Learning Based Airline Delay Estimation

Team Name: GOAT

Group Members: Yuvaneswaren Ramakrishnan Sureshbabu, Santhosh Reddy Katasani Venkata, Arunaswin Gopinath, Pulipati Kushank.
Group Leader: Yuvaneswaren Ramakrishnan Sureshbabu

1 Introduction

Flight delays are a persistent challenge for the aviation industry, affecting airline operations, customer satisfaction, and overall travel efficiency. These delays stem from a variety of factors, including unpredictable weather conditions, air traffic congestion, and logistical complexities, making accurate prediction an intricate and demanding task. The primary objective of this project is to develop a robust system capable of forecasting flight delays with precision, enabling airlines to optimize their operations and enhance passenger experience.

To tackle this challenge, we leverage the power of Apache Spark to process and analyze large-scale, real-world flight data efficiently. By applying a comprehensive approach, we evaluate multiple machine learning and deep learning models to identify the most effective solution for delay prediction. The outcome of this project aims to empower airlines with actionable insights, reduce delays, and ensure smoother, more reliable travel experiences for passengers. Through this innovative effort, we address a critical issue in the aviation industry while showcasing the transformative potential of predictive analytics in real-world applications.

2 Relevant Work

Various studies have been undertaken to investigate a machine learning-based approach to airline delays prediction, each of them contributing unique aspects to the study of the issue. In [1], the authors used a Gradient Boosting Classifier to the American Airlines flight delay data, attaining an accuracy of 85.73 % based on different preprocessing techniques, including handling imbalanced data. [2] proposed a comprehensive framework using PySpark, where four classifiers were compared-Logistic Regression, Random Forest, Gradient Boosted Trees, and Decision Trees. The results showed Logistic Regression and Random Forest classifiers had the best results, demonstrating the scalability and speed advantages of PySpark as a tool to process large datasets for predicting delays on airlines.

The third work [3] utilized PySpark, testing multiple classifiers for predicting flight delays. It underscored the necessity of proper classifier strength for exponent performance and called attention to the relative benefit of distributed frameworks for large-scale prediction systems. These studies collectively illustrate how effectively combining machine learning algorithms with distributed processing tools like PySpark can improve the accuracy and operational efficiency of delay prediction models, and thus improve passenger experience in the aviation industry .

3 Data Collection

3.1 Dataset Overview

The dataset we used is Airline Delay Prediction Dataset (2014), obtained from the U.S. Department of Transportation. It provides detailed information about flight delays, including key factors such as weather conditions, carrier-related issues, and aircraft-specific causes. The dataset captures attributes like scheduled and actual flight times, delay durations, and reasons for delays.

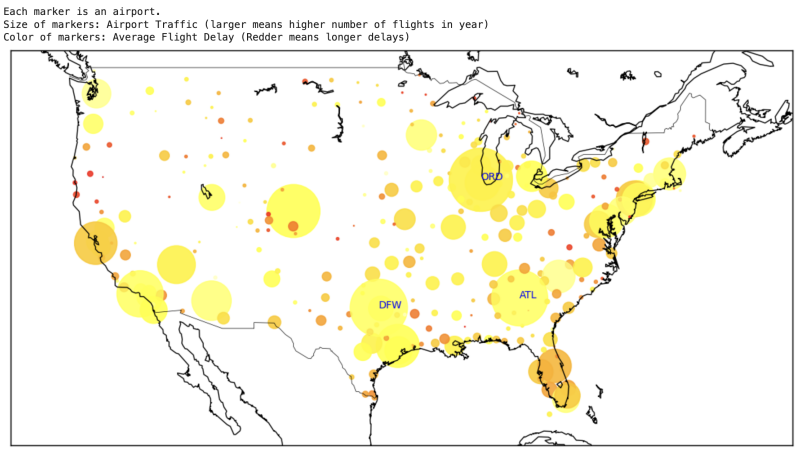
This dataset would form the backbone of predictive modeling in flight delay forecasting. Through the use of such data, airlines can undertake efficient planning and decision-making to improve overall operational efficiency and customers' satisfaction. The dataset also allows an in-depth study of how external factors-for example, weather conditions-affect flights' punctuality.

3.2 Dataset Characteristics

The dataset consists of 5,819,811 records spanning 18 attributes and is stored in a 300 MB CSV file. It encompasses data for flights throughout the entire year of 2014, providing a comprehensive overview of seasonal and temporal trends. The attributes in the dataset are diverse, categorized into numerical, categorical, and temporal data types. Numerical attributes include flight times (e.g., DepTime, ArrTime), delays (DepDelay, ArrDelay), and Distance. Categorical attributes cover airlines (UniqueCarrier) and airport codes (Origin, Dest). Additionally, temporal attributes capture details such as the date, time, day of the week, and month of flights. This structured dataset enables thorough analysis, including time-based trends, airline-specific delay patterns, and other operational insights.

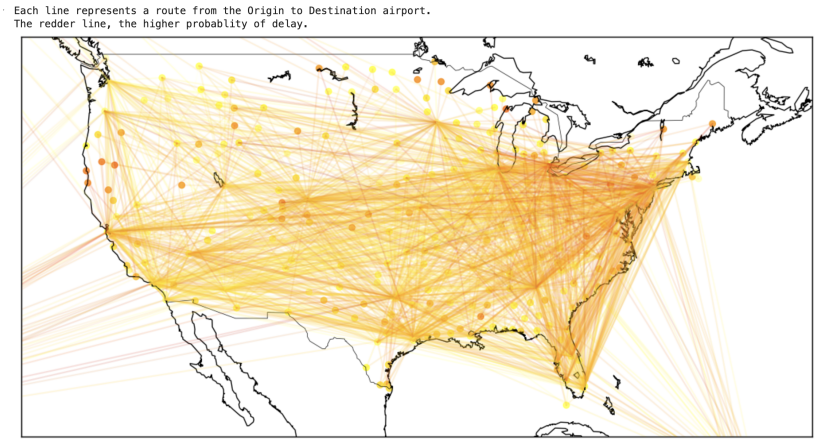
4 Exploratory Data Analysis

Exploratory Data Analysis (EDA) is the process of visually and statistically examining data to understand its main patterns, trends, and relationships. It helps identify key insights, anomalies, and guides further analysis. In our project, we performed EDA and created various graphs to uncover trends and important details within the data, setting a strong foundation for deeper analysis

