural Network are excluded from further results discussion. The final models selected were Decision Tree, Logistic Regression, Random Forest, and XGBoost.

4 EXPERIMENTS

4.1 Data

The experiment is based on three primary datasets. First is the historical flight On-Time Performance data from Airline Data Inc filtering only 31 large hub airports from the year 2023 [10]. Secondly, the historical weather data comes from Meteostat's Python package based on the coordinates and requested hours [6]. Thirdly, the International Air Transport Association (IATA) airport information was compiled by Bilz using Travel Hacking Tool, an aviation data API provider [2]. Lastly, a NOAA API was connected through a Python library to retrieve the live weather data for up to 100 hours ahead [12].

To prepare the data for analysis, the On-Time Performance dataset was merged with the airport information to enrich the dataset with the latitude and longitude coordinates, elevation, and time zone. Each airport's timezone was considered, and each flight's scheduled departure and arrival times were converted to UTC in order to retrieve the historical and real-time weather data. Finally, all of the flight schedules were merged with the corresponding floored hour bin and the latitude and longitude coordinates of the airport from the past weather dataset.

Next, data cleaning included unit conversions, filling missing values with estimates based on other meteorological parameters, and bucketing wind direction into eight categorical cardinal directions (e.g. North, South, etc.). This bucketing assisted with the one-hot encoding of the input vector. Flights were evaluated based on departure tardiness and flagged as delayed or on time. Canceled flights were scrubbed out of the dataset entirely. All rows without temperature data were dropped, and other weather data points not available in the historical weather data were back-filled using industry standard estimation formulae based on the other weather features.

The data available via NOAA forecast contains different information than the historical weather reports; therefore, some feature engineering was necessary. For example, precipitation is a key factor when determining the timeliness of flight departures, but it is reported in absolute terms in the historical data as a volumetric measure of precipitation. However, precipitation is measured in probability for forecast data. For this reason, the historical data precipitation probability is set to 100 if it is nonzero and 0 otherwise.

After handling missing data, there were approximately 2.8 million records for 2023, with 2.2 million flights running on time and 637K flights delayed. The extreme imbalance in the prediction class initially led to poor model performance, so all delayed flight rows were kept, and an equally sized group of on-time flights were randomly sampled and shuffled into the data, resulting in a training dataset containing 1.27 million rows of flights, with an equal number of delayed and on-time flights.

4.2 Experimental Setup

The data was scaled, encoded, and randomly split into 80% training and 20% validation data. Four models were constructed to predict

the presence of a flight delay: Decision Tree, Logistic Regression, Random Forest, and XGBoost.

The real-time weather forecast is transformed using the same scaler and fed as a scaled input vector to the chosen model to produce a prediction window of potential flight delays.

The vast majority of flights are either on-time or fall within a non-delay window, making the predicted feature highly imbalanced. This necessitated a more sophisticated data preparation approach. After scaling and splitting the dataset into training and testing sets, three processing methods were applied:

(1) **As-is**: The dataset was used without adjustments. While this yielded high overall accuracy, the models were flawed; they consistently misclassified most delays to maximize accuracy. The confusion matrix revealed a bias toward false negatives, with very few delays correctly identified.
(2) **50-50 Resampling**: All delayed samples were retained, and an equal number of non-delayed samples were randomly selected and shuffled in. This balanced the classes but reduced the training data by approximately two-thirds. This approach improved model performance significantly.
(3) **SMOTE (Synthetic Minority Oversampling Technique)**: SMOTE generates synthetic samples for the minority class by interpolating between existing minority samples and their nearest neighbors in feature space. This method effectively increased the representation of delayed flights without reducing the majority class size, preserving the overall dataset size. By addressing the class imbalance, SMOTE allowed models to better capture patterns related to delays, resulting in more balanced confusion matrices and greatly improved detection of delayed flights. To ensure fairness in model comparison, the models trained on SMOTE then predicted the un-interpolated scaled dataset to verify the performance metrics.

