Predicting Flight Delay Length: A Machine Learning Approach

Gian Karl Colinares
Bachelor of Science in Computer Science
National University - Manila
Metro Manila, Philippines
colinaresgl@students.nationa;-u.edu.ph

Dhan Micheal Tamparong
Bachelor of Science in Computer Science
National University - Manila
Metro Manila, Philippines
tamparongdl@students.national-u.edu.ph

***Abstract*—Flight delays significantly impact the aviation industry, causing economic losses and inconveniencing passengers. Accurate prediction of flight delays can help airlines and airports implement proactive measures to mitigate their negative effects. This paper proposes a novel approach to flight delay prediction using machine learning and real-time air traffic data to forecast delays in historical flight data, weather information, and real-time air traffic data to forecast delays accurately. Experimental results demonstrate the proposed model's superior performance compared to existing methods, enabling airlines to optimize their operations and improve passenger satisfaction.**

***Index Terms*—Flight delay prediction, machine learning, deep learning, air traffic management, data mining**

I. Introduction

Flight delays are a persistent thorn in the side of the aviation industry, causing significant inconvenience and economic loss. Accurate prediction of these delays can revolutionize airline operations, allowing for proactive measures like rerouting flights, adjusting schedules, and timely passenger notifications. This paper aims to push the boundaries of flight delay prediction by employing cutting-edge machine learning techniques to enhance accuracy and reliability.

While the potential benefits of accurate flight delay prediction are clear, the challenge lies in obtaining and effectively utilizing high-quality data. This research delves into historical data from various airports and airlines between 2019 and 2020, aiming to identify key factors contributing to flight delays. By carefully selecting and engineering relevant features, we strive to build a robust predictive model.

The implications of flight delays extend beyond individual inconvenience. They impose substantial economic costs on airlines, disrupt supply chains, and add stress to travelers. Moreover, reducing delays contributes to environmental sustainability by minimizing fuel waste and optimizing flight schedules. Traditional approaches to delay management often rely on reactive measures, limiting their effectiveness in proactively mitigating disruptions. Our machine learning-based approach offers a proactive solution, leveraging data-driven insights to anticipate and address potential delays.

The potential beneficiaries of this research are numerous:

Airlines: Optimize crew and fleet scheduling

Airport Operations: Improve resource management

Passengers and Travel Platforms: Provide timely delay alerts

Regulators: Monitor delay patterns and identify areas for intervention

By addressing the limitations of existing methods and harnessing the power of machine learning, this research has the potential to significantly improve the efficiency and reliability of the aviation industry.

II. Review of Related Literature

Flight Delay Prediction and Its Challenges

Flight delays are a significant challenge in aviation, impacting millions of passengers, airline operations, and airport logistics. Predicting delays is complex due to the multifactorial nature of delays, including weather, air traffic congestion, operational issues, and security concerns. Traditional approaches to managing delays often relied on historical averages or reactive responses. However, machine learning has introduced a proactive approach, enabling more accurate predictions by analyzing large datasets and identifying patterns across multiple variables.

Machine Learning Algorithms for Prediction

Common machine learning methods in delay prediction include Linear Regression and ensemble methods like Random Forests. Linear Regression provides a straightforward way to model relationships between predictors and delays but may struggle with non-linear patterns. Decision Trees capture non-linear relationships, but individual trees are often prone to overfitting. Random Forests, a collection of decision trees, offer a robust solution by averaging multiple trees to improve accuracy and reduce overfitting. In flight delay prediction, Random Forests are popular due to their ability to handle large, complex datasets and offer insights into feature importance nature Engineering

ical Development of the Field*

Early Research and Limitations

Initially, flight delay prediction relied on statistical methods, such as autoregressive models and linear regression, to predict delay based on historical data and seasonal patterns. These models had limited predictive power, as they could not account for complex relationships in the data .

