Hindawi
International Journal of Aerospace Engineering
Volume 2024, Article ID 3385463, 12 pages
https://doi.org/10.1155/2024/3385463

Research Article

Predicting Flight Delays with Machine Learning: A Case Study from Saudi Arabian Airlines

Meshal Alfarhood, Rakan Alotaibi, Bassam Abdulrahim, Ahmad Einieh, Mohammed Almousa, and Abdulrhman Alkhanifer

Department of Computer Science, College of Computer and Information Sciences, King Saud University, P.O. Box 51178, Riyadh 11543, Saudi Arabia

Correspondence should be addressed to Meshal Alfarhood; malf@ksu.edu.sa

Received 19 September 2023; Revised 16 December 2023; Accepted 29 February 2024; Published 15 March 2024

Academic Editor: Fangzhou Fu

Flight delays are a major concern for both travelers and airlines, with significant financial and reputational consequences. Accurately predicting flight delays is crucial for enhancing customer satisfaction and airline revenues. In this paper, we leverage the power of artificial intelligence and machine learning techniques to build a framework for accurately predicting flight delays. To achieve this, we collected flight information from September 2017 to April 2023, along with weather data, and performed extensive feature engineering to extract informative features to train our model. We conduct a comparative analysis of various popular machine learning architectures with distinctive characteristics, aiming to determine their efficacy in achieving optimal accuracy on our newly proposed dataset. Based on our evaluation of various architectures, our findings demonstrate that CatBoost outperformed the others by achieving the highest test accuracy and the lowest error rate in the challenging use case of Saudi Arabia. Moreover, to simulate real-world scenarios, our framework evaluates the best-performing model that has been selected for deployment in a web application, which provides users with the ability to accurately forecast flight delays and offers a user-friendly dashboard with valuable insights and analysis capabilities.

1. Introduction

Air travel is a favored transportation mode for many individuals. However, flight delays represent a significant challenge for both air travelers and airline operators. Flight delays can generate both short-term and long-term negative impacts on airlines, including financial losses and various other issues involving passenger dissatisfaction, reputational harm, and additional crew expenses. According to the FAA/Nextor, the estimated annual costs of delays, including direct costs to airlines and passengers, lost demand, and indirect costs, amounted to 28 billion USD in 2018 [1]. On the other hand, for travelers, flight delays may prompt them to seek alternative airlines or other modes of transportation. Various factors contribute to flight delays, including weather conditions, air traffic congestion, technical difficulties, and connecting flights. As such, it is critical for airline operators to accurately estimate flight delays, as this could help improve customer satisfaction and airline revenues.

Flight delays represent a worldwide challenge, and Saudi Arabia is not immune to this issue. In this work, we take Saudi Arabia as a case study to investigate flight delays, with a specific focus on Saudi's domestic flights. In January 2022, Saudia Airlines and Flynas Airlines recorded on-time departure rates of 67.50% and 54.60%, respectively [2]. The observed ratio of on-time flights to delayed flights in Saudi Arabia is 60% to 40%, which presents a significant challenge in accurately predicting future delays. It is noteworthy that according to the United States Federal Aviation Administration (FAA), a flight is considered not on time (delayed) once the actual departure/landed time exceeds 15 minutes beyond the scheduled departure/landed time [3].

Artificial intelligence (AI) and machine learning (ML) techniques, in particular, have achieved great success in

resolving various real-world problems. They are increasingly being used to tackle the issue of flight delays in the aviation industry. ML-based approaches involve using historical flight data to build predictive models that can predict potential delays. These models can take into account a variety of factors, such as weather patterns, air traffic congestion, and maintenance issues, to generate more accurate predictions.

The primary objective of this paper is to develop a supervised machine learning model capable of accurately classifying flights as either "On-time" or "Delayed." Additionally, in the event that a flight is classified as delayed, the model should accurately predict the duration of the delay in minutes. To achieve this objective, we initially collected data from three distinct sources and applied data mining techniques to preprocess the data. The preprocessing techniques included combining the data from the various sources into a unified dataset, addressing missing values, resolving inconsistencies in the data, and selecting relevant features based on their impact on the classification and regression tasks. Specifically, we collected a comprehensive dataset comprising all domestic flights in Saudi Arabia over the past five years, along with relevant weather data, to facilitate our analysis.

We then investigate various prominent machine learning models including CatBoost [4], XGBoost [5], LightGBM [6], random forest [7], and deep learning models such as multilayer perceptron (MLP) [8] with the aim of assessing their suitability for achieving optimal accuracy in predicting flight delays on our newly proposed dataset. Our analysis indicated that, among the various machine learning architectures evaluated, CatBoost demonstrated the highest test accuracy in classifying flights as either delayed or on time, achieving a score of 76%. Moreover, CatBoost exhibited the lowest error value when predicting the duration of flight delays, achieving a mean absolute error (MAE) of 12.19. These results suggest that CatBoost may be a promising machine learning technique for addressing challenges related to the proposed dataset.

This paper presents three main contributions that significantly advance the field of flight delay prediction:

(i) Comprehensive dataset: we introduce a comprehensive dataset comprising domestic flight records in Saudi Arabia over the past five years. This dataset is derived from three diverse sources, including weather data, resulting in a rich and extensive collection of information for analysis. By incorporating a wide range of variables, we provide a more holistic view of the factors influencing flight delays in the region

(ii) Advanced feature engineering: our study employs extensive feature engineering techniques to extract informative features from the dataset. This process enhances the model's predictive capabilities by capturing the underlying patterns and relationships within the data. Furthermore, we conduct a comparative analysis of various popular machine learning architectures, each possessing unique characteristics. This analysis enables us to identify the most effective architecture for achieving optimal accuracy on the newly proposed dataset

(iii) Practical deployment: to simulate real-world scenarios, we select the best-performing model and deploy it in a user-friendly web application. This application empowers internal users by providing them with the ability to forecast flight delays. By making our solution accessible and practical, we aim to enhance decision-making processes for aviation stakeholders and improve the overall user experience

The remainder of this paper is organized as follows. Section 2 presents an overview of previous works that have utilized machine learning techniques to predict flight delays across various regions around the world. Section 3 outlines our proposed solution for predicting Saudi flight delays. Section 4 includes implementation details and experimental results. The paper concludes in Section 5 with a summary of the findings and a discussion of future research directions.