Table 1: Data Preparation Performance (Random Forest)

Preparation Method	Accuracy	F1-Score
As-is	0.79	0.34
50-50	0.72	0.73
SMOTE	0.89	0.89

Table 1 displays the accuracy of the three data balancing methods for a Random Forest model. The hyperparameters remained consistent across each balancing technique. It is clear that the imbalance yields poor model performance. While the 50-50 split technique improved model performance, the SMOTE technique proved to be the best option.

4.3 Results

The performance of the various models is depicted in Table 2. Among the models explored, the Random Forest classifier achieved the best overall performance, demonstrating high accuracy, precision, recall, and F1 score. This strong performance can be attributed to the ensemble nature of Random Forest, which combines multiple decision trees to reduce overfitting and improve generalization. The class imbalance posed a major challenge for this model, but adjusting the input data, feature selection, and hyperparameters yielded a high-performing model.

Table 2: Model Performance

Model	Accuracy	Precision	Recall	F1 Score
Decision Tree	0.87	0.87	0.88	0.88
Logistic Regression	0.63	0.63	0.63	0.63
Random Forest	0.89	0.89	0.90	0.89
XGBoost	0.78	0.86	0.67	0.75

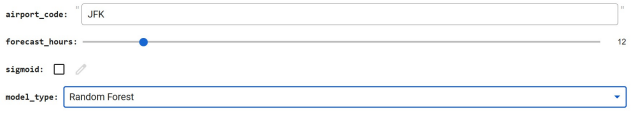
Besides Random Forest, other models exhibited varying degrees of success. The Decision Tree model performed relatively well, benefiting from its simplicity, but it lacked the robustness of the ensemble approach provided by Random Forest. Logistic Regression struggled, likely due to its linear nature, which made it less effective at capturing the nonlinear relationships present in the dataset. This model was only chosen because the prediction class was binary. XGBoost, while not achieving the top performance in this experiment, showed promise due to its probabilistic output, which does not automatically threshold predictions to 0 or 1. This makes it particularly advantageous for integration into the dashboard where nuanced decision-making or custom thresholds might be required.

The final model selection leveraged the Random Forest classifier with tuned hyperparameters, specifically `n_estimators = 50`, which provided a balance between computational efficiency and predictive performance. This combination of hyperparameters and the inherent strengths of Random Forest resulted in a reliable and robust model for predicting flight delays. This model, as well as XGBoost, were saved and used to predict flight delays at a given airport for flight forecasts.

5 DASHBOARD

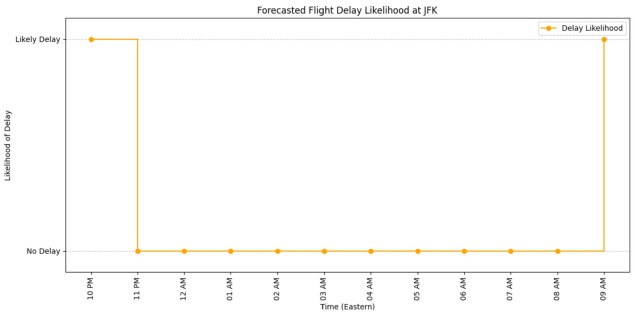
After training and testing our models, we exported them for use in an interactive dashboard. The dashboard displays the delay probability over time for a forecast window at a given airport. As shown in Figure 9, a user can choose an airport, number of forecast hours, and model type. If the model type is XGBoost, the log-odds are transformed into probabilities when the sigmoid checkbox is checked. In a random forest model, the probabilities of delay are 0 or 1.

