JetLagged - Prediction of Airline Flight Delay

Ved Waje[1], Abhirat More[2], Pranita Bannore[3], and Harshita Lohana[4]
Student, Department of Computer Engineering, VESIT, Mumbai, India

***Abstract*—Flight delays and cancellations are prevalent challenges in the airline industry, arising from factors such as weather conditions, technical malfunctions, air traffic congestion, and crew availability. This project focuses on developing a deep learning-based application to identify and analyze the primary causes of flight disruptions. The application integrates diverse data sources, including weather reports, flight schedules, and historical delay records, to detect patterns and correlations contributing to delays. An advanced predictive approach is employed using the LightGBM regressor combined with Optuna for Bayesian optimization, ensuring high accuracy and efficient model performance. By analyzing complex datasets, the application provides real-time alerts and recommendations for passengers, aiding airlines in enhancing decision-making and operational efficiency. Ultimately, this project aims to improve air travel reliability, reduce operational costs, and contribute to the advancement of smart transportation systems**

***Index Terms*—Flight delay prediction, Machine learning, Classification, Light GBM, Linear Regression.**

I. INTRODUCTION

With the rapid expansion of air travel in India, flight delays have become a major concern for both airlines and passengers. These delays not only result in lost time but also reduce passenger trust, leading to financial losses for airlines and damage to their reputation. Accurate prediction and monitoring of flight delays are important for improving operations and making travel more reliable. This study focuses on predicting flight arrival delays by analyzing various factors, including departure delay time, flight distance, and weather conditions. Using these key factors helps airlines reduce disruptions and minimize inconvenience for passengers.

To achieve this, a predictive model was developed using the LightGBM regressor, known for its efficiency and strong performance with large datasets. The model predicts flight delays in minutes based on multiple input factors, such as weather data, departure time, and distance between airports. Model accuracy is further improved using Optuna, a Bayesian optimization framework, which adjusts hyperparameters to get better results. Unlike traditional approaches that involve multiple steps, the proposed method directly predicts delay duration in minutes, making the process more straightforward and efficient.

Furthermore, the predictive model is integrated into a Flask- based API, allowing users to input delay-related factors such as weather conditions, distance, and departure times to get real- time delay predictions. This API can be easily added to airline systems and travel applications to provide passengers and airlines with quick and accurate delay information. The ability to predict delays and provide useful information helps airlines make better decisions and improves the travel experience, leading to a more efficient and reliable airline industry in India.

II. RELATED WORKS

A study investigated various classification models, such as Random Forest and Support Vector Machines, to predict flight delays. It conducted extensive experiments to evaluate performance using metrics like accuracy and F1 score, highlighting the effectiveness of these traditional machine learning techniques in providing reliable predictions for flight delays [1]. Machine learning techniques, including decision trees and neural networks, were implemented to predict flight delays. The emphasis was placed on feature selection and model evaluation, demonstrating that careful optimization of input features significantly enhances prediction accuracy and underscoring the importance of identifying relevant features to improve model performance [2]. A deep learning approach utilizing Long Short-Term Memory (LSTM) networks was explored for predicting flight delays. By analyzing complex temporal patterns in historical flight data, the study achieved improved prediction accuracy, demonstrating how LSTM networks can learn from sequential dependencies in the data, making them a valuable tool for capturing the intricacies of flight operations [3]. Aviation delay estimation was performed using deep learning techniques, showcasing various models that effectively predict delays based on numerous influencing factors. The study provided insights into the benefits of using deep learning approaches for accurate delay predictions, con- tributing to the field's understanding of factors affecting flight schedules [4]. Predicting flight delays was approached using time series classification and deep learning regression mod- els. The research employed Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks to extract meaningful features from time series data, showcasing the effectiveness of deep learning in providing accurate predictions in a highly variable environment [5]. Furthermore, Wang and Chen [6] introduced a deep learning framework for flight delay prediction, utilizing graph convolutional networks and multi-task learning to capture spatial dependencies among airports and temporal correlations in flight data, employing a hierarchical graph fusion approach and a dynamic multi-task learning strategy to simultaneously predict arrival and departure delays with superior performance on a large-scale dataset.

