FLIGHT DELAYS PREDICTION MODEL USING MACHINE LEARNING

SITI NUR ELISYA AQMAR BINTI MOHAMAD KAMAL

UNIVERSITI TEKNOLOGI MALAYSIA

UNIVERSITI TEKNOLOGI MALAYSIA
DECLARATION OF Choose an item.

Author's full name	**: SITI NUR ELISYA AQMAR BINTI MOHAMAD KAMAL**		
Student's Matric No.	: MCS241056	Academic Session	: 2025/2026
Date of Birth	: 23/10/2000	UTM Email	: sitinurelisyaaqmar@graduate.utm.my
Choose an item. Title	: FLIGHT DELAYS PREDICTION MODEL USING MACHINE LEARNING		

I declare that this Choose an item. is classified as:

☒ **OPEN ACCESS** — I agree that my report to be published as a hard copy or made available through online open access.

☐ **RESTRICTED** — Contains restricted information as specified by the organization/institution where research was done. *(The library will block access for up to three (3) years)*

☐ **CONFIDENTIAL** — Contains confidential information as specified in the Official Secret Act 1972)

(If none of the options are selected, the first option will be chosen by default)

I acknowledged the intellectual property in the Choose an item. belongs to Universiti Teknologi Malaysia, and I agree to allow this to be placed in the library under the following terms :

1. This is the property of Universiti Teknologi Malaysia
2. The Library of Universiti Teknologi Malaysia has the right to make copies for the purpose of research only.
3. The Library of Universiti Teknologi Malaysia is allowed to make copies of this Choose an item. for academic exchange.

Signature of Student:

Signature :

Full Name: SITI NUR ELISYA AQMAR BINTI MOHAMAD KAMAL
Date : 27 JUN 2025

Approved by Supervisor(s)

Signature of Supervisor I:	Signature of Supervisor II
Full Name of Supervisor I	Full Name of Supervisor II
Date :	Date :

NOTES : If the thesis is CONFIDENTIAL or RESTRICTED, please attach with the letter from the organization with period and reasons for confidentiality or restriction

"I hereby declare that I have read this project report and in my opinion this project report is sufficient in term of scope and quality for the award of the degree of Master in Data Science"

Signature : _____

Name of Supervisor I :

Date :

Signature : _____

Name of Supervisor II :

Date :

Signature : _____

Name of Supervisor III :

Date :

Declaration of Cooperation

This is to confirm that this research has been conducted through a collaboration Click or tap here to enter text. and Click or tap here to enter text.

Certified by:

Signature :
Name :
Position :
Official Stamp
Date

* This section is to be filled up for theses with industrial collaboration

Pengesahan Peperiksaan

Tesis ini telah diperiksa dan diakui oleh:

Nama dan Alamat Pemeriksa Luar **:**

Nama dan Alamat Pemeriksa Dalam **:**

Nama Penyelia Lain (jika ada) **:**

Disahkan oleh Timbalan Pendaftar di Fakulti:

Tandatangan :

Nama :

Tarikh :

FLIGHT DELAYS PREDICTION MODEL USING MACHINE LEARNING

SITI NUR ELISYA AQMAR BINTI MOHAMAD KAMAL

A project report submitted in fulfilment of the
requirements for the award of the degree of
Master in Data Science

Faculty of Computing
Universiti Teknologi Malaysia

JUNE 2025

DECLARATION

I declare that this project report entitled *"Flight Delays Prediction Model using Machine Learning"* is the result of my own research except as cited in the references. The project report has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature	:	*elisya*.....
Name	:	SITI NUR ELISYA AQMAR BINTI MOHAMAD KAMAL
Date	:	27 JUNE 2025

ACKNOWLEDGEMENT

Firstly, I would like to express my deepest gratitude to Allah SWT for providing me with the strength, patience, and perseverance necessary in order to complete successfully this project proposal.

Throughout the development of this proposal, I would like to extend my deepest gratitude to my lecturer, Dr Shahizan Bin Othman, for his valuable guidance, insightful suggestions, and consistent encouragement. The expertise and support that he provided during this research have played a crucial role in shaping its direction and quality.

Furthermore, I would like to extend my sincere appreciation to the Faculty of Computing, Universiti Teknologi Malaysia (UTM), for providing the necessary academic environment and the necessary resources to conduct this research.

My sincerest thanks go out to my fellow classmates and group members who have provided me with constructive feedback and motivation throughout the learning process. I am grateful to each of them for their knowledge sharing.

Lastly, I am deeply grateful for the support, prayers, and understanding that my family and friends have shown me throughout this journey. I would not have been able to complete this project proposal without the help and encouragement of all of the above.