2. Related Work

Machine learning models are increasingly being used to tackle a range of complex problems across various domains, including image recognition [19], speech recognition [20], natural language processing [21], and other predictive analytics. These models can be broadly categorized into supervised, unsupervised, and reinforcement learning, as well as specialized architectures designed for specific applications. The selection of an appropriate machine learning model depends on various factors, including the nature and size of the dataset, the type of inputs and outputs, and the desired performance metrics [22].

The prediction of flight delays is a significant undertaking in aviation research and related applications. Recent technological advancements have enabled the increasing application of machine learning techniques to this task, resulting in promising outcomes. This section presents a comprehensive overview of prior research on machine learning techniques for predicting flight delays, across diverse regions globally.

To begin with, Alharbi and Prince [9] employed a hybrid approach that utilized machine learning as a data mining tool to predict flight delays using a deep learning classification algorithm. They tested three predictive models: logistic regression, decision tree, and multilayer perceptron (MLP) with principal component analysis (PCA). The authors utilized two sources of data, the General Authority of Civil Aviation (GACA) in Saudi Arabia and the Kaggle dataset. The hybrid model, which is MLP with PCA, achieved the highest testing accuracy of 0.8957 for the GACA dataset and 0.9843 for the Kaggle dataset. While the model demonstrated good performance for Saudi Arabian data, the limited size of the dataset and the relatively small number of features may prevent its ability to generalize well for real-world scenarios.

Furthermore, Khan et al. [10] utilized various machine learning algorithms, including random forest, decision tree,

Naive Bayes, K-nearest neighbor, and multilayer perceptron, to predict flight delays using a publicly available Kaggle dataset of United States domestic air traffic data from 2015. While the authors demonstrated satisfactory results in their work, the dataset that they used only covered a single year. We believe that extending the dataset to encompass multiple years may improve the model's predictive capabilities. In addition, Zhang and Ma [11] presented a flight delay prediction model for forecasting the departure delay of Newark Liberty International Airport using the CatBoost algorithm. They used a dataset containing 226k records and 11 features. The presented model achieved a prediction accuracy of 0.77. Similarly, Ding [12] conducted an evaluation of three algorithms, namely, Naive Bayes, C4.5, and multiple linear regression, to estimate flight delays. The results indicated that the multiple linear regression algorithm achieved a higher accuracy rate of approximately 80% compared to the other two algorithms. Like other previous work, the dataset that they used only covered a single year.

Moreover, Yiu et al. [13] utilized five distinct machine learning algorithms, including decision tree, random forest, K-nearest neighbors, Naive Bayes, and artificial neural networks (ANN), for predicting flight delays. The data utilized in this study comprised flight data from Hong Kong International Airport between March 31, 2018, and April 30, 2018, with parameters such as airline, actual airtime, aircraft size, weekday of departure, weekday of arrival, departure delay status, and arrival delay status. They achieved above 80% accuracy using the ANN. However, the dataset utilized in the study was limited in scope, comprising only one airport's data for a single month. In addition, Tang [14] applied seven machine learning algorithms to a dataset consisting of 28,820 rows of flight data departing from JFK Airport between November 2019 and December 2020. The highest accuracy rate of 97% was achieved using the decision tree algorithm. Additionally, Khan et al. [23] propose a parallel-series model and an adaptive bidirectional extreme learning machine (AB-ELM) method for predicting and analyzing flight delays. The study focuses on understanding the causes of flight delays, particularly the IATA-coded flight delay subcategories, and shows that the proposed methods, along with proper sampling approaches, are effective in uncovering hidden patterns and achieving a high accuracy of 80.66% using Hong Kong's international airlines.

Additionally, Hatıpoğlu et al. [15] applied gradient boosting techniques, specifically XGBoost, LightGBM, and CatBoost, to flight data from a Turkish airline company. The dataset used consisted of only 18,148 international flights. They achieved 96.9% accuracy with the XGBoost algorithm. Moreover, Al-Tabbakh and El-Zahed [16] evaluated eight classification algorithms using the open-source software Weka. The study employed a dataset of 512 records obtained from EGYPTAIR. In addition, Kiliç and Sallan [17] utilized machine learning and artificial intelligence techniques to predict flight delays in the US airport network. Their results demonstrate that the gradient boosting machine model outperformed other models in terms of predictive accuracy, making it an effective solution for predicting arrival flight delays in the US airport network.

Lastly, Birolini and Jacquillat [24] develop predictive and prescriptive analytics models to forecast primary delays and optimize day-ahead aircraft routing, resulting in improved robustness of airline operations and reduced delay costs. The models achieve a mean absolute error of 7-8 minutes in predicting delays and demonstrate the benefits of creating shorter aircraft rotations and strategically allocating schedule slack to mitigate delays.

Table 1 shows a comparison between all the mentioned related works according to the region of the airlines, the selected classifier, the number of obtained features in the dataset, and, lastly, the number of samples in the dataset.

3. Methodology

This section outlines our proposed framework for predicting flight delays. Firstly, we describe the overall pipeline of our approach. Secondly, we detail every step of the pipeline including the Saudi flight dataset that we gathered for our experiments, documenting the processes of data preprocessing, feature engineering, and dataset partitioning. Finally, we present our web application, which simulates real-world scenarios for users by providing them with forecasted information regarding the status of their future domestic flights in Saudi Arabia.

3.1. Overall Pipeline. The overall pipeline adopted in our paper involves a series of sequential steps to construct the overall framework including data collection, data integration, data preprocessing, feature engineering, data splitting, model training, model evaluation, and choosing the best-performing model. Figure 1 illustrates the complete pipeline. Each step in the pipeline is explained thoroughly in the following sections.

3.1.1. Saudi Flight Dataset. As previously stated, the scope of this paper is to investigate the issue of flight delays in Saudi Arabia. To the best of our knowledge, no prior research has been conducted on this scope, and there is currently no publicly available dataset suitable for exploration. Consequently, we collected our own dataset consisting of all domestic flights in Saudi Arabia over the last five years, totaling 775,000 domestic flights. The collected flights cover the period from September 2017 to April 2023. We collect the data from three different sources, as follows:

(i) Flightradar24 [25]: flight data was collected for this study from the Flightradar24 website, spanning from September 2019 to April 2023, comprising 469k samples. The collected data consisted of various features, including flight number, origin airport, destination airport, flight date, scheduled time of departure, scheduled time of arrival, actual time of departure, actual time of arrival, flight status (landed, diverted, canceled), aircraft type, and tail number

(ii) FlightEra [26]: flight data spanning from September 2017 to August 2019, comprising 306k samples, were collected for this study from the FlightEra

Table 1: Comparative study among the mentioned related work in terms of the region, the classifier, the number of features, and the size of the dataset.