TABLE I: Sample of Flight Delay Dataset

From	To	Airline	Departure Delay	Arrival Delay	Distance	Airline Rating	Airport Rating	Weather
DEL	HYD	Air Asia	-1	1	1244	0.5	0.88	Partly cloudy
DEL	HYD	Indigo	-14	-5	1244	0.7	0.88	Partly cloudy
BOM	DEL	Go Air	-14	-8	1138	0.3	0.9	Patchy rain possible
BOM	DEL	Go Air	-14	12	1138	0.3	0.9	Patchy rain possible
BLR	BOM	Air Asia	0	-15	848	0.5	0.9	Moderate or heavy rain shower
DEL	HYD	Vistara	17	14	1244	0.8	0.88	Thundery outbreaks possible
DEL	HYD	Vistara	0	0	1244	0.8	0.88	Thundery outbreaks possible
BLR	BOM	Air Asia	0	0	848	0.5	0.9	Moderate or heavy rain shower

Xu et al. [7] proposed an integrated approach combining a random forest model with 100% accuracy for a 15-minute delay standard and a genetic algorithm to optimize ground service processes, achieving zero delays across all flights by refining service sequences. Xiao et al. [8] developed a method using transfer entropy with a low-dimensional approximation of conditional mutual information to construct delay propagation relation networks, revealing that large airports receive delays while small airports propagate them, with enhanced accuracy in smaller airline systems. Matsuno and Andreeva- Mori [9] analyzed achievable airborne delays through speed control for international arrivals at Tokyo International Airport, demonstrating 2-4 minute delays per 30 minutes of flight time with high compliance and 2-3% fuel savings via statistical simulations. Lastly, Lykou et al. [10] presented a risk-based method to assess interdependencies and congestion delays in the aviation network, using dependency graphs and historical data to identify critical airports and predict delay-prone flight chains, offering strategic insights to mitigate delay impacts.

III. METHODOLOGY

A. *Preparation of Final Dataset*

To prepare the final dataset for research on flight delay prediction, several data preprocessing steps were performed. Initially, missing values were handled using forward fill, and invalid delay values (represented as -1) were replaced with NaN. Rows with missing values in the 'Departure Delay' and 'Arrival Delay' columns were dropped to ensure data completeness.

The dataset, as shown in Table I, contains various flight- related parameters such as departure delay, arrival delay, distance, airline rating, airport rating, and weather conditions. As part of the pre-processing phase, the ’Distance’ column was later categorized into meaningful bins, creating a new ’Dis- tance Category’ feature, and the original ’Distance’ column was removed. Similarly, weather descriptions were simplified by grouping them into broader categories such as ’Cloudy,’ ’Sunny,’ ’Rain,’ and ’Storm’ to reduce complexity.

Finally, categorical features such as ’Airline,’ ’Simplified Weather,’ ’Distance Category,’ ’From,’ and ’To’ were one-hot encoded to convert them into a numerical format suitable for machine learning models. The target variable, ’Arrival Delay,’ was separated, while all feature columns were converted to a numeric format to ensure consistency for model training.
The final dataset consists of 10,500 rows, covering multiple flight delay factors essential for accurate prediction.

B. Preparing a Predictive Model
As shown in Figure 1, the LightGBM algorithm is utilized to build a classification model that predicts whether a flight will be on-time or delayed. LightGBM is highly efficient, especially with large datasets, as it employs leaf-wise growth rather than the traditional level-wise growth of decision trees. Various hyperparameters, including learning rate, maximum depth, and leaf-related parameters such as minimum data in a leaf and lambda L2 regularization, are fine-tuned to enhance the model’s performance. The maximum depth has been limited to 6 to balance complexity and mitigate the risk of overfitting. Setting the minimum data in a leaf to 20 prevents the model from becoming overly specific, thereby avoiding overfitting. The learning rate is set to 0.1, ensuring that the model learns at a moderate pace without overfitting.

LightGBM, like other gradient boosting models, constructs trees sequentially, with each tree designed to reduce the residual errors made by previous trees. It is well-suited for classification tasks, such as predicting flight delays, and can handle large datasets efficiently due to its fast training speed and lower memory usage. LightGBM is particularly advantageous when working with datasets that contain both numerical and categorical features.