ABSTRACT

In the aviation industry, flight delays have a significantly impact on passenger satisfaction, operational management, and financial costs. In this study, a flight delay prediction model is developed using machine learning techniques to identify flights that are likely to be delayed based on operational and weather related data. There are a number of features included in the dataset, such as departure and arrival times, airline, weather conditions, and airport traffic information, to name a few. There have been three types of machine learning models developed and compared like Random Forest, XGBoost, and a Attention-based Bidirectional Long Short-Term Memory (ATT-BI-LSTM). A comparative analysis of that ATT-BI-LSTM model the bi-LSTM model showed that the ATT-BI-LSTM model was the most accurate and able to accurately detect actual flight delays out of the bunch. As a result of this study machine learning can be used in aviation to predict flight delays earlier, help to make proactive decisions, and reduce the impact of delays on operations and customer service.

ABSTRAK

Dalam industry penerbangan, kelewatan penerbangan mempunyai kesan yang ketara terhadap kepuasan penumpang, pengurusan operasi dan kos kewangan. Dalam kajian ini, model ramalan kelewatan penerbangan dibangunkan menggunakan teknik pembelajaran mesin untuk mengenal pasti penerbangan yang mungkin tertunda berdasarkan data berkaitan operasi dan cuaca. Terdapat beberapa ciri yang disertakan dalam set data, seperti masa berlepas dan ketibaan, syarikat penerbangan, keadaan cuaca dan maklumat trafik lapangan terbang, untuk menamakan beberapa. Terdapat tiga jenis model pembelajaran mesin yang dibangunkan dan dibandingkan seperti Random Forest, XGBoost dan ATT-BI-LSTM. Analisis perbandingan model ATT-BI-LSTM dan model bi-LSTM menunjukkan bahawa model ATT-BI-LSTM adalah yang paling tepat dan dapat mengesan kelewatan penerbangan sebenar daripada kumpulan itu dengan tepat. Hasil daripada kajian ini pembelajaran mesin boleh digunakan dalam penerbangan untuk meramalkan kelewatan penerbangan lebih awal, membantu membuat keputusan proaktif, dan mengurangkan kesan kelewatan ke atas operasi dan perkhidmatan pelanggan.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Introduction

In the aviation industry, flight delays are a common challenge that affects both airlines and passengers. The delays not only impact customer satisfaction and operational costs, but also safety margins and regulatory compliance. The ability to predict flight delays has become a critical objective for enhancing airline efficiency and customer satisfaction. Since the advent of digital transformation in aviation, machine learning (ML) has become a powerful tool for delay prediction, able to analyze big datasets and identify complex patterns that conventional statistical models might miss (Kandpal et al., 2023)

The dynamic nature of air traffic systems, heavily influenced by variables such as weather, congestion, airport operations, and aircraft-specific issues, makes delay prediction difficult. Researchers have explored a wide range of machine learning algorithms over the last decade, from Random Forest and support Vector Machines to deep learning architectures like LSTM in order to build accurate and scalable predictive models (Fernandes et al., 2023)

An attention-based bidirectional LSTM model (ATT-BI-LSTM) was developed and evaluated in this study using three different machine learning methods, which are Random Forest, XGBoost, and Attention-based Bidirectional LSTM. The models were selected based on their previous performance and ability to handle structured, unstructured, and sequential data. The scalability of the flight delay prediction model is vital in order for it to be effective and efficient because it allows it to accommodate the vast amount of data generated by the aviation industry, which is continually growing. In an era of growing air traffic and data complexity, scalable models are capable of coping with larger volumes of data without compromising performance. By employing this capability, airlines can improve their decision-making and operational efficiency, allowing them to make better, more informed decisions. It

can be challenging to handle aviation industry data due to the sheer volume, diversity, and real-time nature of it. A robust data management system is required to integrate data from multiple sources, such as flight schedules, weather conditions, air traffic control, and maintenance logs. Furthermore, data quality and consistency are crucial to making reliable predictions and decisions, since inaccuracies can undermine them.

1.2 Problem Background

There are many factors responsible for the frequent delays experienced by the airline industry, such as weather conditions, technical problems, air traffic congestion, and inefficiencies in operational processes. A delay of more than an hour not only affects the operations of the airline but also creates inconvenience for passengers and results in economic losses for the airline (Hossain et al., 2020). There are billions of dollars in losses caused by delays in the airline industry every year, as a result of fuel waste, missed connections, and compensation requirements set forth by the Federal Aviation Administration (FAA). With airline operations becoming more complex every day, conventional prediction systems based on rule-based logic or simple regression are no longer sufficient when it comes to making accurate predictions.