	Country	Algorithm	#features	#samples
[9]	Saudi Arabia	MLP with PCA	14	15,668
[10]	United States	MLP	19	3,000,000
[11]	United States	Categorical Boosting	11	226,234
[12]	China	Multiple linear regression	25	100,000
[13]	Hong Kong	Artificial neural networks	7	25,074
[14]	United States	Decision tree	23	28,820
[15]	Turkey	XGBoost	19	18,148
[16]	Egypt	Rules.PART	9	512
[17]	United States	Gradient boosting machine	36	5.6 million
[18]	China	CatBoost	12	25,000

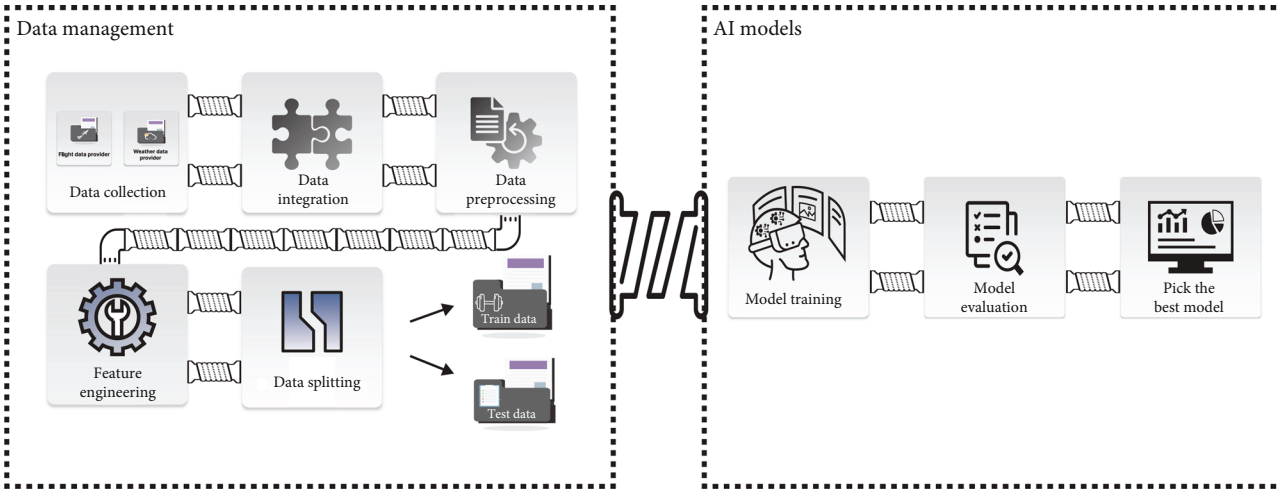


Figure 1: The overall employed methodology entails a series of sequential stages aimed at constructing the flight delay prediction framework.

website and included the same features collected from the Flightradar24 website

(iii) Visual Crossing [27]: weather data ranging from September 2017 to April 2023 were obtained from the Visual Crossing website. The obtained data consisted of multiple features, including temperature, humidity, dew point, precipitation, snow, wind gust, wind speed, wind direction, pressure, visibility, cloud cover, and condition

Ultimately, the data collected from the aforementioned sources were merged to create a final dataset comprising 775k data samples and 22 features.

3.1.2. Data Preprocessing and Feature Engineering. We preprocessed our original dataset such that we removed the samples that miss important features, such as origin airport, destination airport, scheduled time of departure, actual time of departure, and scheduled time of arrival, accounting for around 1% of the original 775k dataset. Moreover, any data samples with origin or destination airports outside of the 27 Saudi airports were also removed. Instances with the same airport as both origin and destination were also excluded as they represented a small percentage of the data. Regarding weather features, we observed that the wind gust and pressure features had a significant number of null values, accounting for 70% and 40%, respectively. Given their low correlation and importance, we decided to remove these features from the dataset. Notably, no significant outliers were detected in the dataset. Additionally, we ensured that all times were valid and correctly formatted, with zero-padding hours to align with the Python datetime format.

We add more features to our dataset via feature engineering. Feature engineering refers to the process of transforming raw data into meaningful information (i.e., more features) that can be used to improve the framework's predictive performance. As a result, in this phase, we extracted the airline feature from the flight ID and calculated the straight-line distance between the origin and destination airports, adding distance as a feature. To enhance time-related features, we replaced the date feature with year, month, and

day of the month. Subsequently, we extracted the day of the week and day of the year from these features and computed the Hijri date from the Gregorian date, creating the features Hijri year, Hijri month, Hijri day of the month, and Hijri day of the year. Hijri date refers to the Islamic calendar system, which is a lunar calendar consisting of 12 months in a year of 354 or 355 days. The start of each month is based on the sighting of the new moon. The Hijri calendar is commonly used in Islamic countries and is also used to determine important Islamic events and holidays.

Moreover, to account for instances where airport congestion causes flight delays, we incorporated features to identify special days in the year that might cause an increase in passenger traffic. Thus, we add the following features: Is_weekend, Is_vacation, Is_holiday, and Is_Hajj. Also, we add other features such as the number of flights on the same day and the rate of delays for the same flight in the previous month.

As a result, the final preprocessed dataset comprised of 765k data samples and 38 features. Table 2 shows the final features that we use in this paper including the engineered features.

On the other hand, the data-splitting process is very important for our evaluations, and thus, we have meticulously crafted a well-designed split, carefully considering the dataset's size and the date information. The data-splitting process is aimed at dividing the available data into subsets that can be used for training, validation, and testing purposes. The validation subset helps to evaluate the performance of the model during the training phase. Therefore, we split our data into a ratio of 90/5/5, whereby 90% of the data was used for training, 5% was used for validation, and 5% was used for testing. Furthermore, we have ensured that each class, year, and month is proportionally represented in each subset, as this can facilitate the training of a model on a diverse and comprehensive dataset, reflective of the underlying patterns and trends present in the data over time. This approach optimizes the utilization of the considerable amount of data available to us, while guaranteeing that the data is equally distributed across the various subsets.