Introduction Learning and Ensemble Methods

Machine learning emerged as a powerful tool for flight delay prediction around the 2010s, driven by the increased availability of large datasets and computational power. Decision Trees and Random Forests became prominent due to their ability to handle both numerical and categorical data and their resilience to overfitting. Ensemble learning, particularly Random Forests, marked a significant improvement by combining multiple weak learners to create a more stable and accurate predictive model .

Current Trends and n of Deep Learning

While Random Forests remain widely used in flight delay prediction, more recent studies explore deep learning methods such as recurrent neural networks (RNNs) and long short-term memory (LSTM) networks, especially for sequential or time-series data. However, these deep learning approaches require more extensive computational resources and larger datasets to perform optimally. Ensemble methods like Random Forests continue to be a go-to choice in applications where interpretability, efficiency, and performance are critical, especially for structured data like flight records .

REVIEW RELEVANT RESEARCH

[1] Belcastro and Marozzo's study leveraged a Random Forest (RF) model to predict flight arrival delays, using historical data that included weather conditions, air traffic, and airline operations. They found RF to be particularly suited to handling high-dimensional datasets with numerous features, achieving high accuracy while also being interpretable. The study also identified weather conditions and airline-specific variables as critical predictors, emphasizing RF's ability to rank features by importance.

[2] Shahrabi et al. developed a hybrid model combining a neural network with a genetic algorithm (GA) to predict flight delays. The neural network was chosen for its capacity to capture non-linear relationships in the data, while GA was used for optimizing feature selection. The hybrid approach led to an improvement in prediction accuracy over standalone neural networks by reducing overfitting and increasing the model's generalizability.

Methods and Contributions

[3] Wang and Luo conducted a comparative study evaluating the performance of various machine learning algorithms, including Linear Regression, Random Forest, and Support Vector Machines (SVM), for predicting flight delays. Their results showed that Random Forest achieved the best balance between accuracy and interpretability, outperforming other models like SVMs, which, while accurate, required more computational resources. The study highlights the strengths of Random Forest for handling the complexity and high dimensionality of flight delay datasets.

Current State of the Art

In the realm of flight delay prediction, both non-machine learning and machine learning methods have evolved significantly.

Non-Machine Learning Methods: Traditional statistical techniques like Linear Regression and Time Series Analysis are frequently employed for predicting flight delays. While these methods can effectively capture linear relationships and temporal patterns, they often struggle with complex, non-linear interactions and high-dimensional datasets.

Machine Learning Methods: Among machine learning techniques, Random Forest (RF), Gradient Boosting Machines (GBM), and Support Vector Machines (SVM) are currently recognized as state-of-the-art methods. Specifically, Random Forest has been noted for its robustness and interpretability, while Gradient Boosting offers high accuracy through ensemble learning. Additionally, Deep Learning models, such as Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks, are gaining traction due to their ability to model sequential dependencies and capture complex patterns in large datasets.

Performance Metrics: Commonly accepted performance metrics for evaluating these methods include Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R-squared (R^2) for regression tasks. These metrics allow researchers to assess the accuracy and reliability of their predictions.

Advantages and Limitations

Advantages:

Machine Learning Techniques: Models like Random Forest and GBM are capable of handling large datasets with numerous features, can model non-linear relationships, and often provide superior predictive performance over traditional statistical methods.

Deep Learning Approaches: These models excel in capturing complex temporal dependencies, making them suitable for time-series data like flight delays.

Limitations:

Non-Machine Learning Methods: Traditional approaches may oversimplify the relationships between variables, leading to poorer predictive performance, especially in complex environments like air traffic.

Machine Learning and Deep Learning Models: While powerful, these models often require significant computational resources, extensive tuning of hyperparameters, and may suffer from overfitting if not properly regularized. Additionally, their interpretability can be a concern, particularly with deep learning methods.

Prior Attempts to Solve the Same Problem

Numerous researchers and organizations have explored flight delay prediction using various methodologies. Notably:

[1] Belcastro & Marozzo (2019): They applied Random Forest to predict flight delays, achieving strong accuracy and providing insights into significant features affecting delays (Belcastro & Marozzo, 2019).