It is well known that traditional statistical methods have been used to study delays; however, they are often unable to capture the complex relationships between variables that have been studied. With the help of machine learning models, we can forecast delays in a more dynamic and adaptable way (Choudhury et al., 2021). In a recent study, Sinha et al., (2023) found that machine learning-based models can predict nonlinear relationships between factors such as weather, route congestion, and aircraft turnaround times better than traditional models used to predict nonlinear relationships between variables. Over the past few years, a great deal of progress has been made in the field of prediction accuracy through the use of ensemble learning methods such as Random Forest, XGBoost and deep learning methods such as LSTM (Duvvuru et al., 2023)

Due to these advances, it has become very common for models to have difficulty generalizing across different datasets and operational contexts, despite the fact that these advances have been made as a result of overfitting or data imbalances

that cause problems with overfitting and generalization caused by overfitting. It is also important to note that interpretability remains one of the biggest challenges, particularly in the case of black-box models, which include many factors. Therefore, it is essential to have a comprehensive comparative assessment approach that uses a consistent methodology and evaluation metrics to evaluate the performance of different machine learning models in order to evaluate their performance.

1.3 Problem Statement

Although air transportation systems have advanced technologically, flight delays persist. As a result, current solutions are often reactive rather than predictive, which limits the ability of airlines and airport authorities to proactively manage potential delays before they happen. In order to accurately predict flight delays in advance through analysis of various influencing factors, a data-driven and intelligent solution is required that will be able to do so in real time. Using machine learning, such predictive systems can be developed with greater accuracy and reliability as well as higher levels of precision.

1.4 Research Goal

Using historical flight and weather data, this study aims to build a machine learning model that can predict whether a flight will be delayed or on time using historical data and different machine learning algorithms in order to develop an accurate prediction system. It should enable operational decision-making and improve schedule reliability by providing accurate.

1.4.1 Research Objectives

The objectives of the research are :

a) To collect and process flight and weather data for delay prediction

b) To have a machine learning model for classifying delayed flights, such as Random Forest, XGBoost, and ATT-BI-LSTM, developed and compared.
c) To identify the best model based on ROC-AUC, F1-score, and accuracy-precision metric.

1.5 Scope of Research

There was a limitation in this study that limited it to only testing structured historical flight data coupled with weather information. This study focuses on classifying flight delays based on departure time, distance, weather conditions, and other operational variables. As far as machine learning techniques are concerned, only supervised techniques are considered. Through the use of Python-based tools, the models will be trained and evaluated using a cross-validation analysis. Specifically, the scope of this study does not include unscheduled flights, external geopolitical factors, or real-time streaming data.