Figure 9: User Controls for Dashboard



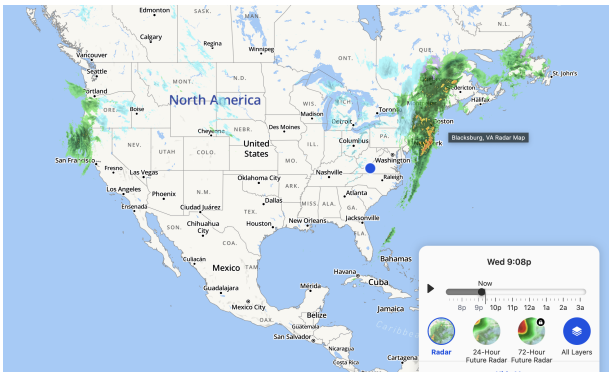
Based on the Random Forest model, shown in Figure 10 is the graph indicating the likelihood of weather delays at JFK for the next eleven hours. The dashboard shows likely delays until 11 pm, which makes sense because New York was experiencing delays due to weather at the time (see weather map in Figure 11, taken at the same time the forecast was generated). The weather forecast showed the Northeastern region experiencing extreme precipitation.

Figure 10: Forecasted Flight Delay Likelihood at JFK



A limitation of this dashboard is that it does not incorporate some temporal features like scheduled arrival time because it only considers the departing airport provided by the user. Crucially, even though the training data is based on large hub airports, the dashboard is designed to take input from any airport within the US. This generalization comes with the limitation that we do not have network centrality metrics for every airport. As a result, the current dashboard does not fully capture connectivity between airports as well as the training models. Nonetheless, the user interface provides users the ability to tailor the predictions to their scenarios. By integrating the selected models with real-time forecasting, the dashboard provides a way of easily utilizing the underlying model. The accuracy of the dashboard model without network centrality and temporal features is 63%.

Figure 11: US Weather Map at Time Forecasted



6 DISCUSSION

Model performance based on weather, network centrality, and temporal data exceeded initial expectations. Random Forest achieved an accuracy and F1 score of 0.89, and Decision Tree achieved an accuracy of 0.87 and an F-1 of 0.88. Both outperformed other models like Logistic Regression and XGBoost due to their ability to capture complex relationships while reducing overfitting through ensemble learning. These results are particularly impressive because only 74% of delays are caused by the weather [7].

There are no clear patterns in flights that were misclassified with higher frequency; therefore, model ensembling beyond the implicit ensembling within the random forest model was not pursued to enrich future flight delay predictions.

The developed dashboard enables air traffic controllers, airlines, and passengers to make informed decisions based on real-time predictions. Although the training data was an equal proportion of delayed and non-delayed flights, the model output properly classifies delays conservatively, which is consistent with the relatively low volume of overall delayed flights. By integrating machine learning with weather forecasting, this tool can reduce disruptions and enhance travel experiences. However, users should remain cautious about overreliance on predictions, particularly for scenarios involving non-weather delays or airports with limited data representation.

Despite its strengths, the model has limitations. The lack of network centrality metrics for smaller regional airports affects prediction accuracy for less connected nodes. Real-time air traffic data and airline-specific operational factors are not yet included either. Ethical considerations, including biases in historical data and privacy concerns, must also be addressed before dashboard deployment.

7 CONCLUSION

In this study, we explored the applications of machine learning and network centrality techniques to predict flight delays by integrating flight and weather data. By engineering features across three categories of weather, temporal, and network centrality, we validated how necessary it is to combine domain context with machine learning. Only when doing so, our models can enhance their performance and accuracy.

The Random Forest classifier emerged as the most robust model, achieving an F1-score of 0.89, with balanced performance across accuracy, precision, and recall. This highlights the capability of the bagging ensemble method to capture complex relationships and also reduce overfitting and variance. Additionally, techniques such as SMOTE, which generates synthetic samples, can be effective in addressing the problem of imbalanced dataset in classification tasks. This led to the improvement of our model's ability to identify delayed flights.

Our analysis uncovered many key factors that influenced flight delays, including some unexpected factors. Surprisingly, temporal factors like Arrival Hour Bucket play a significant role in predicting delays, with afternoon/evening arrivals more prone to delays possibly due to the backlog occurring throughout the day. Furthermore, the Dew Point weather feature was important, which shows the impact of air density on flight operations. Lastly, network centrality metrics like Closeness Centrality provided additional context about the role of airport connectivity in delays.