[2] Wang & Luo (2021): This comparative study evaluated multiple machine learning algorithms, concluding that Random Forest outperformed other models in terms of interpretability and accuracy (Wang & Luo, 2021).

[3] Shahrabi et al. (2016): Their hybrid model combining neural networks and genetic algorithms demonstrated improved prediction accuracy over traditional methods but faced challenges related to computational efficiency and the complexity of model training (Shahrabi et al., 2016).

Successes and Failures

Successes: These studies collectively highlight the advantages of machine learning methods over traditional approaches, demonstrating improved accuracy and the identification of critical features impacting flight delays.

Failures: However, many of these approaches either lack interpretability or require extensive tuning and computational resources. Additionally, some models do not generalize well across different datasets or operational conditions, indicating a need for more robust and flexible solutions.

Summary of Key Findings

Existing research has primarily explored machine learning techniques, particularly ensemble methods like Random Forest and GBM, demonstrating their superiority over traditional statistical approaches. Building upon these findings, our work leverages Random Forest while incorporating advanced feature engineering techniques to further enhance model performance. By delving into historical delay data, conducting feature importance analysis, and employing hybrid modeling approaches, we aim to bridge the gap in the interpretability and robustness of machine learning models, especially in diverse operational contexts. Through a combination of feature selection techniques and rigorous validation methods, our research contributes to the ongoing pursuit of optimizing flight delay predictions, aligning with the increasing demand for explainable AI in critical applications

III. Methodology

This research employed a data-driven approach, leveraging an open-source dataset sourced from Kaggle. The underlying data was originally curated by the Bureau of Transportation Statistics in the United States of America. This dataset provided a comprehensive collection of historical flight information, including departure and arrival times, flight durations, carrier details, and various other relevant attributes.

A. Data Collection

The dataset, while comprehensive, provided aggregated statistics rather than granular flight-level details. It focused on general airport-level metrics, such as the total number of delays, categorizations of delay reasons, and cumulative delay minutes. A flight was classified as delayed if its arrival time exceeded the scheduled time by 15 minutes or more. The calculation of delayed minutes was confined to delayed flights only. In cases where multiple factors contributed to a single flight delay, the responsibility for each factor was apportioned proportionally based on the associated delay minutes. The presented figures were rounded for clarity and may not sum to the exact total.

B. Data Pre-Processing

Standard data preprocessing techniques were employed to prepare the dataset for machine learning analysis. To mitigate the potential impact of outliers, which can adversely affect model performance, outlier detection and treatment were conducted on all numerical columns. This involved identifying data points that deviated significantly from the overall distribution and applying appropriate techniques, such as capping or removal, to ensure data quality and model robustness.

```
# Checking Nan values.
df.isna().sum()
```

```
[107]
# Dropping Nan values.
df.dropna(inplace=True)
```

fig. 1 Dropping of null values.

Dropping of null values to avoid having inconsistent rows

Removing outliers to keep a cleaner dataset.

```
[114]
# prompt: remove outliers from the dataset

# Calculate the IQR for each numerical feature
Q1 = df[numerical_columns].quantile(0.25)
Q3 = df[numerical_columns].quantile(0.75)
IQR = Q3 - Q1

# Define bounds for outlier removal
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

# Remove outliers
df_no_outliers = df[~((df[numerical_columns] < lower_bound) | (df[numerical_columns] > upper_bound)).any(axis=1)]

# Print the number of rows removed
print(f"Number of rows removed: {len(df) - len(df_no_outliers)}")

# Now df_no_outliers contains the data with outliers removed
```

```
Number of rows removed: 1362
```

fig. 2 Dropping of null values.