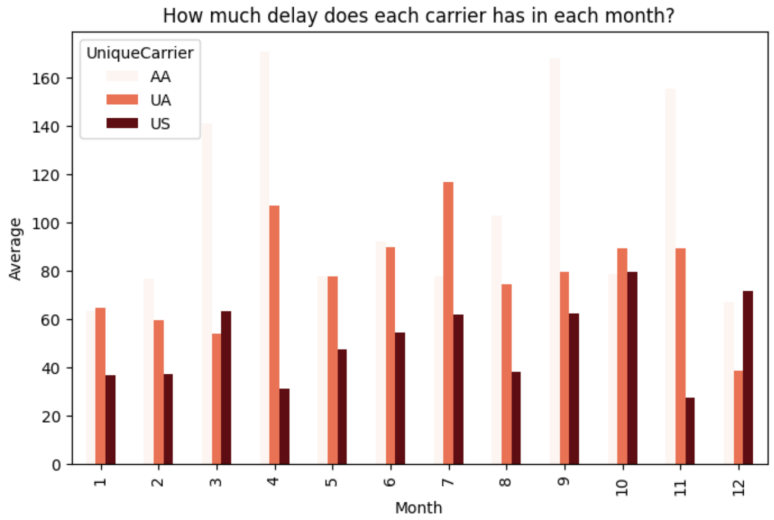


This map visualizes U.S. airports, with each circle representing an airport. The size of the circle indicates the airport's annual traffic, with larger circles for busier airports. The color gradient, from yellow to red, represents the average flight delay, with yellow indicating shorter delays and red showing longer delays.

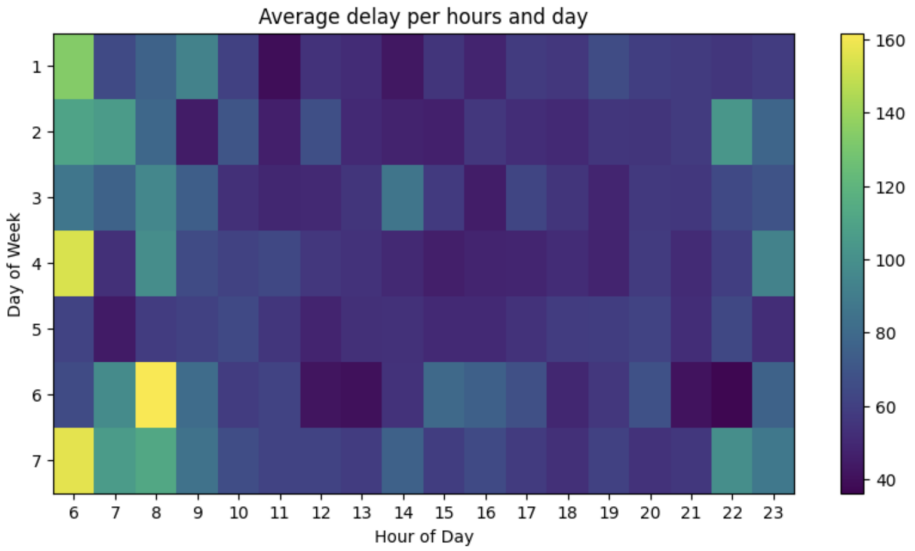
Airports in the Southeast, like Atlanta (ATL), have high traffic and relatively short delays, while Midwest hubs like Chicago (ORD) show moderate delays. Western airports, particularly in California, tend to have longer delays, reflected in more orange and red markers. The map highlights both airport activity and delay patterns across the U.S.



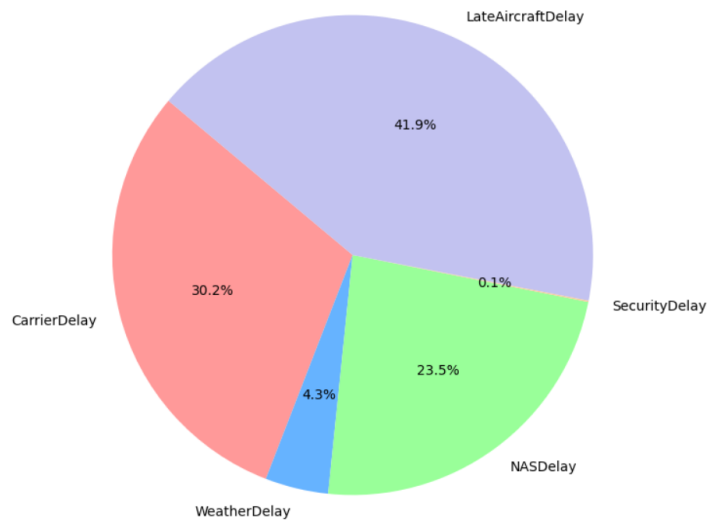
This graph illustrates the U.S. flight network, with lines representing flight routes and dots representing airports. The density of the lines shows route frequency, with major hubs like Chicago, Atlanta, and Los Angeles having more connections. The size and color intensity of the dots indicate airport traffic, with larger or darker dots for busier airports. Colors range from yellow (less traffic/delays) to orange/red (heavier traffic/higher delays). The graph highlights the distribution of flight delays across the network, with delays potentially propagating as planes move between airports.