Regression Model (Linear Regression Model): The second part of the prediction involves using a multiple linear regression model to predict the exact delay time in minutes. This regression model establishes a linear relationship between several independent variables (flight departure details, weather attributes, distance, etc.) and the dependent variable (flight arrival delay time). The inputs for this model are the output of the LightGBM classifier (indicating whether the flight will be delayed or not) along with the flight and weather data Multiple linear regression is effective in modeling relationships between the independent and dependent variables, where the predicted flight arrival delay time is calculated as a function of several explanatory variables. By combining the classification outcome from the LightGBM model with this regression, the model provides both a binary prediction of delay and a continuous prediction of how long the delay will be.

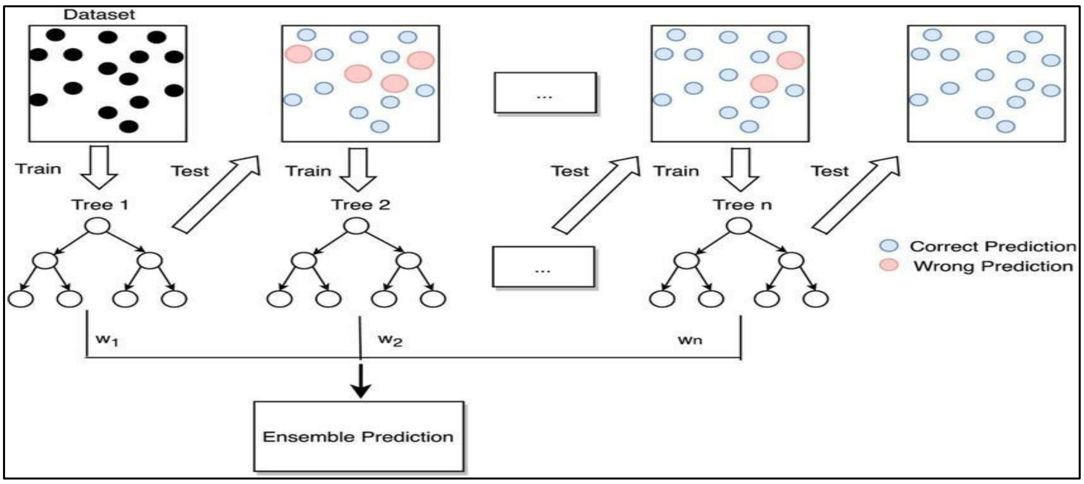

Fig. 1: Ensemble prediction process using multiple decision trees with the LightGBM algorithm, showing training and testing phases with correct and wrong predictions.

C. Data Input to LGBM Regressor

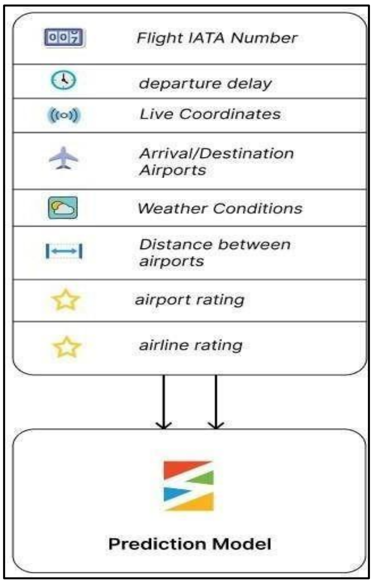

Fig. 2: Data tuples input to the Prediction Model.

The LightGBM (LGBM) regressor uses multiple flight- related parameters to estimate delays with accuracy. The input data is gathered from various sources and processed before being passed to the model. The main input is the Flight IATA Number, which the user provides. Based on this identifier, additional flight details are retrieved from the AviationStack API, including departure delay, live coordinates (if available), arrival and destination airports, distance between airports, airport rating, and airline rating.

Weather conditions play an important role in delay prediction and are obtained from the WeatherAPI. If live coordinates are available, weather data is collected based on the flight's longitude and latitude; otherwise, it is taken from the departure airport. These inputs, covering both real-time and historical data, are then processed and provided to the LGBM Regressor, as shown in Figure 2, which predicts the delay duration.