3.2. Air-Aware App. We have developed a user-friendly web application that utilizes our overall pipeline to predict flight delays and ultimately serves as a real-scenario case. The application entails an accurate flight status forecasting system that is designed to enhance the user experience. Figure 2 shows samples of the implemented web application. It shows our developed user-friendly web application that leverages machine learning algorithms to accurately predict domestic flight delays in Saudi Arabia. It also shows our implemented dashboard that provides a detailed analysis of the Saudi flight dataset, presenting various visualizations and insights to facilitate easy interpretation and decision-making.

4. Experiments

This section presents the experiments conducted to evaluate the performance of our proposed framework for predicting flight delays. We begin by describing the evaluation metrics used to assess the performance of machine learning models. We then present the experimental settings, which include the tools that we use in our implementation process. After that, we present the experimental results obtained from the evaluation of our approach, highlighting the performance of our models in predicting flight delays. Overall, the experiments are aimed at demonstrating the effectiveness of our framework in enhancing the accuracy of flight status forecasting and improving the overall user experience. Finally, we show our analysis of feature importance in the model for predicting flight delays.

4.1. Evaluation Metrics. The present study incorporates both classification and regression models in its analysis. As these models are designed with distinct objectives and performance measures, it is essential to employ appropriate evaluation metrics for each modeling paradigm. The use of appropriate evaluation metrics is critical for accurate model performance assessment, the identification of areas for improvement, and informed decision-making regarding model selection and deployment.

In order to evaluate the performance of our classification model, We adopt five evaluation metrics for the classification part: accuracy, recall, precision, $F1$-score, and area under the ROC curve (AUC), which provide insights into the model's ability to correctly classify instances into different categories. Details of these metrics are described below.

First, the recall is the ratio of the true positives to the sum of the true positives and false negatives:

$$\text{recall} = \frac{\text{tp}}{\text{tp} + \text{fn}}. \tag{1}$$

Second, the precision is the ratio of the true positives to the sum of the true positives and false positives:

$$\text{precision} = \frac{\text{tp}}{\text{tp} + \text{fp}}. \tag{2}$$

Third, the accuracy is the ratio of the number of correct predictions to the total number of predictions:

$$\text{accuracy} = \frac{\text{tp} + \text{tn}}{\text{tp} + \text{tn} + \text{fp} + \text{fn}}. \tag{3}$$

Fourth, the $F1$-score is the harmonic mean of precision and recall:

$$F1 - \text{score} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}, \tag{4}$$

where tp represents the number of true positives, tn represents the number of true negatives, fp represents the number of false positives, and fn represents the number of false negatives.

Fifth, AUC represents the area under the receiver operating characteristic (ROC) curve. The ROC curve plots the true positive rate against the false positive rate at various

Table 2: The list of all the features included in the preprocessed dataset.

Feature name	Description	Data type
Airline	The airline company code	String
Flight	The flight number	Integer
Origin	The IATA code that represents the airport of the flight's departure	String
Destination	The IATA code that represents the airport of the flight's arrival	String
STD	The scheduled departure time of the flight, expressed in minutes of the day	Integer
ETA	The scheduled arrival time of the flight, expressed in minutes of the day	Integer
Distance	The straight-line distance from the origin airport to the destination airport	Float
Year	The Georgian year	Integer
Hijri_year	The Hijri year	Integer
Month	The Georgian month	Integer
Hijri_month	The Hijri month	Integer
Day	The Georgian day of year	Integer
Hijri_day	The Hijri day of year	Integer
DOW	The day of week	Integer
Is_weekend	Whether the departure date is a weekend	Integer
Is_vacation	Whether the departure date is a school vacation	Integer
Is_holiday	Whether the departure date is a national or Islamic holiday	Integer
Is_Hajj	Whether the departure date is within Hajj days and departs from or arrives at either Jeddah, Medina, or Taif airports	Integer
Duration	The expected flight duration	Integer
Timestamp	The timestamp of the departure date and time	Integer
#flights	The number of scheduled flights on the same day	Integer
Delay_rate	The frequency of flight delays for the same flight number during the previous month	Float
Org_temp	The temperature in the origin airport	Float
Org_dew	The dew point in the origin airport	Float
Org_hum	The humidity in the origin airport	Float
Org_percip	The precipitation in the origin airport	Float
Org_vis	The visibility in the origin airport	Float
Org_wspeed	The wind speed in the origin airport	Float
Org_cloud	The cloud cover in the origin airport	Float
Org_cond	The weather condition in the origin airport	Integer
Des_temp	The temperature in the destination airport	Float
Des_dew	The dew point in the destination airport	Float
Des_hum	The humidity in the destination airport	Float
Des_percip	The precipitation in the destination airport	Float
Des_vis	The visibility in the destination airport	Float
Des_wspeed	The wind speed in the destination airport	Float
Des_cloud	The cloud cover in the destination airport	Float
Des_cond	The weather condition in the destination airport	Integer

