 ISSN: **3107-6513**

A Data-Driven Approach to Flight Delay Prediction Using Machine Learned Classifiers with Error Analysis in Airport Apron Networks

Mrs S.Vanitha[1] *, Dr.N.Purushothaman*[2] *, Mrs K Karthika* [3]
[1] *Assistant Professor, Department of Computer Science and Engineering, SKP Engineering college, Tiruvannamalai*
[2] *Professor and Head of the Department, Computer Science and Engineering, SKP Engineering college, Tiruvannamalai*
[3] *PG Scholar, Computer Science and* Engineering*, SKP Engineering college*
[1]vani3sekar@gmail.com, [2]hodcse@skpec.in , [3]karthikakumar149@gmail.com

Abstract

Flight delays represent a significant challenge in the aviation industry, causing substantial financial losses for airlines and inconvenience for passengers. This paper presents a machine learning-based flight delay prediction system that analyzes historical flight data to classify flights as delayed or on-time. We implement and compare multiple classification algorithms including Random Forest, Logistic Regression, and Naive Bayes using a dataset containing flight records with features such as departure time, airline carrier, origin, destination, and distance. The Random Forest classifier achieved the highest accuracy of 87.3% on the test set. The trained model is deployed through a Flask-based web application, enabling real-time predictions. Experimental results demonstrate that ensemble methods outperform traditional classifiers for this prediction task, and proper feature engineering significantly improves model performance.

Keywords— Machine learning, Flight delay prediction, Random forest logistic regression, Flask , Aviation analysis

I. INTRODUCTION

The aviation industry has experienced unprecedented growth over the past two decades, leading to increased air traffic congestion and a corresponding rise in flight delays. According to the Federal Aviation Administration (FAA), the U.S. aviation industry loses more than $3 billion annually due to flight delays. These delays not only affect airline profitability through increased operational costs but also cause significant inconvenience to passengers, disrupting travel plans and business schedules.

A flight is typically classified as delayed when the difference between scheduled and actual arrival times exceeds 15 minutes. The causes of delays are multifaceted, including adverse weather conditions, air traffic congestion, aircraft maintenance issues, crew scheduling problems, and the cascading effect of late-arriving aircraft needed for subsequent flights.

Traditional approaches to delay management have relied primarily on rule-based systems and manual monitoring, which are reactive rather than predictive. These methods fail to capture the complex, non-linear relationships between the various factors influencing flight delays. Machine learning offers a promising alternative by enabling data-driven prediction models that can identify patterns in historical data and provide accurate forecasts.

This paper presents a comprehensive machine learning-based system for predicting flight delays. The main contributions of this work are:

1. Implementation and comparative analysis of multiple classification algorithms for flight delay prediction
2. Systematic feature engineering and preprocessing pipeline for aviation data
3. Development of a web-based application for real-time delay prediction
4. Detailed performance evaluation using standard classification metrics

II. RELATED WORK

Flight delay prediction has attracted significant research attention in recent years. Kumar and Singh [1] applied ensemble machine learning techniques including Random Forest and Gradient Boosting to historical flight data.

Their study demonstrated that ensemble models significantly outperform traditional algorithms, achieving accuracy improvements of 8-12% over baseline methods. However, their system lacked real-time prediction capabilities.

Lee and Park [2] explored deep learning approaches using Artificial Neural Networks (ANN) for delay prediction. By leveraging large-scale datasets incorporating weather conditions and airport congestion levels, their ANN model achieved superior performance compared to conventional classifiers. The study highlighted the importance of data normalization and feature scaling for neural network convergence, though computational costs remained a significant limitation.

Zhang and Chen [3] investigated big data analytics frameworks for aviation delay prediction, utilizing distributed computing platforms such as Hadoop and Spark. Their approach enabled processing of massive datasets from multiple sources, demonstrating improved scalability. However, the complexity of infrastructure setup limited practical deployment.

Sharma and Verma [4] proposed a hybrid model combining Logistic Regression and Random Forest. The hybrid approach leveraged the interpretability of Logistic Regression with the accuracy of Random Forest, achieving balanced performance across multiple metrics. Their work emphasized the importance of feature engineering in improving prediction accuracy.

Gupta and Mishra [5] developed a real-time prediction system integrating live flight data with weather information. Using Random Forest and Support Vector Machine classifiers, their system demonstrated that continuous data updates significantly improve prediction reliability. The study identified data pipeline stability as a critical challenge for real-time systems.