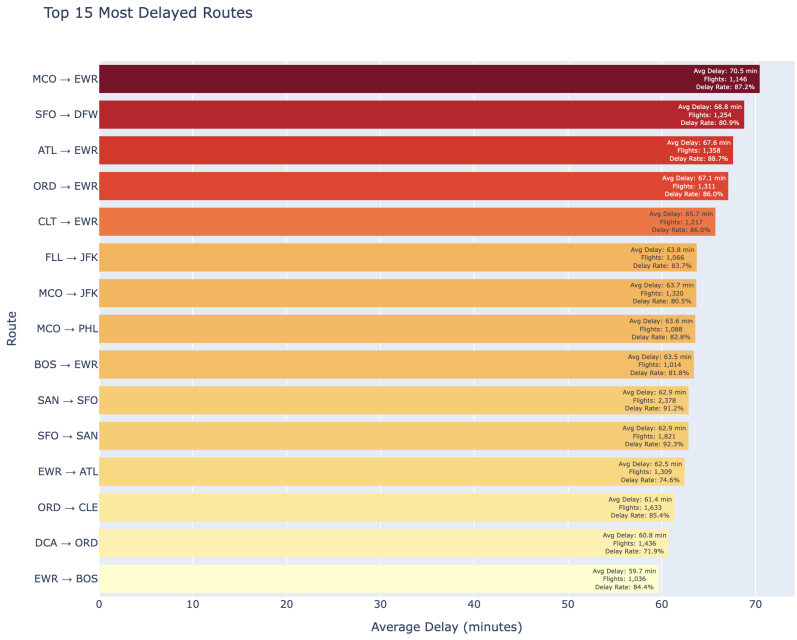
The bar chart displays the average monthly departure delay for three airlines (AA, UA, and US) from San Jose Airport (SJC). The x-axis shows months, and the y-axis shows average delay in minutes. AA, UA, and US are represented by light, medium, and dark green, respectively. The chart highlights noticeable monthly fluctuations in delays. For instance, AA shows significant delays in March and December, indicating potential seasonal or operational challenges during these months. In contrast, US generally has lower delays across most months, suggesting more consistent performance. This visualization is valuable for identifying seasonal delay patterns and understanding how different carriers are impacted by time-of-year factors.



This heatmap visualizes average flight delays by hour of day and day of week. The x-axis represents the time of day (6 AM to 11 PM), and the y-axis shows days of the week (Monday to Sunday). Each cell indicates the average delay for a given hour and day. The color scale, from purple (lower delays) to yellow (higher delays), shows delays ranging from 40 to 160 minutes, with intermediate colors representing varying delay levels.



This pie chart visualizes the contribution of each type of flight delay as a percentage of total delays. The data is aggregated by summing the delay columns using PySpark, then converted to a pandas DataFrame for further analysis. The percentage for each delay type is calculated based on the total sum of delays, and the chart shows how each category contributes to the overall delay distribution.



This visualization highlights the top 15 flight routes with the highest average delays, ranging from 60 to 70 minutes. Routes are color-coded from dark red (highest delays) to light yellow (lower delays). The most problematic routes include MCO → EWR (70.5 min avg delay), SFO → DFW (68.8 min avg delay), and ATL → EWR (67.6 min avg delay). Newark (EWR) is frequently listed as a destination, appearing in 5 of the top 15 delayed routes.

4.1 Class imbalance

There was high class imbalance in the target variable, whereby the majority class dominated the minority class tremendously; for instance, flights that were not delayed dominated the flights that were delayed. This created an imbalance in the data, which most machine learning models would tend to bias toward the dominating class, leading to poor performance in predicting the minority class.

Figure 1: Class Imbalance

4.2 Missing Data

During this phase, missing values were identified in several key fields, such as CarrierDelay, WeatherDelay, and other delay-related attributes. These missing values often occurred in cases where no delay was reported, which is expected for on-time flights.Additionally, canceled flights, though largely excluded from the dataset, occasionally had incomplete records, particularly for flight times and delay information. The proportion of missing data was analyzed to assess its potential impact on the dataset's reliability and subsequent modeling efforts.

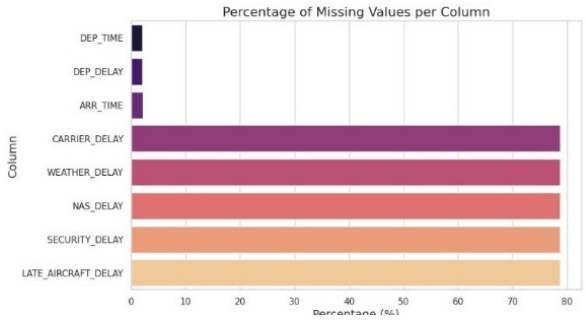


Figure 2: Percentages of missing data for each column

5 Pre-processing

5.1 Handling Missing Data

The dataset contained missing values in several important fields, such as CarrierDelay, WeatherDelay, and flight timing attributes. Since these fields play a crucial role in analyzing flight performance and delays, it was essential to handle the missing data effectively to maintain the integrity of the dataset. To address the missing delay reasons we used the techniques similarly used in [9]. , an imputation strategy was implemented. In cases where CarrierDelay or WeatherDelay values were missing, it was assumed that no delay occurred for that particular reason, and the missing values were replaced with 0. This approach ensured that the dataset remained comprehensive without misrepresenting the actual delay factors.

However, not all missing data could be logically imputed. Rows containing missing critical information, such as departure or arrival times, which are essential for accurate analysis, were removed from the dataset. Removing these incomplete rows helped maintain the dataset's reliability, ensuring that the analysis would not be affected by potential biases arising from incomplete or erroneous records. This careful handling of missing data allowed for a more robust and trustworthy dataset, minimizing the impact of data gaps on subsequent analyses.

5.2 Addressing Class Imbalance

In the dataset, the target variable—whether a flight was delayed or not—was highly imbalanced, with the majority class representing non-delayed flights significantly outnumbering the delayed flights. This imbalance posed a challenge to the machine learning model, as it tended to favor predictions for the majority class, leading to poor performance in predicting the minority class (delayed flights).

To address this issue we used the techniques similarly used in [10]. First, oversampling was applied to the minority class, generating synthetic samples to balance the dataset without losing information from the majority class. Additionally, class weight adjustment was used in algorithms like Logistic Regression, where the weights of each class were automatically adjusted to be inversely proportional to their frequencies in the dataset. This helped the model give more focus to the minority class, improving its predictive performance.

5.3 Feature Engineering

To enhance the predictive power of the dataset, several new features were created through feature engineering. A key addition was the Delayed Binary Feature, a binary variable indicating whether a flight was delayed or not, simplifying the target variable for binary classification models.Additionally, time-based features such as the day of the week and month were introduced to capture temporal patterns in flight delays. This allowed the analysis of trends, such as whether delays were more frequent on certain days or during specific months. Furthermore, distance categories were created to group

flights based on their distances, helping to explore the relationship between flight length and delay patterns. These engineered features improved the model's ability to identify important patterns and make more accurate predictions.

6 Machine Learning Models

The section discusses the experiments of various machine learning models developed to predict flight delays. Models were designed with the Logistic Regression classifier, Random Forest Classifier, and Decision Tree Classifier. Every model was selected because of its own strengths, which had to be elaborated with a series of hyper-parameters tuning and performance evaluations by using the dataset. A final choice was to come out from the model able to give the most reliable and actionable prediction in the airline's delay management.