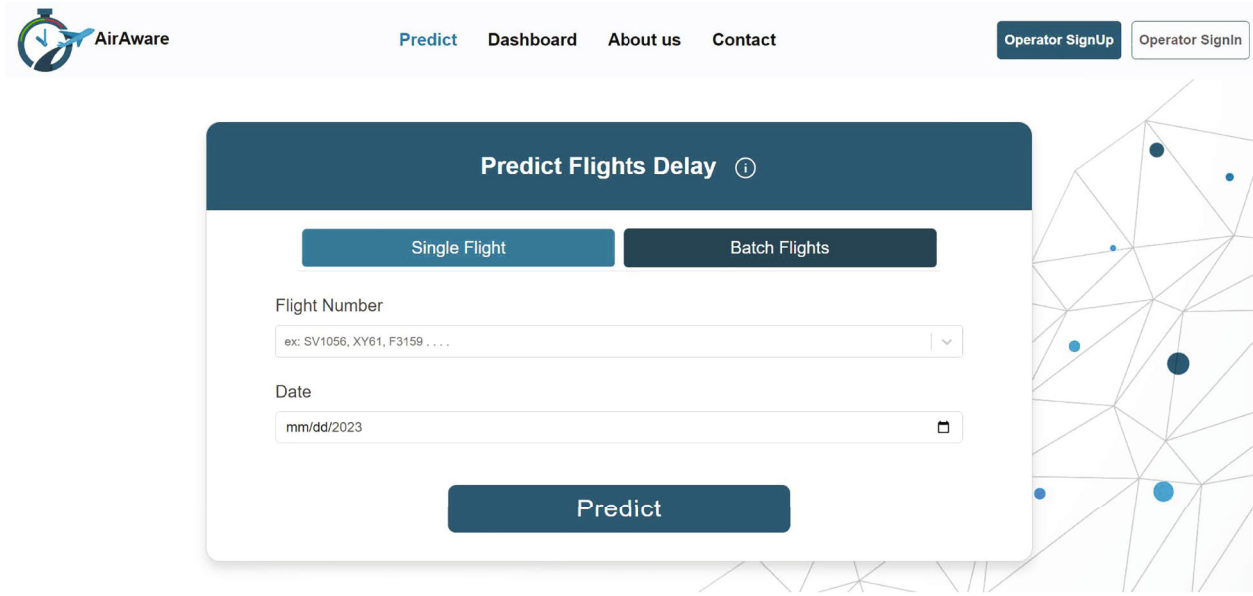
classification thresholds. AUC provides a measure of the model's ability to discriminate between positive and negative instances, with a higher AUC indicating better performance.

On the other hand, in order to evaluate the performance of our regression model, we also adopt three evaluation metrics for the regression part: the mean absolute error (MAE), the mean squared error (MSE), and the root mean squared error (RMSE), which provide a measure of the model's ability to accurately predict continuous values (i.e., flight delay time in minutes). Details of these metrics are described below.

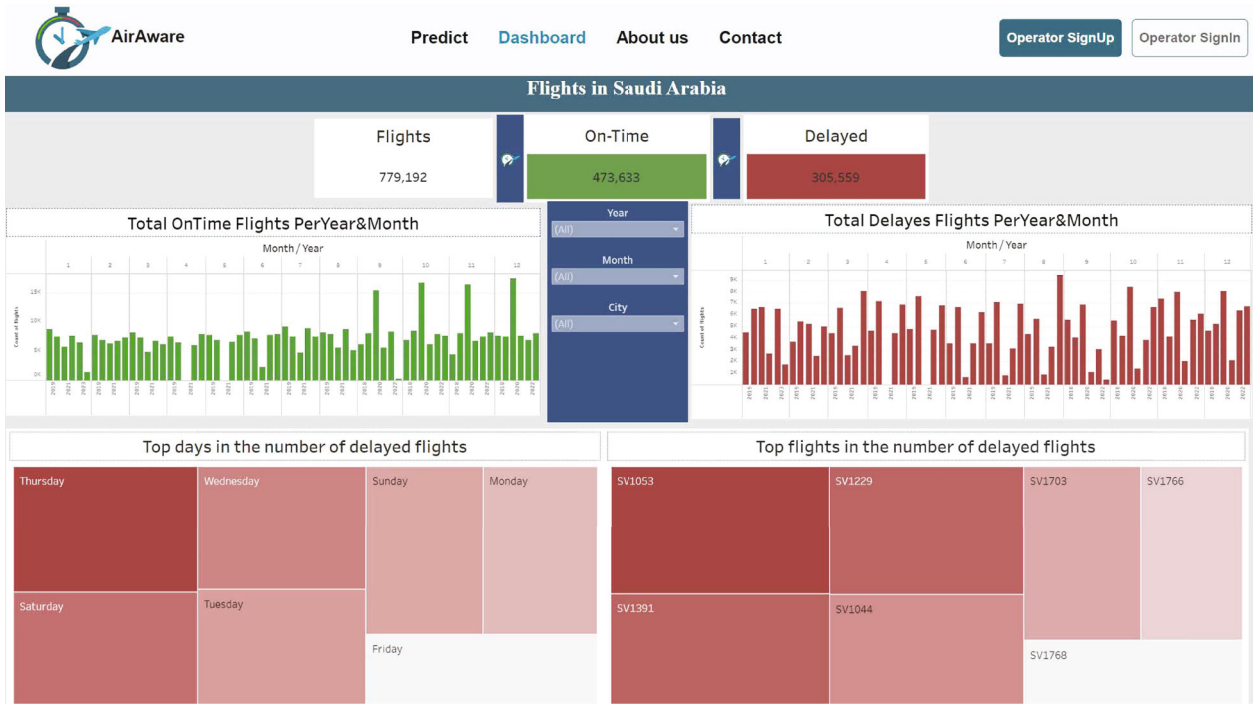
First, the mean absolute error (MAE) is calculated as

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \widehat{y}_i|, \tag{5}$$

where n is the number of samples, y_i represents the true value of the target variable for the i-th sample, and $\widehat{y}_i$ represents the predicted value of the target variable for the i-th sample.



(a) The user-friendly web application



(b) The dashboard

FIGURE 2: (a) The user-friendly web application that we have implemented to predict domestic flight delays in Saudi Arabia. (b) We have also provided a dashboard with a detailed analysis of the data in the Saudi flight dataset, presenting various visualizations and insights.

Second, the mean squared error (MSE) is calculated as

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2, \tag{6}$$

where n is the number of samples, y_i represents the true value of the target variable for the i-th sample, and $\hat{y}_i$ represents the predicted value of the target variable for the i-th sample.

Third, the root mean squared error (RMSE) is calculated as

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}, \tag{7}$$

where n is the number of samples, y_i represents the true value of the target variable for the i-th sample, and $\widehat{y}_i$ represents the predicted value of the target variable for the i-th sample.

4.2. Experimental Setup. In order to conduct our experimental study and evaluate the proposed approach, we utilized Python programming language for implementation purposes. Specifically, we leveraged the Selenium library to collect our flight data from relevant websites. Additionally, we employed the TensorFlow [28] library to develop and implement our multilayer perceptron (MLP) model, while the scikit-learn library [29] was utilized for implementing other models such as CatBoost and XGBoost. For training our models, we utilized Colab [30], a cloud-based platform that provides free access to GPUs, with the following configurations: GPU-T4 and 12.7 GB of RAM. The use of Colab ensured that our models were trained in an efficient and timely manner. Overall, the experimental settings allowed us to effectively evaluate the performance of our proposed framework and choose the best-performing model.