The result of this research is a user-friendly dashboard that leverages the trained Random Forest and XGBoost models to provide real-time predictions of flight delays based on weather forecasts. This tool has the potential to assist air traffic controllers and passengers in making informed decisions, limiting disruptions, and improving overall travel experiences.

Future work can build upon these findings by incorporating additional dynamic variables, such as live air traffic data or airport-specific operational factors, to further enhance predictive capabilities. Moreover, the dashboard can be updated to include network

centrality by expanding the dataset to include smaller regional airports and international routes. This may possibly lead to a more generalized model.

By combining machine learning with domain knowledge, this study aimed at improving operational efficiency and passenger satisfaction in the aviation industry.

8 ACKNOWLEDGEMENTS

We would like to thank Dipasis Bhadra, a senior quantitative economist from the Federal Aviation Administration, for working with us to retrieve the dataset from Airline Data Inc. and providing great internal insights that initiated us to the research topic.

9 AUTHOR CONTRIBUTIONS

The team all contributed to a fair share of collaboration. Nhan conducted preliminary project research and collaborated with domain expert to identify key features to retrieve the necessary datasets from the database. She also sourced additional datasets and API, such as historical weather, airport information, and NOAA forecasts. Then, the data preprocessing of joining the flights and weather data was performed into a starter dataset for modeling and exploratory analysis. Vanessa conducted literature reviews for SMOTE and imbalanced data techniques and created the initial models. Next, she performed network centrality and temporal feature engineering. In addition, she conducted exploratory data analysis both on weather features and on all engineered features. Jackson contributed to the project by performing weather and holiday feature engineering, preparing data post-processing, retrieving forecasts from NOAA, and implementing unit conversion functions to standardized inputs from forecasting features to existing features. Additionally, Jackson created, tuned, and analyzed the models to ensure optimal performance. Later on, Vanessa's findings were implemented into Jackson's model to fine-tune its performance. Finally, Jackson created a dashboard and charts to allow user interactions. We all contributed to the paper equally.

10 DATA AND CODE AVAILABILITY

The dataset and code are available through the link below, which consists of of the data folder, including the raw data and processed data outputs. It also contained all the preliminary and preprocessed code files that went into the model along with documentation. https://drive.google.com/drive/folders/1XGWIiMgocHnJVsfm907OHzJJG8Rs_WI8?usp=sharing

The buttoned-up models and front end are available in this dashboard file: https://colab.research.google.com/drive/1IBm1qJXOnbrqPRLAyyTK3MOMuhoPhD8C?usp=sharing

REFERENCES

[1] Joseph Ajayi, Yao Xu, Lixin Li, and Kai Wang. 2024. Enhancing Flight Delay Predictions Using Network Centrality Measures. *Information* 15, 9 (2024), 559.

[2] Alexander Bilz. 2021. Airports: A Complete List of IATA Airports. https://github.com/lxndrblz/Airports GitHub repository. Accessed: 2024-10-27.

[3] Stefan Borsky and Christian Unterberger. 2019. Bad weather and flight delays: The impact of sudden and slow onset weather events. *Economics of Transportation* 18 (03 2019), 10–26. https://doi.org/10.1016/j.ecotra.2019.02.002

[4] Sun Choi, Young Jin Kim, Simon Briceno, and Dimitri Mavris. 2016. *Prediction of weather-induced airline delays based on machine learning algorithms.* 1–6 pages. https://doi.org/10.1109/DASC.2016.7777956

[5] Ryan Conrick, Matthew Nemunaitis, and Ian L. Jirak. 2023. Density Altitude: Climatology of Daily Maximum Values and Evaluation of Approximations for General Aviation. *Weather, Climate, and Society* 15, 4 (2023), 917–932. https://doi.org/10.1175/WCAS-D-22-0026.1

[6] Meteostat Developers. 2024. Meteostat Python Library. https://github.com/meteostat/meteostat-python GitHub repository. Accessed: 2024-10-27.