6.1 Logistic Regression

Logistic Regression is a simple yet effective baseline model for binary classification tasks. In this project, it was trained on the provided dataset to predict whether a flight would be delayed. Since Logistic Regression produces probability scores as outputs, its performance can be evaluated using various metrics such as AUC-ROC, precision, recall, F1 score, and accuracy.

After addressing the class imbalance in the dataset using oversampling techniques, a Logistic Regression model was trained and evaluated on the test dataset. The results show that the precision achieved was 0.6496, which refers to the proportion of instances that were truly delayed out of the total predicted as delayed. This metric is especially important in scenarios where minimizing false positives is crucial. A precision of approximately 65% indicates that about two-thirds of the time, the model's prediction of a delay was correct, showing decent accuracy in identifying true delayed flights.

The recall, which measures the proportion of actual delayed instances that the model correctly identified, was 0.6457. This means that the model successfully captured most of the delayed cases but still missed a significant portion. A recall of approximately 64.6% suggests that while the model can detect many delays, there is still room for improvement in fully covering all potential delay instances.

The F1 score, which is the harmonic mean of precision and recall, was recorded at 0.6476. This metric provides a single evaluation point for balancing precision and recall, and an F1 score of around 64.8% reflects that the model maintained a moderate balance between the two. This result indicates that the Logistic Regression model performed reasonably well in handling the minority class of delayed flights.

Finally, the overall accuracy of the model was 0.6486, meaning that around 64.9% of the predictions were correct. While this accuracy might seem lower compared to the 99% achieved on the imbalanced dataset, it is much more reliable. The high accuracy on the imbalanced dataset was likely biased toward the majority class of non-delayed

flights, whereas the balanced dataset removes this bias, offering a more trustworthy measure of the model's true predictive power.

6.2 Logistic Regression After Hyperparameter Tuning

The objective of hyper-parameter tuning for Logistic Regression in this project was to enhance the model's performance by selecting the best combination of hyperparameters through a grid search methodology with cross-validation. This approach allowed us to identify the optimal settings that maximize the performance of the Logistic Regression model on the balanced dataset.

After performing the tuning, the best hyperparameters were determined. The regularization parameter, regParam, was set to 0.01. This value for regularization reduces the penalty on weights, enabling the model to capture meaningful relationships in the data without overfitting. The elasticNetParam was set to 0.0, implying the use of L2 regularization, where all coefficients are penalized equally. This helps in generalization without creating sparsity. The maximum number of iterations, maxIter, was increased to 50, which allowed the optimization algorithm to converge more efficiently and ensure that weights were adjusted correctly.

After tuning, the performance metrics showed improvements. The precision of the model increased to 0.7010, indicating a higher share of correctly predicted delayed instances out of all predictions made as delayed. This improvement shows that the model is now more reliable in predicting the minority class. The recall was slightly improved to 0.6459, showing that the model continues to identify delayed instances quite well, capturing most of the true positive cases. The F1 score, which balances precision and recall, rose to 0.6517. This slight increase suggests that the model has achieved a better balance between the two metrics, leading to more consistent performance.

The overall accuracy of the model after tuning remained relatively stable at 0.6488, similar to the untuned model. However, accuracy is a less reliable metric in the context of class imbalance, so the improvements in precision, recall, and F1 score are more significant for evaluating the model's performance. The ROC-AUC score, which measures the model's ability to discriminate between classes at various thresholds, improved slightly to 0.6472, indicating a modest increase in the model's ability to differentiate between delayed and non-delayed instances.

When compared to the untuned model, precision showed a notable improvement, which means the tuned model is better at predicting the minority class (delayed flights). Although the improvements in F1 score and recall were smaller, they demonstrate that the model has become more balanced and reliable after tuning. The tuned model better matches the complexity of the dataset, leading to a more stable and consistent performance overall.

Metric	Training (Before)	Testing (Before)	Training (Tuned)	Testing (Tuned)
AUC-ROC	0.7038	0.7041	0.7008	0.7010
Precision	0.6489	0.6496	0.6454	0.6459
Recall	0.6454	0.6457	0.6513	0.6517
F1 Score	0.6472	0.6476	0.6483	0.6488
Accuracy	0.6483	0.6486	0.6469	0.6472

Figure 3: Performance Metrics: Pre-Tuning vs Post-Tuning for Logistic Regression

6.3 Random Forest Classifier

Using the techniques similarly used in [4] RF was chosen because it is an ensemble learning method that combines multiple decision trees to improve robustness and accuracy. This model can handle both numerical and categorical data, so it was quite suitable for our dataset. Hyperparameters such as the number of trees and maximum depth were tuned for better performance.

The performance of Random Forest on the testing dataset is summarized as follows: The performance of the Random Forest model on the testing dataset showed an AUC-ROC of 0.6942, indicating good but slightly lower discriminative ability compared to Logistic Regression. Its Precision of 0.6131 reflected moderate accuracy in correctly predicting delayed flights, though some false positives were present. The model excelled in Recall with the highest value of 0.7444, effectively identifying delayed flights. The F1 Score of 0.6724 demonstrated a reasonable balance between Precision and Recall, while an Accuracy of 0.6446 highlighted its overall correctness in classifying delays and non-delays.

The Random Forest Classifier's high Recall suggests that it's particularly effective at detecting delayed flights, which aligns with our project's objective to minimize the impact of missed delays (false negatives). However, its Precision indicates a tendency to overpredict delays, which may result in unnecessary alerts. The slightly lower AUC-ROC compared to Logistic Regression points to some difficulty in differentiating delayed and on-time flights in certain cases.

Despite its strong performance, Random Forest has a relatively higher computational cost due to its ensemble processing. Additionally, while it generalizes well, the lower Precision compared to Logistic Regression highlights its trade-off between capturing delayed flights and avoiding false positives.

6.4 Decision Tree Classifier

For interpretability and efficiency, similar to the approaches used in [5] the Decision Tree Classifier was done. Although relatively simpler than Random Forest, the former provided insights into how the individual features influenced the predictions, while its decision rules were easy to obtain and trace the logic of a prediction.