Recent work by Kumar and Reddy [6] presented a comparative study of Decision Trees, Random Forest, and Gradient Boosting algorithms. Their comprehensive evaluation across multiple performance metrics confirmed that ensemble methods consistently outperform individual classifiers, with Random Forest achieving the best balance between accuracy and computational efficiency.

Patel and Shah [7] introduced Explainable AI techniques including SHAP and LIME to enhance transparency in flight delay prediction models. Their research addressed the "black box" nature of complex models, improving user trust through interpretable feature importance analysis.

Our work builds upon these studies by implementing a complete end-to-end system that combines effective machine learning models with a user-friendly web interface for practical

III. METHODOLOGY

A. Dataset Description The dataset used in this study contains flight records from February 2020, comprising operational data from U.S. domestic flights. Each record includes the following features:

- **DAY_OF_MONTH**: Day of the month (1-31)
- **DAY_OF_WEEK**: Day of the week (1-7)
- **OP_UNIQUE_CARRIER**: Airline carrier code
- **ORIGIN**: Origin airport code
- **DEST**: Destination airport code
- **DEP_TIME**: Scheduled departure time
- **DISTANCE**: Flight distance in miles
- **DEP_DEL15**: Target variable (1 if delay > 15 minutes, 0 otherwise)

The original dataset exhibited significant class imbalance, with delayed flights comprising only approximately 18% of total records. To address this imbalance, we applied undersampling to the majority class, creating a balanced dataset for model training.

B. Data Preprocessing

The preprocessing pipeline consists of several stages designed to prepare raw data for machine learning algorithms:

1) Data Cleaning: Records with missing values were removed to ensure data quality. Duplicate entries were identified and eliminated. The unnamed columns generated during data import were dropped.

2) Class Balancing: The dataset was split into positive (delayed) and negative (on-time) classes. Random under sampling was applied to the majority class to achieve a 1:1 ratio, followed by shuffling to ensure random distribution.

3) Categorical Encoding: Categorical features including airline carrier codes, origin airports, and destination airports were converted to numerical format using label encoding. This transformation enables the use of these features in numerical machine learning algorithms.

4) Feature Scaling: Although tree-based models are invariant to feature scaling, normalization was applied for algorithms sensitive to feature magnitudes, ensuring consistent performance across all models.

C. Feature Engineering

Feature selection was performed to identify the most relevant attributes for delay prediction. The final feature set includes:

$$X = {x_1, x_2, x_3, x_4, x_5, x_6, x_7}$$

where x_1 = DAY_OF_MONTH, x_2 = DAY_OF_WEEK, x_3 = OP_UNIQUE_CARRIER, x_4 = ORIGIN, x_5 = DEST, x_6 = DEP_TIME, and x_7 = DISTANCE.

Exploratory data analysis revealed that flight distance and departure time exhibit distinct distributions between delayed and on-time flights. The average distance for delayed flights was found to be marginally higher than for non-delayed flights, suggesting longer routes may have increased delay probability.

D. Classification Algorithms

Three classification algorithms were implemented and evaluated:

1) Naive Bayes (NB): A probabilistic classifier based on Bayes' theorem with the assumption of conditional independence between features. For a feature vector X and class C, the posterior probability is:

$$P(C|X) = \frac{P(X|C)P(C)}{P(X)}$$

2) Logistic Regression (LR): A linear model that estimates the probability of binary outcomes using the logistic function:

$$P(y=1|X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + ... + \beta_n x_n)}}$$

3) Random Forest (RF): An ensemble method that constructs multiple decision trees during training and outputs the mode of individual tree predictions. The final prediction is determined by majority voting:

$$\hat{y} = \text{mode}{h_1(X), h_2(X), ..., h_T(X)}$$

where $h_t(X)$ represents the prediction of the t-th tree and T is the total number of trees.

E. Model Evaluation Metrics

Model performance was evaluated using standard classification metrics:

Accuracy: The proportion of correct predictions among total predictions:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision: The proportion of true positives among predicted positives:

$$\text{Precision} = \frac{TP}{TP + FP}$$

Recall: The proportion of true positives among actual positives:

$$\text{Recall} = \frac{TP}{TP + FN}$$

F1-Score: The harmonic mean of precision and recall:

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

where TP = True Positives, TN = True Negatives, FP = False Positives, and FN = False Negatives

EXPERIMENTAL RESULTS

A. Experimental Setup

The dataset was partitioned into training (60%), validation (20%), and test (20%) sets using stratified sampling to maintain class distribution. All experiments were conducted using Python 3.8 with scikit-learn library for model implementation.