1.6 Report Content Layout

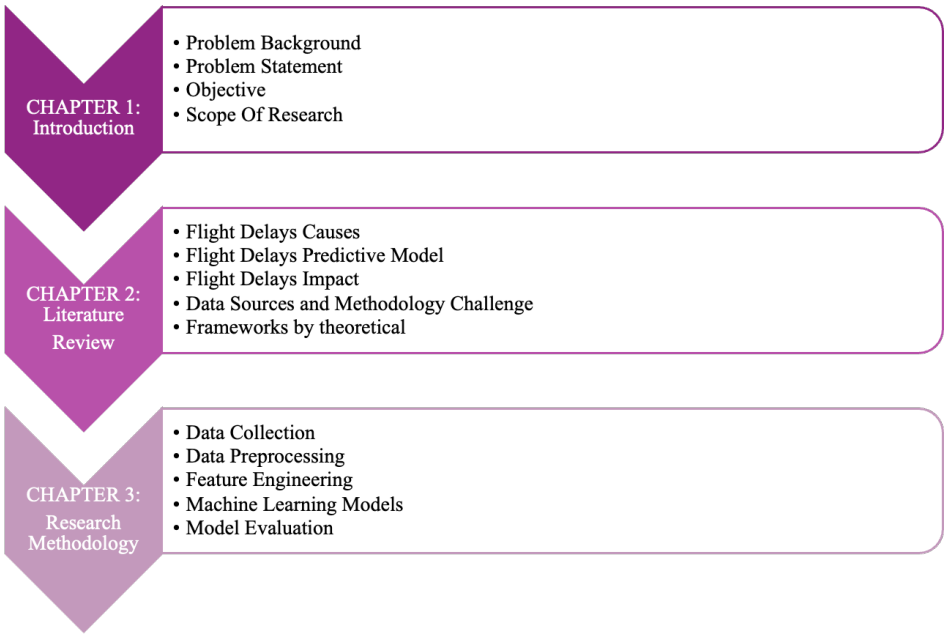

Figure 1.1: Report Content Layout

1.7 Summary

The prediction of flight delays is a complex but vital task that can be enhanced considerably through the application of machine learning techniques that can be applied to this task. This chapter outlined the motivations, objectives, and scope of the research in general. Using structured data combined with modern machine learning algorithms, this study aims to improve the prediction of delays and contribute to better operational planning in the aviation sector by integrating structured data with modern machine learning algorithms.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In order to attract new customers, one of the most important factors is to satisfy your current customers. Therefore, flight delays are one of the most important measures of a carrier's and an airport's quality of service. It is very important to study delays in Malaysia due to a variety of factors, such as the tropical climate and the fact that most of the travel is driven by the holiday season. The purpose of this section is to provide an overview of the study's research scope and its general understanding.

2.2 Flight Delays Causes

According to three studies, flight delays are the result of a combination of operational, environmental, and technical factors. It has been demonstrated that weather conditions such as wind speed, atmospheric pressure, and precipitation have significantly disrupted flight schedules, accounting for 47.46% of the delays in one study (Hatıpoglu, I., & Tosun, Ö., 2024). Another 29.09% of delays are caused by air traffic control bottlenecks, such as route congestion and hourly flight caps (Hatıpoglu, I., & Tosun, Ö., 2024). Airport-specific factors, such as runway availability, increase disruptions (Ng et al, 2020). There are a number of causes of scheduling conflicts, such as late departure times and short turnaround intervals (Hatıpoglu, I., & Tosun, Ö., 2024), as well as technical variables such as the type of aircraft and the number of passengers.

2.2.1 Weather-Related Delays

Various factors, including weather conditions, significantly contribute to flight delays through a variety of mechanisms, as highlighted in the study. The long-term impact of weather-related delays is disproportionately severe, despite the fact that weather delays account for a relatively small proportion of total delays, 3.86% at JFK, with delays lasting an average of 69.81 minutes, far longer than those caused by other delays. Key meteorological factors such as temperature, dew point, humidity, wind speed, wind gusts, atmospheric pressure, and precipitation directly influence flight operations. For instance, extreme temperatures can affect aircraft performance, while high wind speeds and gusts disrupt take-off and landing safety, necessitating delays or diversions. Heavy rainfall, storms, or typhoons reduce visibility and cloud ceiling heights, forcing airports to operate at reduced capacity or halt operations entirely. For example, typhoons usually happen in South Korea from July until September, causing most of the flights to be delayed (Kim & Park, 2024). Additionally, the integration of long-term weather data from 2010 to 2021 revealed that abnormal weather patterns, exacerbated by climate change, are increasing flight disruptions globally.

2.2.2 External Factors

Due to external factors such as political and health disruptions, including pandemic-related travel bans and airspace closures, Kuala Lumpur International Airport (KLIA) faces operational challenges. In 2020, Malaysia's Movement Control Order (MCO) severely disrupted flight schedules and revenue streams due to the COVID-19 pandemic (MAVCOM, 2021). As a result of geopolitical tensions, such as airspace closures in the South China Sea during 2023, flights were rerouted and congestion increased (ICAO, 2022). In particular, Chinese New Year and Hari Raya Aidilfitri create periodic demand spikes for passengers. During the 2023 Hari Raya Aidilfitri season, MAHB reported a 40% increase in KLIA passenger traffic, straining check-in systems and causing average delays of 25-35 minutes. Based on World Bank findings (2022), which stressed the need for a dynamic resource allocation to manage peak-period bottlenecks in Asian aviation hubs, these seasonal fluctuations are in line with those on the World Bank.

2.3 Flight Delays Predictive Model

A number of machine learning models were used to predict delays, including XGBoost (Hatıpoglu, I., & Tosun, Ö., 2024), Random Forest (G.Guan et al, 2020), and neural networks (PDF 3), with ensemble methods outperforming other approaches. Despite overfitting and limited generalizability of single-airport data (G.Guan et al, 2020) and class imbalance in datasets (Hatıpoglu, I., & Tosun, Ö., 2024), SMOTE (Hatıpoglu, I., & Tosun, Ö., 2024) and under-sampling (G.Guan et al, 2020) were able to address class imbalance in datasets.