The performance of Decision Tree Classifier Results is summarized as follows: The performance of the Decision Tree Classifier on the testing dataset revealed an AUC-ROC of 0.5386, the lowest among all models, indicating a limited ability to distinguish between delayed and non-delayed flights. Its Precision of 0.6049 reflected reasonable correctness in predicting delays, though lower than Logistic Regression. With a Recall of 0.6867, the model demonstrated its capacity to identify a significant number of delayed flights. The F1 Score of 0.6432 indicated a moderate balance between Precision and Recall, while its Accuracy of 0.6780, the highest among all models, showcased its overall effectiveness in classifying flights as delayed or on-time.

The Decision Tree's high Accuracy highlights its ability to correctly classify most flights. Its Recall, although lower than Random Forest, demonstrates its competence in identifying delays. However, the low AUC-ROC score indicates that the model struggles with distinguishing between the two classes, reducing its reliability in more nuanced cases. This makes it less effective compared to Random Forest or Logistic Regression for this dataset.

One of the primary challenges with Decision Trees is their tendency to overfit, particularly with large datasets. While this wasn't a significant issue here, it's worth noting as a limitation. Additionally, while the model's interpretability is a strength, its relatively lower AUC-ROC score and moderate Precision suggest that it may not be the most reliable choice for this problem.

7 Machine Learning Models - Observation

The performance of the three models—Logistic Regression, Random Forest, and Decision Tree Classifier—varied across different evaluation metrics, each offering distinct strengths and weaknesses. Logistic Regression emerged as the model with the highest AUC-ROC (0.7041), showcasing strong discriminative power. It maintained a balanced Precision (0.6496) and F1 Score (0.6476) with an overall Accuracy of 0.6486, making it a reliable baseline. However, its Recall (0.6457) was slightly lower than that of Random Forest, indicating a reduced capability in identifying delayed flights.

Random Forest stood out with the highest Recall (0.7444), making it the most effective model for detecting delays, which aligns closely with the project's objective of minimizing missed delays. It also demonstrated balanced overall performance with an F1 Score of 0.6724 and a good AUC-ROC of 0.6942. However, its Precision (0.6131) was lower than Logistic Regression, suggesting a tendency to overpredict delays. Additionally, its ensemble nature made it computationally expensive. On the other hand, the Decision Tree Classifier achieved the highest Accuracy (0.6780), reflecting strong

overall classification, and delivered reasonable Recall (0.6867). Yet, its low AUC-ROC (0.5386) and moderate Precision (0.6049) limited its reliability, especially for nuanced predictions.

Considering these results, Random Forest was identified as the most suitable machine learning model for this task, given its high Recall and balanced performance. Nevertheless, recognizing the limitations of all three models, we transitioned to exploring advanced deep learning architectures such as CNNs, DNNs, and CRNNs to enhance predictive accuracy and address the complexities inherent in flight delay prediction.

8 Deep Learning Models

8.1 DNN - Fully Connected Feed Forward Network

Our DNN model for predicting flight delays is inspired by approaches like the one described in [6], which uses deep learning to analyze spatio-temporal data. Just like DeepST, which models crowd flow by capturing both spatial and temporal dependencies, our model leverages similar techniques to predict flight delays based on real-time data.

The deep learning model implemented for this project is a fully connected deep neural network (DNN) developed to classify flights as delayed or not delayed based on a range of operational features. The architecture comprises an input layer, four hidden layers with 64, 128, 64 and 32 neurons respectively, and an output layer. The hidden layers use the **ReLU** (Rectified Linear Unit) activation function, which is particularly effective for deep networks as it introduces non-linearity and enables the model to learn complex relationships within the data. ReLU also helps mitigate the vanishing gradient problem, a common issue in deep networks, ensuring stable learning during backpropagation. The output layer employs a sigmoid activation function, producing a probability score that allows for binary classification between delayed and on-time flights.

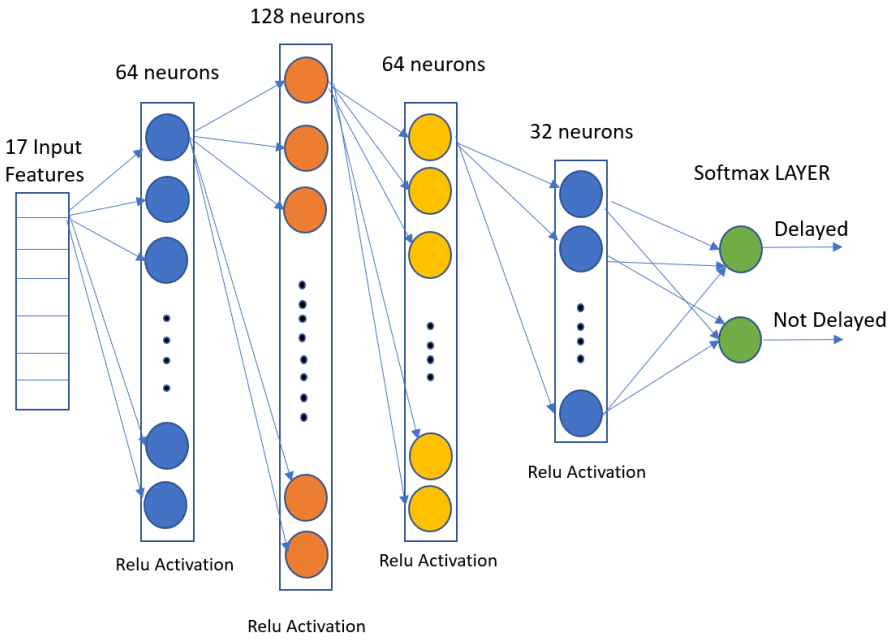


Figure 4: DNN - Architecture diagram

The DNN architecture is well-suited for airline delay prediction as it effectively handles high-dimensional data and uncovers intricate dependencies among features like carrier, origin, and destination. By allowing all relevant features to contribute, the model learns subtle relationships that impact delays. Its ability to represent hierarchical patterns in the data, combined with scalability and flexibility, makes it robust for capturing the complexities of airline operations and delivering actionable predictions.

