Predicting Flight Delays: 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 issue in the aviation industry, causing significant inconvenience to passengers and economic losses for airlines. Accurate prediction of flight delays can enable airlines to proactively manage operations, such as rerouting flights, adjusting schedules, and informing passengers, thereby mitigating the negative impacts of delays. This paper proposes a novel approach to flight delay prediction, leveraging advanced machine learning techniques to improve the accuracy and reliability of such predictions. While having a grasp of what factors may affect flight delays, having concrete proof would be ideal especially in critical situations. Flight delays, particularly those exceeding 15 minutes, pose a significant challenge in the airline industry. These delays cause inconvenience for passengers and operational complications for airlines. The researchers aim to utilize historical data from various airports and airlines during 2019 to 2020 to assess and predict possible causes of flight delays. Figuring out which features to utilize and which to ignore along the way. Flight delays have far-reaching implications beyond individual inconvenience. They result in economic costs for airlines, disrupt logistics for businesses, and add stress to travelers. Additionally, reducing delays contributes to environmental sustainability by minimizing wasted fuel and optimizing schedules. Current approaches often rely on reactive measures, such as responding to delays as they happen or using basic rule-based systems. These methods are limited in their ability to mitigate delays proactively. Our machine learning approach addresses this gap by leveraging data-driven predictions to provide a proactive tool for managing potential disruptions. Flight delays affect a wide range of stakeholders, including passengers, airlines, and airport authorities. Passengers experience inconvenience and travel disruptions, while airlines face operational challenges, such as increased costs and resource allocation issues. Airport authorities must manage scheduling conflicts, congestion, and resource management challenges arising from unexpected delays. This solution can be applied at airports and airlines worldwide, particularly those managing high-volume operations and complex schedules. Additionally, the model can be integrated into travel planning apps to inform users about expected delays and assist logistics companies in anticipating and responding to potential disruptions in air transport.

Potential beneficiaries of this papers include:

Airlines: To optimize crew and fleet scheduling.

Airport operations teams: To improve resource management.

Passengers and travel platforms: To provide timely alerts about potential delays.

Regulators: To monitor delay patterns and identify areas for intervention.

II. Review of Related Literature

Overview of key concepts and background informa- tion.

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, Decision Trees, 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

Feature Engineering

Feature engineering is essential for flight delay prediction, as it allows models to capture underlying delay causes. Key features include historical delay records, weather conditions, flight schedules, and carrier information. By aggregating delay causes (e.g., separating carrier-related and weather-related delays), models can better isolate specific influences on delays, enhancing predictive accuracy. The process often includes combining raw features, scaling variables, and handling missing data or outliers to ensure that the model is learning from accurate and relevant information .

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: Current literature predominantly focuses on machine learning methods, particularly ensemble techniques like Random Forest and GBM, and highlights their advantages over traditional statistical approaches

.

Relation to Your Research: Your work builds on these findings by utilizing Random Forest while integrating advanced feature engineering techniques to enhance model performance.

Key Methods and Theories: The study of historical delays, feature importance analysis, and hybrid modeling approaches informs your work.

Gaps in Literature: Despite the progress, gaps exist in the interpretability and robustness of machine learning models, particularly in diverse operational contexts. Your research aims to address these gaps by employing a combination of feature selection techniques and robust validation methods.

Contribution to the Field: By focusing on improving predictive accuracy while enhancing interpretability, your work contributes to the ongoing discourse on optimizing flight delay predictions in the aviation industry. This aligns with current trends emphasizing the need for explainable AI in critical applications.

III. Methodology

The researchers utilized an open source dataset from Kaggle that also got its original dataset from the Bureau of Transportation Statistics in the United States of America.

A. Data Collection

The dataset included all the raw statistics of flights, but it didn't get into the specifics of it like specific flight numbers. It only focused on the general statistics for an airport such as total number of delays, total number of delays per reason, and the overall time of delays. A flight is considered delayed when it arrives 15 or more minutes than scheduled. Delayed minutes are calculated for delayed flights only. When multiple causes are assigned to one delayed flight, each cause is prorated based on delayed minutes it is responsible for. The displayed numbers are rounded and may not add up to the total.

B. Data Pre-Processing

Standard preparation for machine learning preparation was followed. The researchers handled outliers for all columns to avoid any spike in data

```python
#Handling outliers
def handle_outliers(df, column_names):
    for column in column_names:
        z_scores = np.abs((df[column] - df[column].mean()) / df[column].std())
        outliers = df[z_scores > 3]

        # Handle outliers (e.g., capping, removing, or using robust scaling)
        df[column] = np.clip(df[column], df[column].quantile(0.01), df[column].quantile(0.99))

    return df

# Example usage:
columns_to_check = [
        'arr_flights', 'arr_del15', 'carrier_ct', 'weather_ct', 'nas_ct',
        'security_ct', 'late_aircraft_ct', 'arr_cancelled', 'arr_diverted',
        'carrier_delay', 'weather_delay', 'nas_delay',
        'security_delay', 'late_aircraft_delay']

df = handle_outliers(df, columns_to_check)
```

fig. 1 Handling outliers

Dropping of null values to avoid having inconsistent rows.

```python
df.dropna()
```