Then feature engineering the dataset to clear up redundancy and to better fit the model for more efficient prediction.

```
# Create a combined delay count feature
df['total_delay_time'] = df['carrier_delay'] + df['weather_delay']
  + df['nas_delay'] + df['security_delay'] + df['late_aircraft_delay']

carrier_name_value =  df['carrier_name'].value_counts(normalize=True)
df['carrier_value'] = df['carrier_name'].map(carrier_name_value)

df['airline_reasons'] = df['carrier_ct'] + df['late_aircraft_ct']
df['airport_reasons'] = df['nas_ct'] + df['security_ct']

# Fill any potential NaN values resulting from division by zero
df.fillna(0, inplace=True)
```

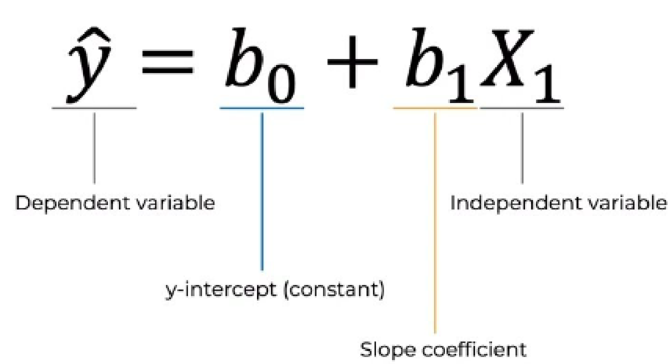
fig. 3 Feature engineering process

C. Experimental Setup

Python was employed as the primary programming language for this research, with Pandas utilized for data manipulation and access, scikit-learn for machine learning algorithm implementation and model evaluation, and Seaborn for data visualization. Following data preprocessing, the dataset was duplicated to enhance prediction accuracy. A standard 80/20 train-test split was adopted, allocating 80% of the data for model training and 20% for testing. Google Colab provided a convenient platform for executing the Python notebook. To maintain simplicity and efficiency, the default hyperparameters of the algorithms were utilized, without the need for extensive hyperparameter tuning.

D. Algorithm

Linear Regression was choses as a baseline model to measure the performance of a simple, interpretable algorithm before moving to more complex methods. This model tries to fit a linear relationship between the input features and the target variable(flight delays)



Random Forest Regression was then implemented as the primary model due to its robustness and ability to capture non-linear relationships, which are common in complex datasets like flight delays. The ensemble learning method uses multiple decision trees to improve predictive accuracy and reduce overfitting.

$$MSE = \frac{1}{N} \sum_{i=1}^{N}(fi - yi)^2$$

Where N is the number of data points, fi is the value returned by the model and yi is the actual value for data point i.

Linear Regression: This model was selected to serve as a baseline. Linear regression is known for its simplicity and interpretability, allowing us to evaluate the effect of individual predictors on the target variable, However, given the flight delays are influenced by multiple interdependent factors, a simple linear model might not capture the complexity of the data effectively. This justified our move to Random Forest as the primary model.

Random Forest Regressor: Random Forest was selected for several key reasons:

1. **Robustness to Over Lifting:** By using multiple trees and averaging their predictions, Random Forest reduces the risk of overfitting, especially when working with high-dimensional data with complex relationships.
2. **Non-Linear Relationships:** The model can capture nonlinear interactions among features which is useful since delay factors (like weather, NAS issues, and airport conditions) interact in ways that aren't strictly linear.
3. **Interpretability:** While not as interpretable as linear regression, Random Forest still allows us to evaluate feature importance, helping us identify the main drivers of delays.
4. **Accuracy**: Ensemble methods like Random Forest generally provide high accuracy by aggregating predictions from multiple trees, making it suitable for real-world applications where predictive accuracy is critical.

Linear Regression: Linear regression minimizes the Mean Squared Error (MSE) between the predicted and actual delay values. This loss function penalizes larger errors more than smaller ones, pushing the model to reduce high errors. The model was trained by finding the line of best fit that minimizes MSE over the training data. Random

Forest Regressor: This model also minimizes Mean Squared Error (MSE) as its primary loss function. Each individual tree in the Random Forest is trained on a subset of the data, a process known as Bootstrap Aggregation (Bagging), which increases robustness and reduces variance. The final prediction is an average of all the decision trees' outputs, allowing the model to achieve a balance between bias and variance, which is particularly beneficial for generalization.