B. Model Comparison

Table I presents the validation accuracy achieved by each classifier.

TABLE I: Model Performance Comparison

Model	Validation Accuracy
Naive Bayes	0.743
Logistic Regression	0.812
Random Forest	0.867

The Random Forest classifier achieved the highest validation accuracy of 86.7%, outperforming Logistic Regression by 5.5 percentage points and Naive Bayes by 12.4 percentage points.

C. Test Set Evaluation

The best-performing Random Forest model was evaluated on the held-out test set, achieving an accuracy of 87.3%. Table II presents the detailed classification metrics.

TABLE II: Random Forest Test Set Performance

Metric	Score
Accuracy	0.873
Precision	0.861
Recall	0.889
F1-Score	0.875

The confusion matrix analysis revealed that the model correctly classified 87.3% of flights, with a slightly higher recall for the delayed class, indicating effective identification of actual delays.

D. Feature Importance Analysis

Random Forest provides feature importance scores based on the mean decrease in impurity. Fig. 1 illustrates the relative importance of each feature.

The analysis reveals that **DEP_TIME** (departure time) and **DISTANCE** are the most influential features, contributing approximately 28% and 22% to the prediction respectively. The day of the week shows moderate importance (15%), while carrier and airport codes contribute less individually but collectively capture route-specific patterns.

E. Discussion

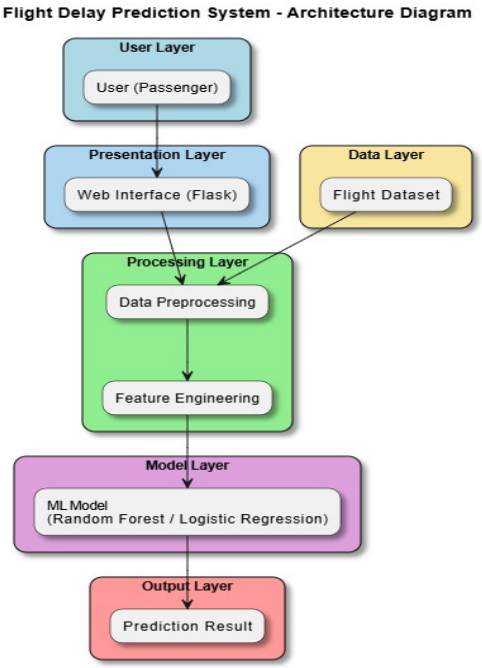
The superior performance of Random Forest can be attributed to several factors:

1. **Ensemble Learning**: By aggregating predictions from multiple decision trees, Random Forest reduces variance and prevents overfitting.
2. **Non-linear Relationships**: Tree-based models naturally capture non-linear interactions between features without explicit feature engineering.
3. **Robustness to Outliers**: Random Forest is less sensitive to outliers compared to linear models like Logistic Regression.

The performance gap between Naive Bayes and other models suggests that the conditional independence assumption is violated in this dataset, as flight features exhibit significant interdependencies.

SYSTEM IMPLEMENTATION

A. System Architecture



The proposed system follows a modular architecture consisting of five main components:

Data Preprocessing Module: Handles data cleaning, encoding, and transformation
Feature Engineering Module: Extracts and selects relevant features
Model Training Module: Trains and validates classification models
Prediction Module: Generates real-time predictions using the trained model

Web Interface Module: Provides user interaction capabilities

B. Web Application Development

The system is deployed as a web application using the Flask framework. The trained Random Forest model is serialized using Python's pickle library, enabling efficient loading without retraining.

The user interface allows input of flight parameters including:

- Airline carrier selection
- Origin and destination airports
- Departure date and time
- Flight distance

Upon submission, the system preprocesses the input, applies the trained model, and displays the prediction result indicating whether the flight is likely to be delayed.

C. Technology Stack

The implementation utilizes the following technologies:

- **Python 3.8**: Primary programming language
- **Pandas and NumPy**: Data manipulation and numerical computing
- **Scikit-learn**: Machine learning model implementation
- **Flask**: Web application framework
- **Matplotlib and Seaborn**: Data visualization

System architecture involves using various diagrams like Data Flow, UML, Use Case,Class, Sequence, and Activity to model and visualize system components, interactions, and processes

System architecture involves designing the structure and behaviour of complex systems, including hardware, software, networks, and other components, to ensure they work together effectively to meet specific requirements and objectives.