4.3. Experimental Results. This section presents the experimental results of our study. Specifically, we focus on the outcomes of two distinct parts of our approach: the classification part, which predicts whether a flight will be on time or delayed, and the regression part, which forecasts the delay duration in minutes if a flight is delayed. The following subsections provide detailed analyses of the results obtained from each of these parts.

4.3.1. The Classification Model. This section shows the classification of domestic flights in Saudi Arabia by our multiple models, distinguishing them as either on time or delayed. It is worth noting that based on our observation, the ratio of on-time flights to delayed flights stands at 60% to 40%. This finding underscores the considerable difficulty inherent in accurately forecasting future delays. Table 3 presents a comparison of the performance results obtained by various machine learning models on the test dataset for our classification task. The models evaluated include CatBoost, XGBoost, LightGBM, MLP, and random forests. The metrics used to evaluate the models' performance include accuracy, recall, precision, $F1$-score, and AUC.

From the table, it is evident that CatBoost achieved the highest accuracy score of 76%, followed closely by XGBoost and LightGBM, which scored 73.1% and 73.2%, respectively. In terms of recall, CatBoost again outperformed the other models, achieving a score of 74.8%, while XGBoost and LightGBM scored 71.6% and 71.7%, respectively. CatBoost also scored the highest precision score of 75.6%, followed by XGBoost and LightGBM with 72.6%. However, the $F1$-score metric indicated that CatBoost was the best-performing model with a score of 75.1%, while the other models scored below 72%.

Overall, the results suggest that CatBoost is the best-performing model, outperforming the other models in most of the evaluation metrics. However, it is worth noting that the differences in performance between the models are not significant, with most models scoring within a 2-3% range of each other. These findings can be used to guide the selection of an appropriate machine learning model for predicting flight delays.

Table 3: Comparison of performance results of different ML-based models on our test dataset for our classification task.

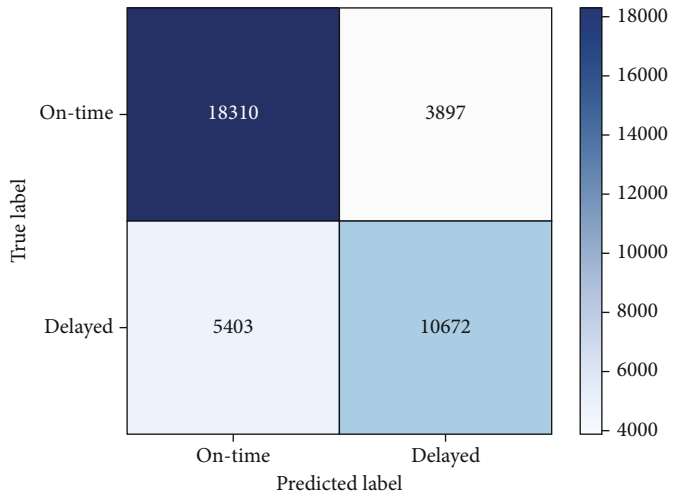
	CatBoost	XGBoost	LightGBM	MLP	Random forests
Accuracy	76	73.1	73.2	72.5	71.3
Recall	74.8	71.6	71.7	71.1	69.8
Precision	75.6	72.6	72.6	71.8	70.6
$F1$-score	75.1	71.9	72	71.4	70.1
AUC	0.74	0.72	0.72	0.71	0.7

4.3.2. Confusion Matrix Comparison. Confusion matrices provide a visual representation of the performance of each model in terms of correctly and incorrectly classified instances across different classes. By comparing the patterns and distributions in the confusion matrices, we can gain insights into the strengths and weaknesses of each model in predicting different classes.

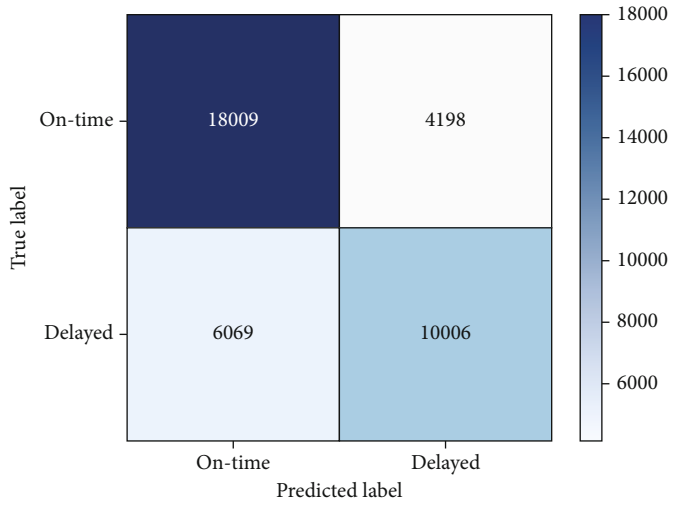
Figure 3 displays the confusion matrices for each model, where the rows in each figure represent the actual classes and the columns represent the predicted classes. The values in the table represent the count of instances that fall into each combination of actual and predicted classes. Once again, CatBoost demonstrates superior performance based on the confusion matrix, with XGBoost and MLP following closely behind. These findings are consistent with the previous results.

4.3.3. The Regression Model. Table 4 presents the performance comparison results of various machine learning (ML) models, namely, CatBoost, XGBoost, LightGBM, and MLP, for the regression task in our study. The evaluation metrics used for the comparison are mean absolute error (MAE), mean squared error (MSE), and root mean squared error (RMSE), which are widely used measures for evaluating regression models. The results indicate that CatBoost achieved the lowest MAE (12.19), followed closely by XGBoost (12.45), LightGBM (12.83), and MLP (12.8). Similarly, CatBoost also obtained the lowest MSE (605.61) and RMSE (24.6) values, outperforming the other models. In contrast, MLP had the highest MSE (657.4) and RMSE (25.64) values among the four models. Overall, the results suggest that CatBoost, again, outperforms the other models in terms of MAE, MSE, and RMSE and is the most suitable ML model for our task.