2.3.1 Statistical and Machine Learning Approaches

An analysis of the Turkish airport data in the period from 2016-2018 is presented in the case study. Using the Synthetic Minority Oversampling Technique (SMOTE), Hatapoglu and Tosun (2024) have incorporated meteorological variables such as temperature, wind speed, atmospheric pressure into the dataset by using the Synthetic Minority Oversampling Technique (SMOTE). As part of the research, seven different machine learning models were evaluated-Logistic Regression, Naive Bayes, Artificial Neural Networks (ANN), Random Forest, XGBoost, CatBoost, and LightGBM-while applying Bayesian optimization for hyperparameter tuning. Based on SHAP (SHapley Additive Explanations) values, scheduled departure time, passenger count, and atmospheric pressure were the best predictors, while wind speed and humidity did not have a significant effect. The gradient-boosting method XGBoost, which exhibits moderate overfitting during training, achieves 80% accuracy with a 0.41 error rate, outperforming other methods.

Gui G et all 2020 combines ADS-B flight data, weather, flight time, and airport data from December 2018 to May 2019, resulting in 5761 instances of flights with a class imbalance of 3368 non-delayed flights and 2393 delayed flights between December 2019 and May 2019. As an example of the detail of the data set, flight altitude, wind speed, airport traffic volume, and weather are just some of the components of the dataset. Quantification and normalization were applied to the data

to alleviate imbalance by representing categorical variables such as airports and weather as numbers. In this study, LSTM networks were used to classify sequential data, and Random Forests were used for classification, with hyperparameter optimization of memory depth and tree depth and grid searches. Even though LSTM is capable of classification with 81.4% accuracy for three classes, 70% accuracy for four classes. As a result of inadequate data, LSTMs overfitted. Even though meteorological variables, including wind speed and airport traffic flow, played a more significant role in predicting airport performance than operational flow.

Based on Hong Kong International Airport's 2018 data, this study predicts delays using decision trees, SVM, Random Forest, and neural networks. There was also weather information and airline code data included in the dataset. It was neural networks that yielded the highest accuracy (albeit with a heavy computational demand), while it was decision trees that priotized speed. Researchers proposed data-sharing workflows between airlines, airports, and insurers to improve delay management. There was a clear dominance of operational factors over weather in the prediction of delay. It is recommended that regression models be used to predict delay duration and hybrid architectures be developed to balance efficiency and accuracy.

Using data collected over five years from U.S. flights, the researchers categorized 18 million flights into two classes: 15 million no delayed flights and 3 million delayed flights. Among the features of the dataset are flight schedules, weather variables such as wind speed and temperature, as well as airport traffic complexity. Undersampling was applied to address class imbalance, reducing Class 0 to match Class 1, but this approach led to a loss of information. As part of the data preprocessing, noise was reduced and features were selected, focusing on departure or arrival times, route congestion, and meteorological factors. In the research, the Levenberg-Marquardt (LM) algorithm is optimized with a stacked denoising autoencoder (SDA). It is a supervised fine-tuning model with supervised unsupervised pre-training for learning robust features from noisy data and a supervised LM for faster convergence. The following steps are critical to the data preprocessing process: Undersampling to balance classes and mitigate majority bias. Using a three-phase model architecture (1) data denoising with autoencoders, (2) supervised fine-tuning with LMs, and (3) evaluating accuracy, precision, recall, and F1-scores. In this

comparison, the SDA-LM model is compared to the Stacked Autoencoder-Levenberg-Marquardt model (SAE-LM) and to the standalone SDA model. An analysis of the impact of data noise on model performance was conducted to combat overfitting. On the balanced dataset, SDA-LM performed better than SAE-LM (82% accuracy) and SDA (79% accuracy), achieving 89% accuracy with improved precision (87%) and recall (85%). SDA-LM maintained robustness despite class bias on the imbalanced dataset. A LM algorithm reduces training time and improves convergence compared to a traditional backpropagation algorithm.

The next study uses data from a 2020 flight dataset with 28821 samples and 23 attributes, including flight schedules, weather variables, and airport-related attributes such as departure or arrival times, wind speed, temperature, and airport codes. Exploratory data analysis (EDA) was performed in Jupyter Notebook to address missing values and normalize features in the dataset. Using Naive Bayes, the data were split into training 70% of the data and testing 30% sets using 10-fold cross-validation. Despite its simplicity, the Naive Bayes model was able to achieve 80.6% accuracy on the test dataset, which is comparable to other methods, such as Logistic Regression. Its performance, however, was marginally lower than ensemble techniques such as Random Forest (90% in another study) or XGBoosts (80% in a Turkish airport study), illustrating its limitations when it comes to capturing complex interactions between variables. To improve accuracy further, hybrid approaches or advanced feature engineering are required in addition to Naive Bayes' potential as a lightweight, interpretable baseline model.