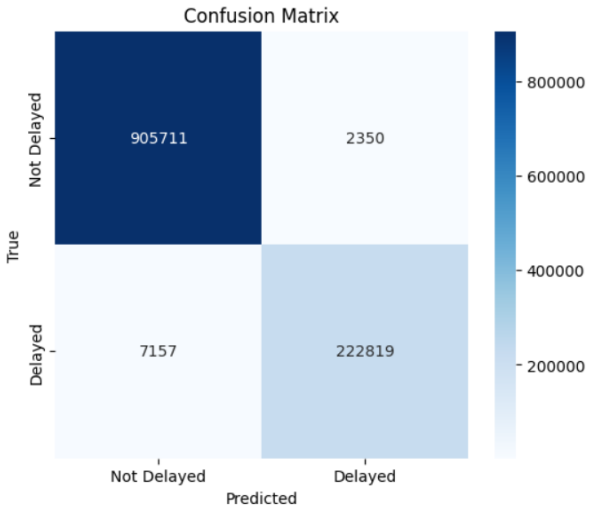


Figure 5: DNN - Confusion matrix

The DNN model showed impressive performance, achieving a test accuracy of **99.16%** using an 80-20 train-test split. The confusion matrix indicates that the model success-

fully identified most delayed and non-delayed flights, with only a small number of misclassifications. It was particularly effective at predicting non-delayed flights, although it had a slightly harder time with delayed flights, which highlights the challenges of predicting such events. Overall, the model's high accuracy and reliable predictions make it a valuable tool for understanding and anticipating flight delays in real-world applications. Figure 5 gives the confusion matrix for this model

8.2 CNN - Convolutional Neural Network

The Convolutional Neural Network (CNN) model designed for predicting flight delays effectively utilizes its architecture to capture sequential patterns within the dataset. The model begins with a **Conv1D** layer comprising 64 filters and a kernel size of 3, activated using the ReLU function to identify localized patterns across the features. This is followed by a max-pooling layer, which reduces dimensionality while retaining significant features. A second Conv1D layer with 128 filters and a kernel size of 3 further enhances feature extraction. The output from these layers is flattened and processed through a dense layer with 64 neurons, also using ReLU activation. To prevent overfitting, a dropout layer with a rate of 0.5 is included before the final output layer, which employs a sigmoid activation function to produce the binary classification of flight delays.

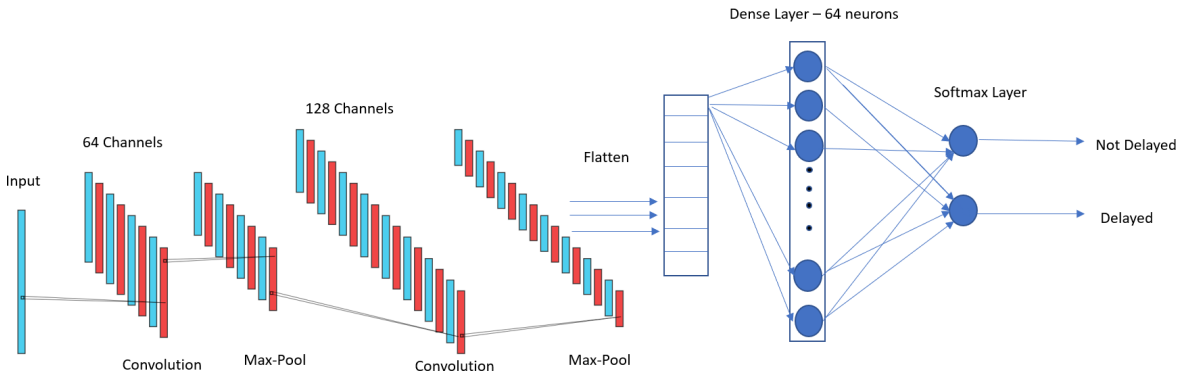


Figure 6: CNN - Architecture Diagram

The dataset was reshaped into a three-dimensional structure to meet the input requirements of 1D convolutional layers, with feature standardization ensuring consistent scaling for efficient training. The CNN used the **Adam optimizer** and **binary cross-entropy loss** for binary classification, while max-pooling layers reduced computational complexity without losing key patterns. This architecture enabled the model to capture intricate relationships among features like carrier, origin, and destination, proving to be a robust method for predicting flight delays. Similar to the multi-task CNN approach described in [7], our model leverages convolutional neural networks to extract complex patterns from multi-dimensional input data effectively.

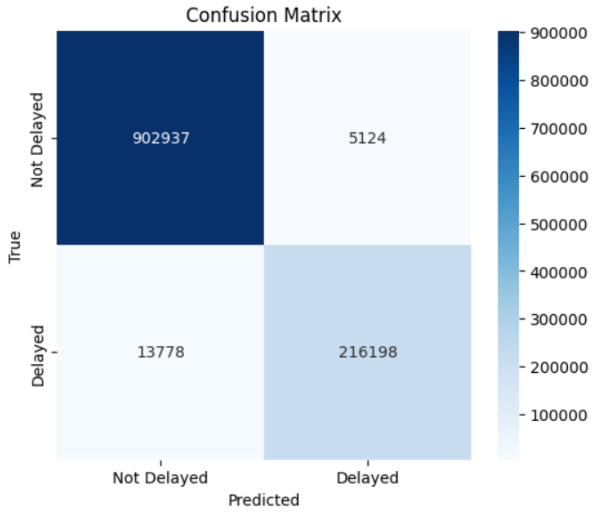


Figure 7: CNN - Confusion Matrix

The CNN model performed well, achieving a test accuracy of **98.34%** in predicting whether flights were delayed or not. Although CNNs are usually designed for tasks involving spatial data, like images, they can still work effectively with structured data. In this case, the model was able to recognize patterns in the features and make reliable predictions. While its accuracy was slightly lower than that of the DNN model, the CNN still delivered strong results, showing its versatility and ability to adapt to different types of data. Figure 7 gives the confusion matrix for this model

8.3 CRNN - Convolutional Recurrent Neural Network

We have developed a convolutional-recurrent neural network (CRNN) model to predict flight delays based not only on various features but also on those most important for delays.