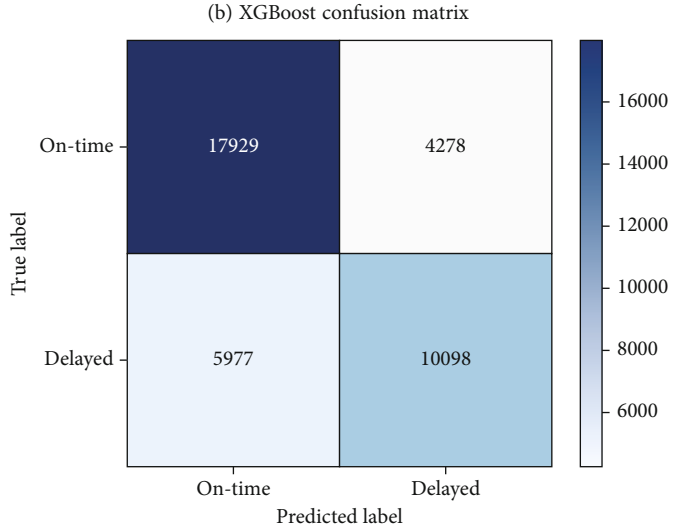
4.3.4. Feature Importance. Feature importance in the CatBoost algorithm refers to a metric that quantifies the relative contribution of each feature in predicting the target variable. It helps identify the most influential features that significantly affect the model's performance and prediction outcomes. The feature importance in CatBoost is typically calculated based on the frequency and magnitude of feature



(a) CatBoost confusion matrix

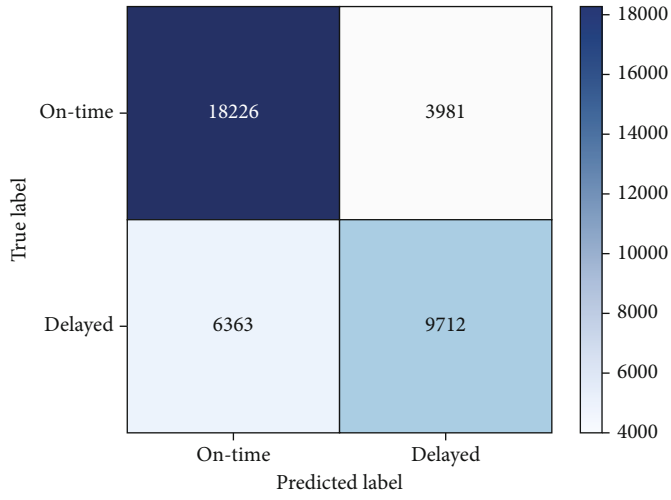


(b) XGBoost confusion matrix

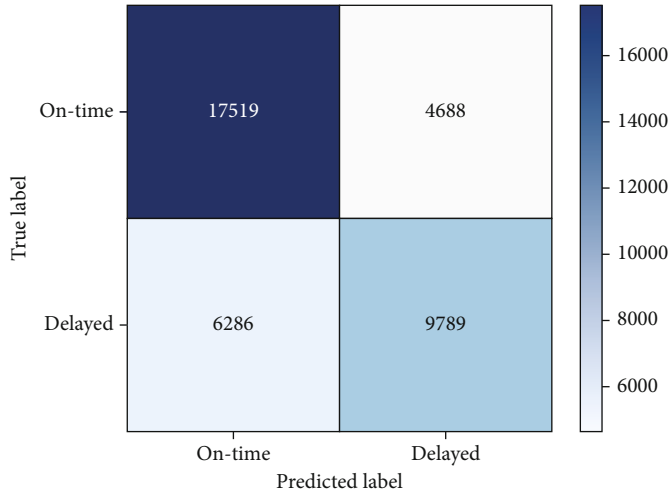


(c) LightGBM confusion matrix

Figure 3: Continued.



(d) MLP confusion matrix



(e) Random forest confusion matrix

Figure 3: The results of the confusion matrices, which provide a visual representation of the performance of each model in terms of correctly and incorrectly classified instances across our different classes (on time and delayed).

Table 4: Comparison of performance results of different ML-based models on our test dataset for our regression task.

	CatBoost	XGBoost	LightGBM	MLP
MAE	12.19	12.45	12.83	12.8
MSE	605.61	618.31	635.82	657.4
RMSE	24.6	24.86	25.21	25.64

usage during the training process. Higher feature importance values indicate greater relevance and impact on the model's predictions, while lower values suggest lesser significance. This information aids in understanding which features are driving the model's decision-making process and can assist in feature selection, interpretation, and improving the overall model performance.

The analysis of feature importance in our machine learning model for predicting flight delays reveals valuable insights. Among the top 7 features deemed most influential are flight delay rate in the previous month, flight number, flight duration, the Georgian day of the year, timestamp, the Hijri day of the year, and the scheduled hour of departure. Notably, our findings indicate that weather data does not rank among the top 20% of influential features that significantly impact the model's performance. This observation may be attributed to the prevailing climatic conditions in Saudi Arabia, characterized by infrequent occurrences of extreme weather phenomena such as snow, heavy rain, and tornadoes, thereby minimizing the weather's influence on flight delays.

5. Conclusion

Flight delays have become a widespread issue that affects both travelers and operators, yet there is a lack of research related to the use of machine learning (ML) techniques to solve this problem. Overall, ML has the potential to revolutionize the aviation industry and help address the challenge of flight

delays in a more proactive and effective way. Therefore, we propose a ML-based system that predicts flight delays for domestic flights in Saudi Arabia. We carefully investigate various machine learning models and deploy the most accurate one into our proposed framework. The system employs flight and weather data as inputs to generate predictions, which were initially collected for the purpose of this work.

We believe that our proposed approach has the potential to make a positive impact on the aviation industry. By reducing flight delays, this work is aimed at improving the travel experience for passengers and increasing the efficiency of airline operations. Furthermore, we put significant effort into collecting and cleaning the data from various sources, which ultimately facilitates the path for future researchers working on similar problems.