[7] Federal Aviation Administration. 2024. FAQ: Weather Delay. https://www.faa.gov/nextgen/programs/weather/faq. Accessed: 2024-12-11.

[8] Kate Ferrara, Eileen Crowley, Matthew Usdin, Matt Soderberg, Maggie Rauch, and Upasana Naik. 2024. 2024 Deloitte holiday travel survey: More time away from home this holiday season, as travel enthusiasm escalates. Deloitte Consumer Industry Center. Retrieved November 12, 2024, from https://www2.deloitte.com/us/en/insights/industry/retail-distribution/holiday-travel-survey.html.

[9] B. Flathers, M. Fronzak, M. Huberdeau, C. McKnight, M. Wang, and G. Wilhelm. 2013. *A Framework for the Development of the ATM-Weather Integration Concept.* Technical Report. The MITRE Corporation, McLean, VA.

[10] Airline Data Inc. 2023. On-Time Performance of 2023 Flights Dataset. https://www.airlinedata.com/ Compiled dataset provided by Airline Data Inc. using U.S. Department of Transportation data. Accessed via subscription.

[11] Gloria Kulesa. 2003. Weather and Aviation: How Does Weather Affect the Safety and Operations of Airports and Aviation, and How Does FAA Work to Manage Weather-Related Effects? *The Potential Impacts of Climate Change on Transportation* (2003).

[12] Paulo Kuong. 2022. NOAA Python SDK. https://github.com/paulokuong/noaa GitHub repository. Accessed: 2024-10-28.

[13] Mark G. Lawrence. 2005. The Relationship between Relative Humidity and the Dewpoint Temperature in Moist Air: A Simple Conversion and Applications. *Bulletin of the American Meteorological Society* 86, 2 (2005), 225–234. https://doi.org/10.1175/BAMS-86-2-225

[14] OAG. 2024. The Busiest Days for Air Travel, 2009-2024. *Aviation Market Analysis Blog* (August 2024). https://www.oag.com/blog/busiest-days-for-air-travel-2009-2024 Accessed: 2024-12-13.

[15] Bureau of Transportation Statistics. 2024. Causes of Flight Delays. https://www.transtats.bts.gov/ot_delay/ot_delaycause1.asp?6B2r=E&20=E Accessed: 2024-12-13.

[16] Daniel Alberto Pamplona, Li Weigang, Alexandre Gomes de Barros, Elcio Hideiti Shiguemori, and Claudio Jorge Pinto Alves. 2018. Supervised Neural Network with multilevel input layers for predicting of air traffic delays. In *2018 International Joint Conference on Neural Networks (IJCNN).* 1–6. https://doi.org/10.1109/IJCNN.2018.8489511

[17] L. Schaefer and D. Millner. 2001. Flight delay propagation analysis with the detailed policy assessment tool. In *Proceedings of the IEEE Systems, Man, and Cybernetics Conference* (Tucson, AZ, USA). 1232–1236. https://doi.org/10.1109/ICSMC.2001.973100

[18] Wei Shao, Arian Prabowo, Sichen Zhao, Piotr Koniusz, and Flora D. Salim. 2022. Predicting flight delay with spatio-temporal trajectory convolutional network and airport situational awareness map. *Neurocomputing* 472 (2022), 280–293. https://doi.org/10.1016/j.neucom.2021.04.136

[19] Philipp Thölke, Yorguin-Jose Mantilla-Ramos, Hamza Abdelhedi, et al. 2023. Class imbalance should not throw you off balance: Choosing the right classifiers and performance metrics for brain decoding with imbalanced data. *NeuroImage* 277 (2023), 120253. https://doi.org/10.1016/j.neuroimage.2023.120253