2.4 Flight Delays Impact

2.4.1 Economic Costs and Environmental

Yazdi et al. (2020) state that prolonged idling and fuel-intensive holding patterns result in an increase in CO2 emissions during flight delays. Additional waste from meals and services during delays, as well as noise pollution from a congested airport, further strain sustainability efforts (Hatipoglu & Tosun, 2024). Delays cost the aviation industry billions of dollars each year, including direct operational expenses

and indirect effects like reduced consumer welfare and inflated airfares (Brueckner et al., 2021). Cascading delays like these highlight the importance of predictive models and optimal scheduling to minimize delays and their long-term consequences (Ng et al., 2020).

2.4.2 Passenger and Airlines

In addition to decreasing passenger satisfaction, loyalty, and future bookings with the same airline, flight delays also significantly affect customer satisfaction. There are several reasons why delays disrupt travel plans, including missed connections, wasted time, and frustration, all of which erode the trust of travellers in airlines (Ng et al., 2020). Airline delays increase operational costs due to higher fuel consumption, extended block times, and penalties for late arrivals (Yazdi et al., 2020). As a result of frequent disruptions, the reputation of a brand is compromised, as well as customer retention and revenue losses (Brueckner et al., 2021). As a result of operational inefficiencies, such as cascading delays across interconnected routes, stock values may suffer (Zhang et al., 2021).

2.5 Data Sources and Methodological Challenges

The quality, completeness, and representativeness of the dataset heavily influence the effectiveness of any predictive model. This study uses a combination of flight schedule information and weather related creatures as its primary dataset. It reflects real-world operational conditions affecting flight punctuality. The model can be used to predict delays using machine learning.

2.5.1 Data Sources Characteristics

In the dataset, flight operation data and weather attributes are combined to provide:

- Flight Operation Variables: MONTH, DAY_OF_WEEK, OP_UNIQUE_CARRIER, DEST, DEP_DELAY, CRS_DEP_M, DEP_TIME_M, CRS_ARR_M, DISTANCE, TAXI_OUT
- Weather Data: Temperature, Dew Point, Humidity, Wind, Wind Speed, Wind Gust, Pressure, and Condition

Despite not mentioning the exact source, this data seems to be derived from both internal airline records and external weather stations. Even so, incorporating these fields is in line with best practices in previous research that have been found to be effective. According to rebollo: Balakrishnan (2014), as well as Choi et al. (2022), these studies highlighted that the weather features significantly influence the accuracy of flight delay prediction models.

This dataset also includes time-based features which are crucial for capturing the influence of departure timing on delays, as demonstrated by Gopalakrishnan & Balakrishnan (2017).

2.5.2 Methodological Challenges

While working with this dataset, several methodological challenges emerged:

1. **Incomplete or missing data**
   There were missing values in some columns, particularly continuous weather-related columns like Wind. To resolve this issue, mean imputation was applied as this method used in many machine learning pipelines. This approach, however, may reduce variability in the data and may not capture its true distribution, as Yu et al. (2022) cautioned.

2. **Irrelevant and high-cardinality features**
   There were no columns such as TAIL_NUM (aircraft tail number) or scheduled timestamps (sch_dep, sch_arr) added due to their high cardinality or non-relevance to the prediction task. In three-based and deep learning models, high cardinality categorical features often increase computational complexity and overfitting.

3. **Data Encoding for Categorical Variables**
   The categorical features OP_UNIQUE_CARRIER, DEST, Condition, and Wind needed to be encoded before being entered into models. In line with Chakrabarty et al. (2019) methodological guidelines, label encoding was used

when tree-based models were analyzed, and one-hot encoding was used when XGBoost was analyzed.

4. The complexity of sequential and temporal modeling

Even though the dataset includes time-based features, it is organized in a flat, tabular structure rather than a sequential one. For models such as Attention-based BI-LSTM, data had to be reshaped into sequences of one time step, which may not capture long-term dependencies. Nguyen et al. (2018) also highlight this limitation of the transformation, while valid for experimentation.

5. Imbalance in class

Most flight delay datasets reveal that the number of non-delayed flights outnumbers delayed flights by a significant margin. As a result of this imbalance, model predictions may tend to be biased towards the majority calss. A variety of mitigation techniques, including F1-score, ROC_AUC, and class weighting, were employed to ensure fair evaluation. These techniques are consistent with Bertsimas & Kallus (2014).