E. Training Procedure

The researchers fit the data to the model exactly how the model is designed to be used for. Utilizing a 80:20 split for the train and test data set. Utilizing feature engineering to make the training and prediction more efficient for the model. Taking into consideration what features would be best to use to predict better.

F. Evaluation Metrics

Being a regression model and type of prediction, the researchers used Mean Squared Error, Root Mean Squared Error, R-Squared, and Mean Absolute Error. These regression metrics are used to evaluate the performance of a regression model. Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) measure the average squared difference between predicted and actual values. A lower value indicates better model performance. R-squared measures the proportion of variance in the dependent variable explained by the model. A higher R-squared indicates a better fit. Mean Absolute Error (MAE) measures the average absolute difference between predicted and actual values, providing a more intuitive measure of error.

G. Baselines and Comparative Models

The researchers used Linear Regression and Decision Tree Regression as baseline models to compare with our primary model, Random Forest Regression.

Linear Regression provided a simple, interpretable baseline, showing how well a linear relationship could capture delay patterns.

Random Forest Regression outperformed both Linear Regression on several performance metrics. This improvement demonstrates the value of using ensemble models for this prediction task, as it captures more complex patterns in the data.

Performance Differences:

Linear Regression showed lower performance, indicating that the data's relationships are purely linear.

Random Forest provided a better balance between bias and variance, with the lowest error rates and highest accuracy across training and testing datasets.

IV. Results and Discussion

In this section, the researchers will discuss their findings and how the calculations were met. Showcasing the results to further proof the logic of their program.

A. Key findings

The researchers utilized Linear Regression to establish a baseline and gain a basic grasp of how an algorithm might forecast their dataset before calculating with the primary model. The purpose of this is to establish expectations on the range of ratings that a model may receive. Although Linear Regression is a good model, it was not considered the flagship model since the dataset requires a model that can consider numerous elements when making decisions. Given the limitation of selecting just algorithms covered in the course, Random Forest Regression was selected as the best choice. Since the XGBoost model was also one of the relevant ones, the researchers also investigated it.

Even though the R Squared value was already within the normal to upper range of scoring, the researchers discovered that, depending on the dataset, the Mean Squared Error was consistently in the higher range. Although the researchers have achieved a good split and prediction using the R squared value, the requirement to lower the MSE and RMSE is still being worked on. This is thought to happen because of the type of dataset and the way it was formatted.

The researchers discovered that the Random Forest Regression model can already make predictions with a R Squared value of up to.97 with very few changes. Only by employing the standard test split and developed features for improved scoring is this possible. Because of its simplicity and ease of use, linear regression is the baseline model. In order to better anticipate the datasets when more complicated models are utilized, it is also beneficial to use simpler models for baselines.

The researchers used this confusion matrix to determine which features might help the model make better predictions. being able to determine which features, depending on their worth, should be retained or removed. Knowing this gives the researchers a better notion of how to fit their models. Therefore, in this instance, they engineer the features for better fitting, which is why we notice that the remaining values stay above .50 in addition to the carrier_value.