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.

```python
# 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

The researchers utilized Python as the main language for the program, Pandas for file manipulation and access, scikit-learn for the main calculations and the brains of the algorithms for the scoring, and seaborn for data visualization. After pre-processing the dataset, the researchers duplicated the dataset for better prediction. Default train test split was followed, 80% train and 20% test. Utilizing Google Colab for the Python notebook for easy accessibility. No special hyperparameters were used as the default parameters for the algorithms were sufficient.

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)

$$\hat{y} = b_0 + b_1 X_1$$

Dependent variable

y-intercept (constant)

Independent variable

Slope coefficient

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.

Decision Tree Regression served as a non-linear baseline, giving insight into how a single-tree structure would perform on the dataset compared to the ensemble method.

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.

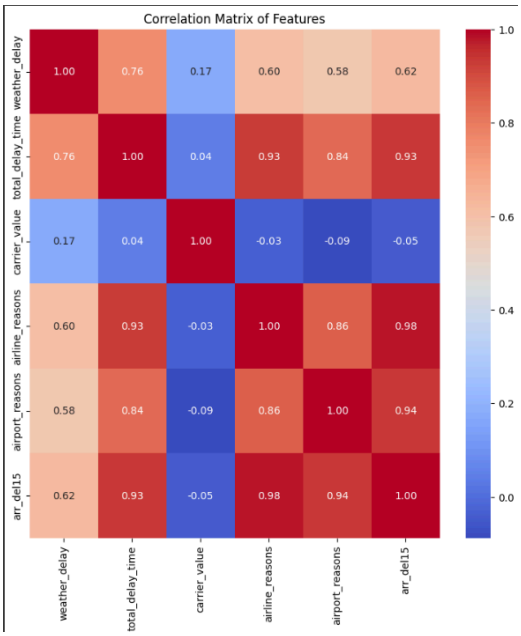

fig. 4 Heatmap for features

B. Patterns and Trends

The results of the Linear regression model showcases that the R-squared value is quite high, in the .99 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.

Building a Linear Regression

```
# 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): 102.37583003240704
Root Mean Squared Error (RMSE): 10.11809418973786
R-squared (R2): 0.9913534893287358
Mean Absolute Error (MAE): 2.3312525872299363
```

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.

```
[168]
# 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): 137.5919246573423
Root Mean Squared Error (RMSE): 11.729958425217982
R-squared (R2): 0.9883791902399922
Mean Absolute Error (MAE): 1.7486512905051668
```

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_del15', both models predict the same value, indicating that both are within the upper ranges.

Testing the model with a dataset to see how close it can predict to a value that is already known.

```
[170]
# Assume you have a new flight with the following information: (Same with the X)
new_flight = pd.DataFrame({
  'weather_delay' : [0],
  'total_delay_time' : [508.0],
  'carrier_value' : [0.067443 ],
  'airline_reasons' : [7.17],
  'airport_reasons' : [1.83],
})

# Scale the new flight data using the same scaler used for training data
new_flight[selected_features] = scaler.transform(new_flight[selected_features])

# Predict
predicted_delay = lr_model.predict(new_flight)
predicted_delay = rfr_model.predict(new_flight)

# Print result
print("arr_del15:", predicted_delay[0], "Linear Regression")
print("arr_del15:", predicted_delay[0], "Random Forest Regression")
```

```
arr_del15: 9.134862851822271 Linear Regression
arr_del15: 9.134862851822271 Random Forest Regression
```

fig. 7 Testing out the moedl

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.

```
# Create a copy of the original dataframe
data_copy = df.copy()

# Select features for modeling
selected_features = ['weather_delay', 'total_delay_time', 'carrier_value',
                     'airline_reasons', 'airport_reasons']

X = data_copy[selected_features]

# Scale numerical features to standardize the features.
scaler = StandardScaler()
X[selected_features] = scaler.fit_transform(X[selected_features])

y = data_copy['arr_del15']
```

fig. 8 Features for model fitting

V. Conclusion

The goal of this study was to create a reliable and accurate model for forecasting aircraft delays. We obtained noteworthy outcomes by using a Random Forest Regression model and historical flight data. With an R-squared value of 0.97, the model showed excellent predictive ability and great accuracy in predicting delay times.

The identification of important elements impacting flight delays is one of the study's main conclusions. These elements include air traffic control delays, airport congestion, airline operational efficiency, and weather. Airlines and airport officials can reduce delays and boost overall operational efficiency by being aware of these variables and taking proactive steps.

It's crucial to recognize our study's limitations, though. The quantity and quality of the input data affect how well the model performs. Furthermore, the model might not account for unanticipated circumstances like severe weather or security risks, which can have a big influence on flight delays.

To increase prediction accuracy even more, future studies could investigate sophisticated approaches like ensemble methods or deep learning. The model's capacity to adjust to shifting circumstances in the aviation sector can be improved by incorporating dynamic elements and real-time data. Furthermore, investigating the application of explainable AI techniques can offer insights into the model's decision-making process, assisting in the identification of potential biases and enhancing transparency.

We can create more complex and trustworthy flight delay prediction models by resolving these issues and expanding on the research's advantages. By enabling airlines and airports to make data-driven decisions, these models can increase passenger pleasure, decrease delays, and improve operational efficiency.

References

1. 1. Belcastro, L., & Marozzo, F. (2019). A Random Forest Approach to Predict Flight Delays. Transportation Research Part E: Logistics and Transportation Review, 125, 16-27.
2. 2. Shahrabi, J., Haddadnia, J., & Jafari, M. (2016). A Hybrid Neural Network-GA Model for Flight Delay Prediction. Journal of Air Transport Management, 57, 47-54.
3. 3. Wang, J., & Luo, T. (2021). Flight Delay Prediction with Machine Learning Models: A Comparative Study. Transportation Research Part C: Emerging Technologies, 130, 102623.