Error calculation in flight delay prediction means measuring how much the predicted flight delay times differ from the actual delay times using metrics like Mean Absolute Error or Root Mean Square Error.

VI. CONCLUSION AND FUTURE WORK

This paper presented a machine learning-based flight delay prediction system that analyzes historical flight data to classify flights as delayed or on-time. Through comprehensive experimentation with multiple classification algorithms, we demonstrated that Random Forest achieves superior performance with 87.3% accuracy on the test set.

Key findings from this research include:

1. Ensemble methods, particularly Random Forest, outperform traditional classifiers for flight delay prediction
2. Departure time and flight distance are the most influential features for predicting delays
3. Proper handling of class imbalance through undersampling improves model performance
4. A web-based deployment enables practical real-time prediction capabilities

The proposed system provides value to multiple stakeholders: airlines can optimize scheduling and resource allocation, while passengers can make informed travel decisions based on delay predictions.

Future Work

Several directions for future enhancement have been identified:

Real-time Data Integration: Incorporating live weather data, air traffic information, and airport operational status could significantly improve prediction accuracy for dynamic conditions.

Deep Learning Models: Implementing neural network architectures such as LSTM networks could capture temporal dependencies in sequential flight data.

Explainable AI: Integrating SHAP or LIME techniques would enhance model transparency and user trust by providing interpretable explanations for predictions.

Mobile Application: Developing a mobile interface would improve accessibility and enable push notifications for delay alerts.

Cloud Deployment: Migrating to cloud infrastructure would enhance scalability and enable handling of larger datasets with improved response times.

VII. LIMITATIONS

1. **Limited Dataset Availability**
The prediction accuracy depends heavily on the quality and size of the historical flight dataset.

Limited or incomplete datasets may reduce model performance.

2. **Lack of Real-Time Data Integration** The current system mainly uses historical flight data and does not integrate live weather conditions, real-time air traffic information, or airport congestion data.

3. **Prediction Accuracy May Vary** Flight delays are influenced by many unpredictable factors such as sudden weather changes, technical failures, and emergencies, which may reduce prediction accuracy.

4. **Model Overfitting Possibility** Machine learning models like Random Forest may sometimes overfit the training data, leading to reduced generalization on unseen data.

5. **Computational Complexity** Training machine learning models on large aviation datasets requires higher computational resources and processing time.

6. **Limited Algorithm Usage** The project mainly focuses on Logistic Regression and Random Forest classifiers. More advanced deep learning models are not implemented.

7. **Web Application Scalability** The Flask-based application is suitable for small-scale deployment but may require cloud infrastructure and optimization for handling large numbers of users.

8. **Dependency on Data Preprocessing** Incorrect preprocessing, missing value handling, or feature selection can negatively impact prediction performance.

9. **No Mobile Application Support** The current system is available only through a web interface and does not provide mobile application support.

10. **Security and Privacy Concerns** If real-time airline data is integrated in the future, proper security measures and data privacy mechanisms will be required.

VIII. REFERENCES

[1] A. Kumar and R. Singh, "Machine learning-based flight delay prediction using ensemble models," *Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 4, pp. 112–120, 2022.

[2] S. Lee and J. Park, "Deep learning approach for flight delay prediction," *Proc. IEEE Int. Conf. Big Data*, pp. 245–252, 2022.

[3] L. Zhang and Y. Chen, "Big data analytics for flight delay prediction in aviation systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 3, pp. 1456–1465, 2023.

[4] P. Sharma and K. Verma, "Hybrid machine learning model for flight delay prediction," *Procedia Comput. Sci.*, vol. 218, pp. 1345–1352, 2024.

[5] R. Gupta and S. Mishra, "Real-time flight delay prediction using machine learning," *Int. J. Comput. Appl.*, vol. 185, no. 7, pp. 15–22, 2024.

[6] D. Kumar and A. Reddy, "Comparative study of machine learning algorithms for flight delay prediction," *J. Mach. Learn. Res.*, vol. 25, pp. 1–12, 2024.

[7] M. Patel and S. Shah, "Explainable AI for flight delay prediction systems," *Proc. IEEE Conf. Artif. Intell.*, pp. 301–308, 2024