2.6 Frameworks by theoretical

Rather than seeing delays as isolated events, the Air Transport System Model emphasizes the interdependence between airports, airlines, and passengers. As an example, Hatipoglu and Tosun (2024) emphasize how disruptions at airports affect airline networks, affecting passenger connections, flights, and operating costs. It is aligned with studies analyzing delay propagation in multi-airport systems, where cascading effects are modeled using network-based approaches (Zhang et al., 2021). Furthermore, the Theory of Constraints (TOC) focuses on identifying operational bottlenecks that exacerbate delays, such as runway congestion. By optimizing buffer times and resource allocation at critical nodes, TOC principles can mitigate delays (Brueckner et al., 2021)

2.7 Conclusion

In conclusion, machine learning models, such as XGBoost, Random Forest, and LSTM networks, are increasingly being used to forecast disruptions and mitigate economic and environmental impacts caused by flight delays. Additionally, during festivals like Hari Raya, tropical weather patterns and cultural travel surges amplify delays, yet localized studies are scarce. In Chapter 3, we will investigate and explore the approach for flight delay prediction using machine learning on Malaysia Airlines.

CHAPTER 3

RESEARCH METHODOLOGYz

3.1 Introduction

This study aims to develop a machine learning-based predictive model that will allow airlines to forecast flight delays in the future. There are two approaches involved in this methodology, which are data-driven techniques involving deep learning models as well as classical machine learning techniques. In this paper, we will look into historical flight data in order to identify patterns and correlations that can be used to predict potential delays more accurately. During the process, data are collected, pre-processed, models are developed, models are trained, evaluated, and a comparative analysis is performed on three different models, which are Random Forest, XGBoost, and an Attention-Based Bidirectional Long Short-Term Memory (ATT-BI-LSTM) network.

3.2 Data Collection

In this study, the dataset was sourced from Kaggle (Flight Take Off Prediction). There are more than 20,000 records in the data between November 2019 and December 2020. There are a number of key variables that should be considered, including flight number, airline, scheduled and actual departure/arrival times, flight distance, weather conditions, and day of week. To enhance the performance of the model, additional data such as weather conditions corresponding to JFK Airport during the departure period were also incorporated in order to enhance the accuracy of the model.

3.3 Data Preprocessing

In order to ensure the reliability and accuracy of a model, effective preprocessing is essential. To carry out the following steps, we took the following steps:

3.3.1 Convert Numeric Columns and Handling Missing Data

This dataset includes several key numeric columns, which are relevant to weather conditions and flight operations, such as, for instance, ‘Temperature’, ‘Dew Point’, ‘Humidity’, ‘Wind Speed’, ‘Wind Gust’, ‘Pressure’, and ‘TAXI_OUT’. To ensure consistent data formatting, each of these columns is explicitly converted to a numeric type, especially if any values were initially stored as strings, so that the data can be formatted consistently. For example, when a value is missing or invalid in a column, then the column's mean is used to fill that value so that data does not get lost, but statistical integrity is maintained. Missing values are filled with the string ‘Unknown’ in the categorical column ‘Condition’. This likely represents the weather conditions at a given time. By using this method, the information in the dataset will still be retained, while a clear indication will be provided of records that originally had missing data in them, rather than removing the records and losing the data as a result.

3.3.2 Remove Irrelevant Column.

In addition to this, the dataset is emptied of columns such as ‘TAIL_NUM’, ‘sch_dep’, and ‘sch_arr’. Depending on their relevance to the prediction task, these features either do not contribute to the prediction task or have a very high number of unique values, which could introduce noise in the prediction task or lead to overfitting in machine learning models. By dropping them, the dataset is streamlined for analysis, making it easier for the analyst to analyze.

3.3.3 Creating Categorical Variables

In order to encode categorical variables such as ‘OP_UNIQUE_CARRIER’, ‘DEST’, and ‘Condition’, label encoding is used. Using this method, each unique text category can be converted into an integer, which makes it numerically suitable for input into machine learning algorithms that require numeric input features to work properly.

3.3.4 Create a Binary Delay Label

The ‘DEP_DELAY’ column is used to create a new binary column called ‘is_delayed’ that contains the values for both delays. If the departure time of a flight has been delayed by more than 15 minutes, then that flight will be labeled with 1 (indicating a delay), whereas all other flights will be labeled with 0. As a result of this step, continuous delay data are transformed into binary classification labels, which enable the development of classification models to be developed.

3.3.5 Remove The Remaining Missing Values From The Table

The last step in the preprocessing step is to remove from the dataset any rows that still contain missing values after all other preprocessing steps have been completed, so if there are any remaining rows in the dataset with missing values, those rows are deleted. In this way, you can make sure the final dataset is clean and does not contain any null values that may interfere with the training or evaluation of the model.