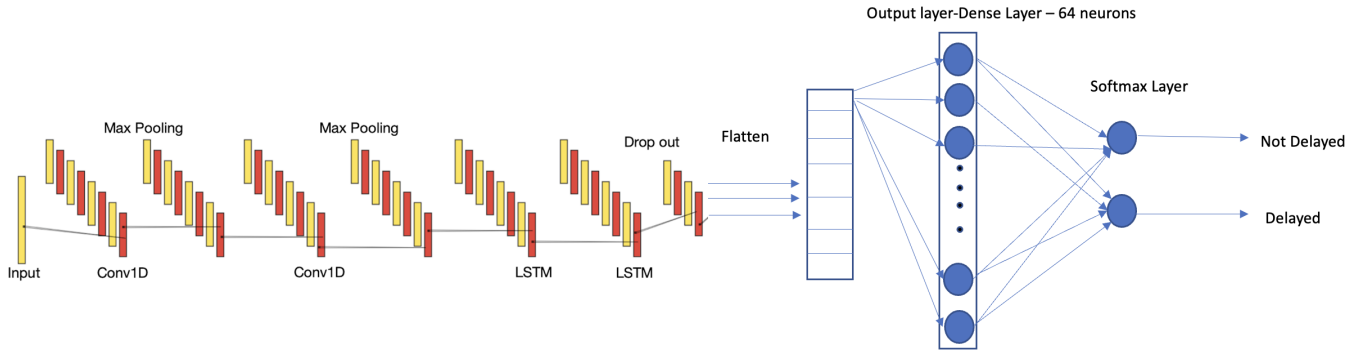


Figure 8: CRNN - Architecture Diagram

The model architecture consists of **Conv1D** layers for extracting spatial features from the data. These layers are intended to detect relevant patterns associated with the flight itself, e.g., carrier, origin, and destination. After each convolutional layer, **Max-Pooling1D** layers are applied to lessen the spatial dimensions of the feature maps,

retaining mostly the vital features while minimizing computation. Following the convolution and pooling layers, **LSTM** layers have been added to capture the temporal dependencies inherent in the data that are necessary to determine the sequence of events leading to delays. The LSTM layers allow the model to learn patterns that play out over time, such as changing flight schedules or delays throughout the day. The model also has a **Dropout** layer to reduce overfitting. The last **Dense layer** applies a **Sigmoid activation** function to solve the binary classification task (delayed or not delayed). We have trained our model with the Adam optimizer and binary cross-entropy loss, measuring performance in terms of accuracy. The dataset is then split into training and test sets, where 80 percent of the data was used for training and 20 percent for testing.

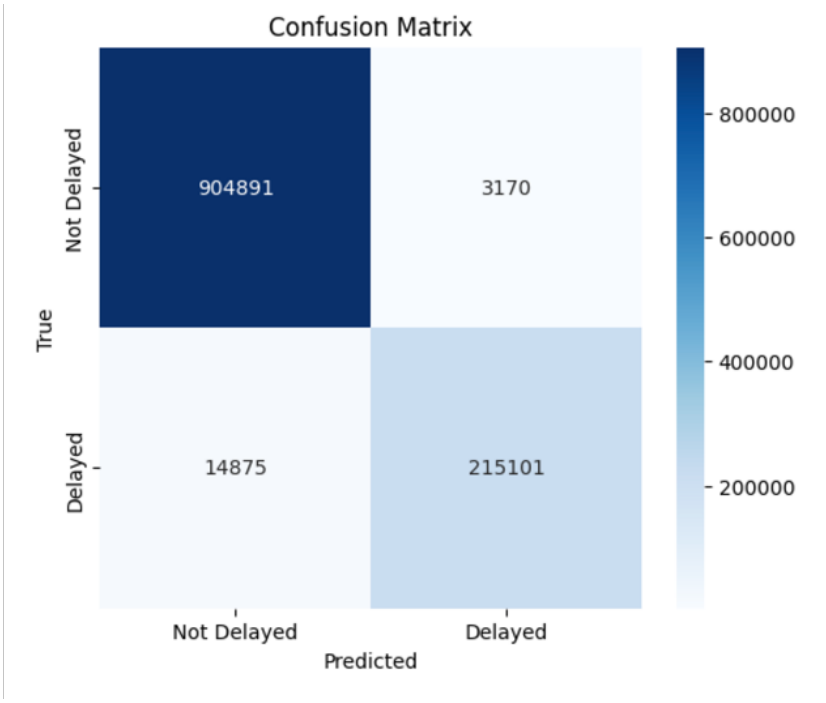


Figure 9: CNN - Confusion Matrix

With a test accuracy of 98.41%, the CRNN model proves to be reliable for predicting flight delays, distinguishing delayed flights from on-time ones with confidence, as shown in the confusion matrix Figure 9.

The key feature which makes this model so strong is the hybrid architecture that combines convolutional layers that extract essential features, such as the carrier or airport information, with LSTM layers which encode time-based patterns. This blending of architectures makes it extremely effective in dealing with the sequence-based nature of flight delays. By learning spatial-temporal relationships, CRNN will adapt to complex patterns in the data, making it a strong candidate in the solution for delay prediction. A similar approach is proposed in reference to [8] in which CNN and LSTM networks work hand-in-hand to enhance prediction accuracy by capturing spatial and temporal dependencies.

9 Deep Learning Models - Observation

In this section, we compare the performance of three deep learning models: DNN, CNN, and CRNN in predicting flight delays. The models were evaluated on the same dataset, with the following test accuracies: the DNN achieved an accuracy of 0.9916, the CNN reached 0.9834, and the CRNN performed with an accuracy of 0.9841. While all three models demonstrated strong performance, the DNN slightly outperformed the CNN and CRNN models.

The superior performance of the DNN can be attributed to its fully connected architecture, which enables the model to learn complex, nonlinear relationships between the features. This allows the DNN to capture intricate patterns in the data by considering all feature interactions. In contrast, the CNN and CRNN models are more suited for identifying spatial or sequential dependencies. The CNN excels in detecting local patterns, while the CRNN combines convolutional and recurrent layers to capture both spatial and temporal relationships. However, for this dataset, the DNN's architecture proved more effective at handling the large and diverse set of features. Despite the differences in architecture, all three models demonstrate the strength of deep learning techniques in predicting flight delays.

10 Discussion

In this project, we compared the performance of machine learning (ML) and deep learning (DL) models for predicting flight delays. Among the ML models, Logistic Regression, Random Forest, and Decision Tree each had their strengths, but none were able to outperform the deep learning models. Logistic Regression showed solid discriminative power but struggled with recall, missing a fair number of delayed flights. Random Forest, while excelling in recall, tended to overpredict delays, which lowered its precision. The Decision Tree model, although simple and easy to interpret, was less reliable, showing poor results in terms of AUC-ROC and precision, making it less suited for this task.