D. Implementing Flask API

As shown in Figure 3, we implemented a Flask API to integrate the predictive models—LightGBM for classification and linear regression for delay time estimation—into a web service. The Flask API allows external systems to interact with the models by sending flight-related data and receiving real-time predictions of flight delay status and, if applicable, the estimated delay time. The Flask framework was chosen for its lightweight and efficient handling of HTTP requests, making it suitable for creating RESTful APIs. The API accepts input data, which includes flight details such as departure time, distance, and weather conditions, and processes them through the models. The following steps were involved in implementing the Flask API:

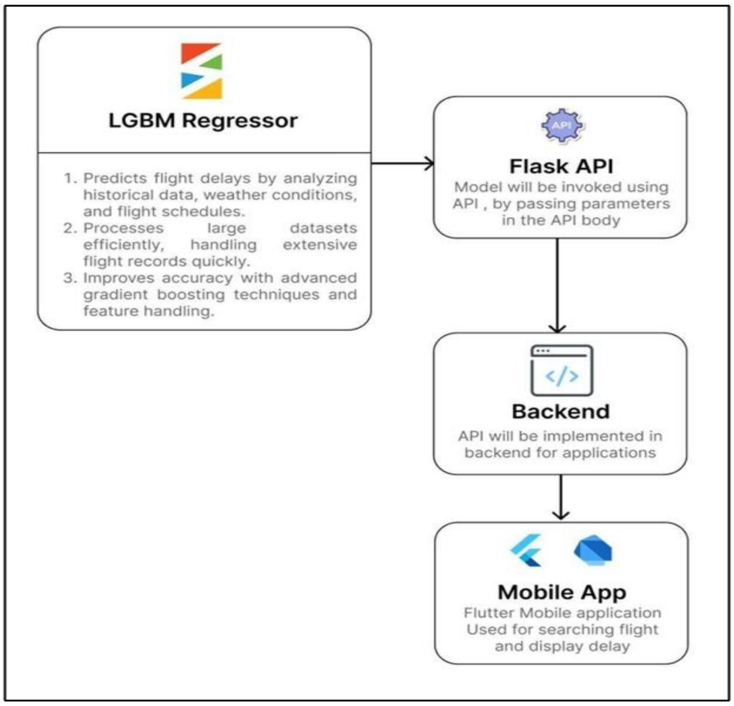

Fig. 3: Implementation of Flask API.

1) *Loading the Model:* The pre-trained LightGBM and linear regression models were loaded using Python's joblib library. These models were trained offline and stored for fast retrieval during API requests. The LightGBM model classifies whether a flight is delayed or on- time, while the linear regression model predicts the delay time for flights classified as delayed.

2) *API Structure:* The Flask API was structured with a single endpoint (/predict), which processes POST requests. Input data is provided in JSON format and consists of the relevant features required for prediction. The API first pro- cesses the input through the LightGBM classifier, and based on the classification result, it either returns a prediction of no delay or, if delayed, passes the data to the linear regression model to predict the delay time in minutes.

3) *Prediction Workflow:* Following are the steps followed for the prediction process.

Step 1: The input data is received and preprocessed into a suitable format for the models.

Step 2: The LightGBM model classifies the flight as either on- time or delayed.

Step 3: If the flight is predicted to be delayed, the linear regression model estimates the delay time. The API then returns both the classification result (on-time or delayed) and the estimated delay time in minutes.

4) *Output Format:* The API returns the output as a JSON response, ensuring that it can easily be integrated into any system requiring delay predictions.

Delay Status: Whether the flight is on-time or delayed. Delay Time: The predicted delay time in minutes, if applicable.

IV. EXPERIMENT AND RESULT

The scatter plot as shown in Figure 4 illustrates the relation- ship between actual and predicted flight arrival delays. Each point represents a flight, with the x-axis showing actual delays and the y-axis showing predicted delays.

Ideally, points should cluster near the diagonal red dashed line, indicating accurate predictions. If the points are widely dispersed or reveal a pattern, it suggests the model may struggle to generalize, leading to inaccurate predictions for certain delay ranges.

A. Implementation of JetLagged Application

Fig. 5: Predict Delay Screen

As shown in Figure 5 the interface includes a prominent input field at the top where users can enter the flight number they wish to track. Below this, a "Predict Arrival Delay" button initiates the prediction process based on the entered flight number. Once the prediction is made, the predicted arrival delay display presents the estimated delay in minutes. At the bottom of the screen, there are two navigation buttons: "Predict" and "Live Track." The user is currently on the prediction screen but can switch to the live tracking screen if needed.