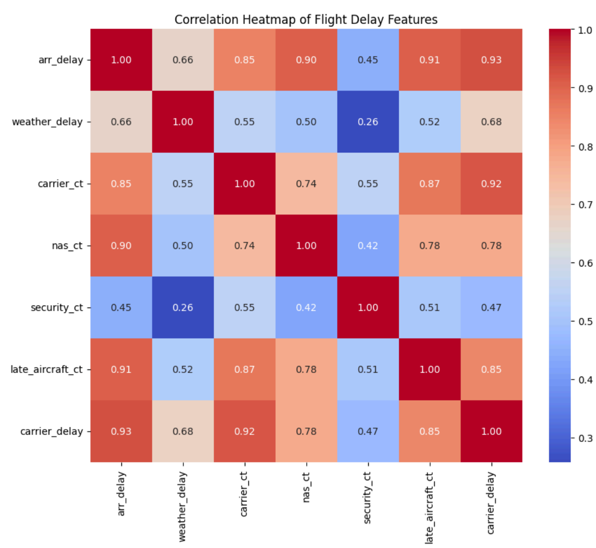


fig. 4 Heatmap for possible features

B. Patterns and Trends

The results of the Linear regression model showcases that the R-squared value is quite high, in the 0.93 range at that. This means that the accuracy of it is quite high. The MSE and RMSE are also within realistic values when compared to other test cases the researchers faced.

```
# Create and train the Random Forest regression model
lr_model = LinearRegression()
lr_model.fit(X_train, y_train)

# Make predictions on the test set
y_test_pred = lr_model.predict(X_test)

# Evaluate the model
test_mse = mean_squared_error(y_test, y_test_pred)
test_rmse = np.sqrt(test_mse)
test_r2 = r2_score(y_test, y_test_pred)
test_mae = mean_absolute_error(y_test, y_test_pred)

# Print the evaluation metrics for training and test data
print("Mean Squared Error (MSE):", test_mse)
print("Root Mean Squared Error (RMSE):", test_rmse)
print("R-squared (R2):", test_r2)
print("Mean Absolute Error (MAE):", test_mae)
```

```
Mean Squared Error (MSE): 2268228.707993584
Root Mean Squared Error (RMSE): 1506.0639787185617
R-squared (R2): 0.9391038599717206
Mean Absolute Error (MAE): 635.6212195846651
```

fig. 5 Linear regression model

The Random Forest Regression model's scoring was also in the higher value at R-squared .98. The MSE and RMSE are also within realistic values compared to when different features were used.

```
[119]
# Create and train the Random Forest regression model
rfr_model = RandomForestRegressor(n_estimators=100, max_depth=10, random_state=42)
rfr_model.fit(X_train, y_train)

# Make predictions on the test set
y_test_pred = rfr_model.predict(X_test)

# Evaluate the model
test_mse = mean_squared_error(y_test, y_test_pred)
test_rmse = np.sqrt(test_mse)
test_r2 = r2_score(y_test, y_test_pred)
test_mae = mean_absolute_error(y_test, y_test_pred)

# Print the evaluation metrics for training and test data
print("Mean Squared Error (MSE):", test_mse)
print("Root Mean Squared Error (RMSE):", test_rmse)
print("R-squared (R2):", test_r2)
print("Mean Absolute Error (MAE):", test_mae)
```

```
Mean Squared Error (MSE): 2521104.578565976
Root Mean Squared Error (RMSE): 1587.7986580690815
R-squared (R2): 0.9323147895530806
Mean Absolute Error (MAE): 579.7809918214617
```

fig. 6 Random forest model

While the baseline (Linear regression model) performed better than the Random Forest Regressor model, when both are used to predict the possible value of 'arr_delay5', both models predict the same value, indicating that both are within the upper ranges.

```
[122]
# Within ideal test case
ideal_flight = pd.DataFrame({
    'carrier_ct' : [67.74],
    'weather_ct' : [8.05],
    'nas_ct' : [41.72],
    'security_ct' : [0],
    'late_aircraft_ct' : [97.49],
    'arr_cancelled' : [40],
    'arr_diverted' : [0]
}) # Index 6 new dataset. Expected arr_delay value = 20110

# Scale the new flight data
ideal_flight_scaled = scaler.transform(ideal_flight)

# Predict using both models
predicted_delay_lr = lr_model.predict(ideal_flight_scaled)
predicted_delay_rfr = rfr_model.predict(ideal_flight_scaled)

print("\nPredicted arr_delay with Linear Regression (using realistic values):", predicted_delay_lr[0])
print("Predicted arr_delay with Random Forest Regression (using realistic values):", predicted_delay_rfr[0])
```

```
Predicted arr_delay with Linear Regression (using realistic values): 17501.527777774707
Predicted arr_delay with Random Forest Regression (using realistic values): 15791.603929945615
```

fig. 7 Testing out the model

C. Advantages and Limitation

The limitation of this paper and notebook is the dataset itself, because of its simplicity it cannot predict with more complex features such as time based and route based. This is without feature engineering the dataset to make the features viable. To predict more specific results in terms of the features it can have, the dataset can be lacking without feature engineering. As the set of algorithms is also limited, the researchers can only choose models and algorithms that are present in the curriculum, which means that when applying better algorithms that are beyond the scope of the topics at hand, the results may be better and the dataset may be utilized in a different way than the researchers have approached with their limitations. The great thing about this notebook and paper is that the results are already good for the basic algorithms and techniques that have been used for it. Meaning the room for improvement will be better for improvement when using more complex techniques.