On the other hand, the deep learning models, especially the DNN, demonstrated a clear advantage. The DNN's fully connected architecture allowed it to learn complex, non-linear relationships between features, making it well-suited for large and diverse datasets like those used for flight delay prediction. While the CNN was effective at identifying spatial patterns and the CRNN combined spatial and temporal dependencies, the DNN proved the most capable of handling the complexity of the data. What sets deep learning apart from traditional machine learning is its ability to automatically learn relevant features from raw data, which allows it to better capture intricate patterns and variable interactions. This makes deep learning models particularly effective for complex prediction tasks, like flight delays, where the relationships between variables are highly non-linear and multifaceted.

11 Conclusion

This project focused on predicting airline delays, utilizing both traditional machine learning and deep learning models. While the machine learning models showed solid performance, the deep learning models, particularly the DNN-based approaches, demonstrated a clear advantage in capturing complex patterns in the data. By effectively analyzing the relationships between various factors, these models were able to predict delays with greater accuracy. As a result, we have successfully predicted flight delays, proving the power of deep learning in this area. With further improvements, these models have the potential to provide even more reliable predictions, contributing to enhanced efficiency in airline operations.

12 Future Enhancements

Looking ahead, one exciting area for improvement is the use of Transformer models, which have recently made a significant impact in deep learning, especially for sequential data. Unlike traditional RNNs and LSTMs, Transformers are better at capturing long-range dependencies. By using attention mechanisms, they can focus on the most important data points, which could help improve flight delay predictions by identifying complex patterns in both time and space.

Another potential enhancement involves expanding the feature set by incorporating additional external data. While we currently focus on operational features like carrier, origin, and destination, integrating other factors such as weather conditions or real-time airport traffic could provide a more complete picture of the factors influencing flight delays. This would likely lead to more accurate predictions.

We could also explore advanced regularization techniques, such as dropout or batch normalization, to help prevent overfitting. These methods would be particularly useful for deep learning models like DNNs, CNNs, and CRNNs, ensuring they generalize better to new, unseen data and are not too tailored to the training set.

Finally, adopting additional evaluation metrics, such as Precision-Recall curves or AUC-ROC, could offer deeper insights into how well the model performs, especially when dealing with imbalanced data. These metrics would help us better understand how well the model identifies delayed flights and pinpoint areas for further improvement.

References

[1] N. Chakrabarty, "A Data Mining Approach to Flight Arrival Delay Prediction for American Airlines," 2019 9th Annual Information Technology, Electromechanical Engineering and Microelectronics Conference (IEMECON), Jaipur, India, 2019, pp. 102-107, doi: 10.1109/IEMECONX.2019.8876970.

[2] N. L. Kalyani, G. Jeshmitha, B. S. Sai U., M. Samanvitha, J. Mahesh and B. V. Kiranmayee, "Machine Learning Model - based Prediction of Flight Delay,"

2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, India, 2020, pp. 577-581, doi: 10.1109/I-SMAC49090.2020.9243339.

[3] A. R. N. and S. M., "Enhancing Airline Operations by Flight Delay Prediction - A PySpark Framework Approach," 2023 International Conference on Ambient Intelligence, Knowledge Informatics and Industrial Electronics (AIKIIE), Ballari, India, 2023, pp. 1-4, doi: 10.1109/AIKIIE60097.2023.10389960

[4] M. Bardach, E. Gringinger, M. Schrefl and C. G. Schuetz, "Predicting Flight Delay Risk Using a Random Forest Classifier Based on Air Traffic Scenarios and Environmental Conditions," 2020 AIAA/IEEE 39th Digital Avionics Systems Conference (DASC), San Antonio, TX, USA, 2020, pp. 1-8, doi: 10.1109/DASC50938.2020.9256474.

[5] R. T. Reddy, P. Basa Pati, K. Deepa and S. T. Sangeetha, "Flight Delay Prediction Using Machine Learning," 2023 IEEE 8th International Conference for Convergence in Technology (I2CT), Lonavla, India, 2023, pp. 1-5, doi: 10.1109/I2CT57861.2023.10126220.

[6] Zhang, Junbo, Yu Zheng, Dekang Qi, Ruiyuan Li, and Xiuwen Yi. "DNN-based prediction model for spatio-temporal data." In Proceedings of the 24th ACM SIGSPATIAL international conference on advances in geographic information systems, pp. 1-4. 2016.

[7] M. Qiu et al., "A Short-Term Rainfall Prediction Model Using Multi-task Convolutional Neural Networks," 2017 IEEE International Conference on Data Mining (ICDM), New Orleans, LA, USA, 2017, pp. 395-404, doi: 10.1109/ICDM.2017.49.

[8] S. H. Rafi, S. R. Deeba, and E. Hossain, "A Short-Term Load Forecasting Method Using Integrated CNN and LSTM Network," *IEEE Access*, vol. 9, pp. 32436-32448, 2021, doi: 10.1109/ACCESS.2021.3058888.

[9] Y. Tijil, N. Dwivedi, S. K. Srivastava and A. Ranjan, "Flight Delay Prediction Using Machine Learning Techniques," 2024 IEEE International Conference on Computing, Power and Communication Technologies (IC2PCT), Greater Noida, India, 2024, pp. 1909-1913, doi: 10.1109/IC2PCT60090.2024.10486482. keywords: Support vector machines;Atmospheric modeling;Machine learning;Forestry;Companies;Vectors;Communications technology;Machine learning;Flight Delay Prediction;Random Forest;Logistic Regression;Support Vector Machine (SVM),

[10] S. Sukhanov, A. Merentitis, C. Debes, J. Hahn and A. M. Zoubir, "Combining SVMS for Classification on Class Imbalanced Data," 2018 IEEE Statistical Signal Processing Workshop (SSP), Freiburg im Breisgau, Germany, 2018, pp. 90-94, doi: 10.1109/SSP.2018.8450746. keywords: Support vector machines;Training;Signal processing;Conferences;Force;Machine learning algorithms; Complexity theory;SVMs;Class imbalance;Undersampling;Ensemble learning methods

[11] Green, C. and Black, D. (2015). Random forests for classification in ecology. Ecological Applications, 25(2):234–245.