Once the user enters a valid Flight IATA number, the live coordinates will be fetched from the AviationStack API, and

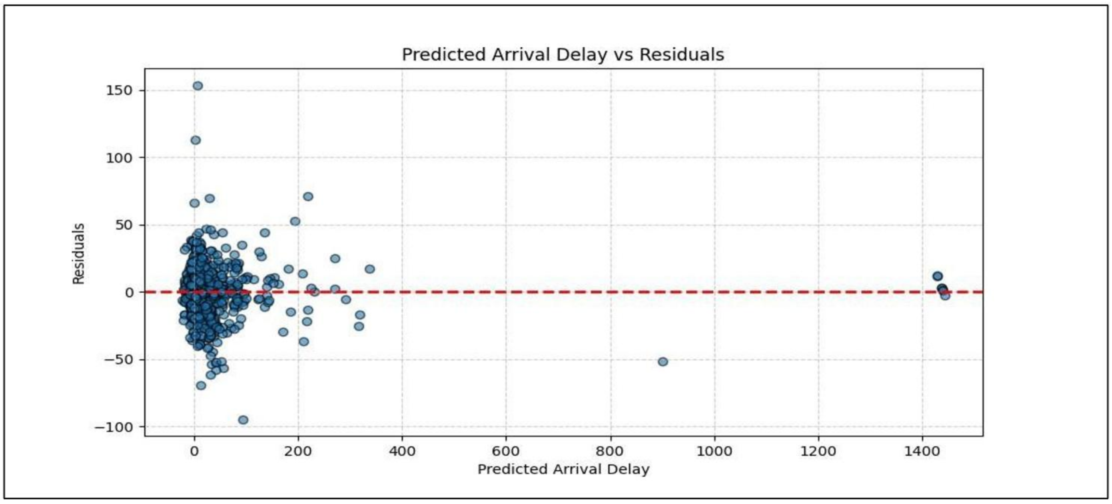

Fig. 4: Scatterplot showing relation between Actual delay and Predicted delay.

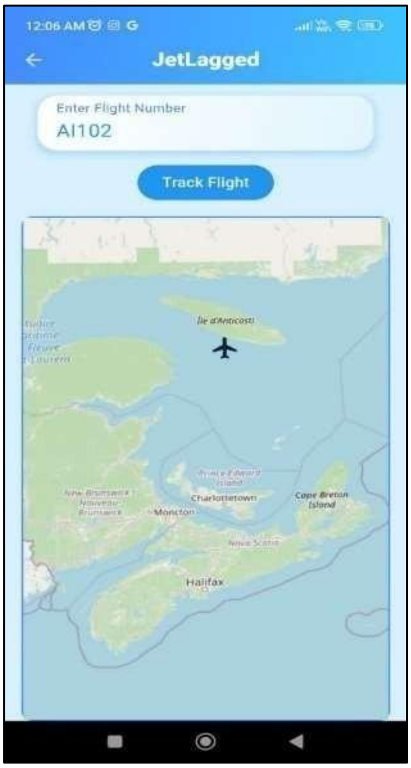

Fig. 6: Plotting of Flight on Map

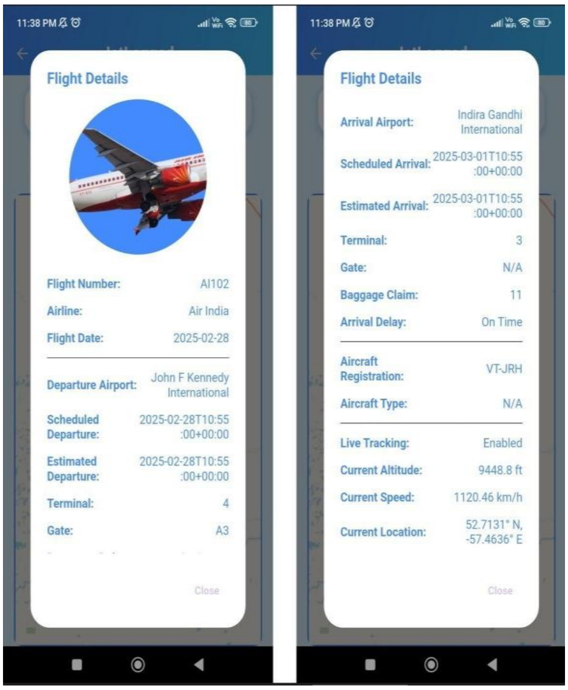

Fig. 7: Fetching Details of Live Flight.