Our framework can practically assist airlines and other aviation stakeholders in their decision-making processes, specifically in the following points:

(i) Decision support: our model provides accurate predictions of key aviation metrics such as flight delays. This information can be utilized by airlines to optimize their operational planning, including crew scheduling, resource allocation, and maintenance activities. Additionally, airport authorities can use these predictions to efficiently manage ground operations, gate assignment, and capacity planning

(ii) Proactive measures: By predicting potential delays or disruptions in advance, our model enables airlines to take proactive measures to minimize the impact on passengers and operations. This includes providing early notifications to affected passengers, rebooking or rescheduling flights, and implementing contingency plans to mitigate the consequences of delays or cancellations

(iii) Customer experience: by reducing flight delays and cancellations, our model contributes to enhancing the overall customer experience. This fosters customer loyalty, improves customer satisfaction ratings, and positively impacts an airline's reputation in the highly competitive aviation industry

Several potential directions for extending this work may be considered. One such direction involves collecting more flight data from additional years, potentially spanning over a decade, to enhance the accuracy and robustness of the trained model. Furthermore, another direction involves gathering additional features from airlines and airports, such as maintenance records, runway congestion, and staffing records, to further enhance the model's accuracy. Finally, expanding the scope of the study to encompass international flights, including those in other countries, may improve the model's generalization and scalability.

Data Availability

The data presented in this study is available on request from the corresponding author.

Conflicts of Interest

The authors declare no conflict of interest.

Acknowledgments

This research was funded by the Deputyship for Research and Innovation, Ministry of Education in Saudi Arabia (grant number IFKSUOR3–031–2).

References

[1] "U.S. passenger carrier delay costs," July 2. 2023. https://www.airlines.org/dataset/u-s-passenger-carrier-delay-costs.

[2] "Monthly on-time performance data," OAG, 31 Jan. 2022. https://www.oag.com/en/on-time-performance-airlines-janaury-2022.

[3] "Bureau of Transportation Statistics," 1 July 2023. https://www.transtats.bts.gov/OT_Delay/OT_DelayCause1.asp.

[4] A. V. Dorogush, V. Ershov, and A. Gulin, "CatBoost: gradient boosting with categorical features support," 2018, https://arxiv.org/abs/1810.11363.

[5] T. Chen and C. Guestrin, "XGBoost: a scalable tree boosting system," in *In proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 785–794, ACM, New York, NY, USA, 2016.

[6] G. Ke, Q. Meng, T. Finley et al., "Lightgbm: a highly efficient gradient boosting decision tree," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[7] T. K. Ho, "Random decision forests," in *Proceedings of 3rd international conference on document analysis and recognition*, Montreal, QC, Canada, 1995.

[8] S. Haykin, *Neural Networks: A Comprehensive Foundation*, Prentice Hall PTR, 1998.

[9] B. Alharbi and M. Prince, "A hybrid artificial intelligence approach to predict flight delay," *International Journal of Engineering Research & Technology*, vol. 13, no. 4, pp. 814–822, 2020.

[10] M. Khan, M. Uddin, M. Sarshaar, and J. Hashmi, "Flight delay prediction using machine learning," *Journal of Engineering Sciences*, vol. 13, no. 5, pp. 254–261, 2022.

[11] B. Zhang and D. Ma, "Flight delay prediciton at an airport using maching learning," in *2020 5th International Conference on Electromechanical Control Technology and Transportation (ICECTT)*, pp. 557–560, IEEE, Nanchang, China, 2020.

[12] Y. Ding, "Predicting flight delay based on multiple linear regression," *In IOP Conference Series: Earth and Environmental Science*, vol. 81, no. 1, article 012198, 2017.

[13] C. Y. Yiu, K. K. Ng, K. C. Kwok, W. T. Lee, and H. T. Mo, "Flight delay predictions and the study of its causal factors using machine learning algorithms," in *2021 IEEE 3rd International Conference on Civil Aviation Safety and Information Technology (ICCASIT)*, pp. 179–183, IEEE, Changsha, China, 2021.

[14] Y. Tang, "Airline flight delay prediction using machine learning models," in *ICEBI '21: Proceedings of the 2021 5th International Conference on E-Business and Internet*, pp. 151–154, New York, 2021.

[15] I. Hatıpoğlu, Ö. Tosun, and N. Tosun, "Flight delay prediction based with machine learning," *LogForum*, vol. 18, no. 1, pp. 97–107, 2022.

[16] S. M. Al-Tabbakh and H. El-Zahed, "Machine learning techniques for analysis of Egyptian flight delay," *Journal of Scientific Research in Science*, vol. 35, Part 1, pp. 390–399, 2018.

[17] K. Kiliç and J. M. Sallan, "Study of delay prediction in the US airport network," *Aerospace*, vol. 10, no. 4, p. 342, 2023.

[18] Z. Zhao, S. Feng, M. Song, and Q. Liang, "A delay prediction method for the whole process of transit flight," *Aerospace*, vol. 9, no. 11, p. 645, 2022.

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[20] A. Hannun, C. Case, J. Casper et al., "Deep speech: scaling up end-to-end speech recognition," 2014, https://arxiv.org/abs/1412.5567.

[21] A. Vaswani, N. Shazeer, N. Parmar et al., "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[22] Y. Bengio, I. Goodfellow, and A. Courville, *Deep Learning*, vol. 1, MIT press, Cambridge, MA, USA, 2017.

[23] W. A. Khan, S. H. Chung, A. E. Eltoukhy, and F. Khurshid, "A novel parallel series data-driven model for IATA-coded flight delays prediction and features analysis," *Journal of Air Transport Management*, vol. 114, article 102488, 2024.

[24] S. Birolini and A. Jacquillat, "Day-ahead aircraft routing with data-driven primary delay predictions," *European Journal of Operational Research*, vol. 310, no. 1, pp. 379–396, 2023.

[25] "Flightradar 24," https://www.flightradar24.com (accessed on 5 July 2023).

[26] "FlightEra," https://www.flightera.net (accessed on 5 July 2023).

[27] "Visual Crossing Corporation," https://www.visualcrossing.com/weather-data (accessed on 5 July 2023).

[28] M. Abadi, P. Barham, J. Chen et al., "TensorFlow: a system for large-scale machine learning," in *12th USENIX symposium on operating systems design and implementation (OSDI 16)*, pp. 265–283, Savannah, GA, USA, 2016.

[29] F. Pedregosa, G. Varoquaux, A. Gramfort et al., "Scikit-learn: machine learning in Python," *The Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[30] "Google Colab," https://colab.research.google.com. (accessed on 5 July 2023).