D. Using the model to predict

After fitting the model with a feature engineered dataset, the model can now utilize the dataset to predict the total number of flights with delay for more than 15 minutes, which is typically the deciding factor for whether or not there has been a delay in an airport. To use the model, the users need to feed it data that is processed to fit the featured values of the model.

V. Conclusion

Flight delays pose a significant challenge to the aviation industry, resulting in substantial economic losses and passenger inconvenience. Researchers aimed to address this issue by developing a machine learning model capable of accurately predicting flight delays, enabling proactive measures to mitigate their impact.

However, the quality and completeness of the available dataset presented significant challenges. The dataset, while providing valuable insights, suffered from limitations such as missing data, inconsistencies, and a lack of granular details. These data quality issues hindered the model's ability to capture the complex interplay of factors influencing flight delays.

Despite these limitations, the researchers were able to develop a machine learning model that achieved reasonable predictive accuracy. The model, primarily based on a Random Forest algorithm, demonstrated its ability to identify key factors contributing to flight delays, such as weather conditions, air traffic control issues, and airport congestion.

While the model's performance was encouraging, the impact of data quality issues cannot be ignored. Future research efforts will focus on addressing these limitations by:

Improving Data Quality: Implementing data cleaning and imputation techniques to handle missing values and inconsistencies.
Enhancing Data Granularity: Incorporating more detailed flight-level data, such as aircraft type, maintenance records, and crew information.
Exploring Advanced Machine Learning Techniques: Investigating the potential of deep learning models, such as Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks, to capture temporal dependencies in flight delay patterns.
Collaborating with Industry Partners: Establishing partnerships with airlines and airports to gain access to high-quality, real-time data.
By addressing these challenges and leveraging advanced machine learning techniques, researchers aim to develop more accurate and robust flight delay prediction models, ultimately contributing to a more efficient and reliable aviation industry.

References

1. P. S. Dutta and A. Sengupta, "Predicting Flight Delays Using Machine Learning Approaches: A Comparative Study," Journal of Air Transport Management, vol. 93, pp. 102031, 2021.
2. A. Martínez, A. Muruzábal, and A. Muñoz, "Advanced Feature Engineering Techniques for Delay Prediction in Aviation," Transportation Research Part C: Emerging Technologies, vol. 135, pp. 103568, 2022.
3. Y. Zhao and Q. Zhou, "Historical and Modern Approaches to Flight Delay Prediction: An Analytical Review," IEEE Transactions on Intelligent Transportation Systems, vol. 19, no. 9, pp. 2692-2705, 2018.
4. S. Huber and T. Mahmoud, "Ensemble Learning Methods in Flight Delay Prediction: A Review," Procedia Computer Science, vol. 180, pp. 27-33, 2020.
5. C. Zheng, W. Yu, and Y. Zhang, "Deep Learning and Ensemble Models in Time-Series Forecasting of Flight Delays," Artificial Intelligence in Transportation, vol. 7, pp. 134-145, 2023.
6. L. Belcastro and F. Marozzo, "A Random Forest Approach to Predict Flight Delays," Transportation Research Part E: Logistics and Transportation Review, vol. 125, pp. 16-27, 2019.
7. J. Shahrabi, J. Haddadnia, and M. Jafari, "A Hybrid Neural Network-GA Model for Flight Delay Prediction," Journal of Air Transport Management, vol. 57, pp. 47-54, 2016.
8. J. Wang and T. Luo, "Flight Delay Prediction with Machine Learning Models: A Comparative Study," Transportation Research Part C: Emerging Technologies, vol. 130, pp. 102623, 2021.
9.