the flight will be plotted on the map as shown in Figure 6. If live details are not available for the entered flight, a snackbar notification will appear, stating ”No Live Data Available.”
After the user clicks on the flight icon, detailed information about the flight is displayed as shown in Figure 7, fetched from AviationStack. This includes crucial details such as the flight number, departure and arrival times, current flight status, airline, aircraft type, and the airport codes for both departure and arrival.

V. CONCLUSION

JetLagged is a helpful system designed to reduce flight delays and cancellations using advanced data analysis. It predicts delays and finds the main reasons for disruptions, benefiting both airlines and passengers. The system is tested for accuracy, performance, ease of use, reliability, and user feedback to make sure it works well. A simple and user-friendly design ensures that both airline staff and passengers can use it easily. The success of Flight Predictor shows how deep learning and data analysis can be useful in real-life situations. By using data from different sources and smart prediction models, it helps airlines run more smoothly, cut financial losses, and improve passenger experience. This project also provides valuable learning experiences by helping developers understand the challenges of building a prediction system and the importance of teamwork. In short, JetLagged proves that data-driven solutions can solve real-world problems effectively.

REFERENCES

[1] P. Meel, M. Singhal, M. Tanwar, and N. Saini, "Predicting flight delays with error calculation using machine learned classifiers," in *2020 7th International Conference on Signal Processing and Integrated Networks (SPIN)*, 2020, pp. 71–76.

[2] R. Khan, S. Akbar, and T. A. Zahed, "Flight delay prediction based on gradient boosting ensemble techniques," in *2022 16th International Conference on Open Source Systems and Technologies (ICOSST)*, Dec 2022, pp. 1–5.

[3] G. Gui, F. Liu, J. Sun, J. Yang, Z. Zhou, and D. Zhao, "Flight delay prediction based on aviation big data and machine learning," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 1, pp. 140–150, Jan 2020.

[4] S. Sharan, M. Sriniketh, H. Vardhan, and D. Jayanth, "State-of-art ma- chine learning techniques to predict airlines delay," in *2021 International Conference on Forensics, Analytics, Big Data, Security (FABS)*, vol. 1, Dec 2021, pp. 1–6.

[5] R. Kothari, R. Kakkar, S. Agrawal, P. Oza, S. Tanwar, B. Jayaswal, R. Sharma, G. Sharma, and P. N. Bokoro, "Selection of best machine learning model to predict delay in passenger airlines," *IEEE Access*, vol. 11, pp. 79 673–79 683, 2023.

[6] T. Wang and S.-C. Chen, "Multi-task local-global graph network for flight delay prediction," in *2022 IEEE 23rd International Conference on Information Reuse and Integration for Data Science (IRI)*, Aug 2022, pp. 49–54.

[7] Z.-T. Xu, Y.-J. Li, H.-F. Zuo, T.-Z. Xu, Q. Wang, W.-W. Yu, H.-S. Yan, B. Liu, T. Chen, and M.-H. Zhou, "Ground handling process optimization model linked to flight delay prediction results," *IEEE Access*, vol. 12, pp. 114 838–114 857, 2024.

[8] Y. Xiao, Y. Zhao, G. Wu, and Y. Jing, "Study on delay propagation relations among airports based on transfer entropy," *IEEE Access*, vol. 8, pp. 97 103–97 113, 2020.

[9] Y. Matsuno and A. Andreeva-Mori, "Analysis of achievable airborne delay and compliance rate by speed control: A case study of international arrivals at tokyo international airport," *IEEE Access*, vol. 8, pp. 90 686– 90 697, 2020.

[10] G. Lykou, P. Dedousis, G. Stergiopoulos, and D. Gritzalis, "Assessing interdependencies and congestion delays in the aviation network," *IEEE Access*, vol. 8, pp. 223 234–223 254, 2020.