3.4 Feature Engineering

In order to improve the model’s ability to detect complex relationships within the data, a series of engineered features was added to the dataset to enhance its detection capabilities. As a result of these transformations, we have been able to elevate the representation of temporal, operational, and weather-related factors that have been shown to influence delays to a more accurate form.

3.4.1 Time Patterns

A number of temporal features have been designed to capture typical traffic patterns and weekly behavior patterns. There was introduced a binary variable was introduced for the purpose of indicating peak operational hours, specifically the period between 6:00 a.m. to 10:00 a.m. and 4:00 p.m. to 7:00 p.m., when airports often experience higher congestion and a higher risk of delays. Furthermore, a weekend indicator was created in an effort to flag flights that are scheduled to take place on Saturdays and Sundays, due to the fact that these days tend to have different traffic patterns and staffing patterns compared to weekdays.

3.4.2 Operational Complexity

Taking into account the airport's operational load and complexity, a flight volume per hour feature was added by counting departures per hour. By using this metric, we can be able to determine how congested an airport is. In addition, flight distances were categorized into categories like short-haul, medium-haul, and long-haul. As a result of categorization, it is possible to identify patterns in delays that may be related to the length or type of the route that has cause delays.

3.4.3 Climate-Based Features

The raw weather descriptions were used as the basis for many of the features in order to better reflect adverse conditions in the weather. It is possible to create binary flags for fog, rain, and snow by parsing keywords from textual weather data in order to allow the model to be able to immediately recognize the presence of disruptive weather events. In addition, a temperature range variable was introduced, based on the difference between maximum and minimum temperatures. As an indicator of weather instability, this feature can significantly impact flight schedules.

3.5 Machine Learning Models

In this study, three predictive models were developed and evaluated in order to identify flights that are likely to encounter delays in takeoff, and to identify them before they

begin flying. The models include both traditional ensemble methods and advanced deep learning architectures integrating tubular and sequential features. The purpose of this study was to assess the predictive accuracy and robustness of different modelling approaches.

3.5.1 Random Forest Classifier

Random Forest classifiers were used to benchmark the performance of deep learning models against a classical machine learning baseline. This method constructs multiple decision trees by bootstrapping subsets of data and randomly selecting features at each split. Using this technique, it is much easier to enhance robustness as well as mitigate overfitting, especially when the dataset contains noise or missing values that may adversely affect the model.

In addition to being effective for tabular data, Random Forest is also highly interpretable. An advantage of this platform is its ability to calculate feature importance scores, which can be used to determine which factors contribute to flight delays. With its robustness, the Random Forest model often serves as an appropriate baseline, especially when dealing with high-dimensional datasets.

With the help of the scikit-learn Python library, the Random Forest classifier was implemented. The performance of the model was further improved by tuning hyperparameters such as the number of estimators and the maximum depth of trees using grid search and cross-validation. As a result of this systematic tuning, the model's generalization ability was optimized while computational efficiency was maintained.

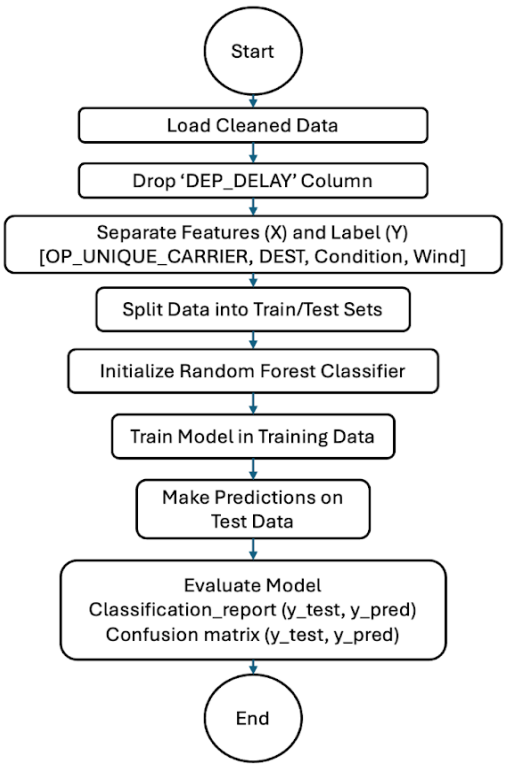

Figure 3.1: Flowchart Random Forest Classifier

3.5.2 XGBoost

It was discovered that the XGBoost algorithm performs well in structured data tasks due to its ability to boost gradients and its high accuracy. This approach was implemented by using the official XGBoost library, and parameters such as learning rate, maximum depth, and subsampling ratio were tuned based on the library and then evaluated using cross-validation to ensure proper performance. Based on the results, it was found that XGBoost was more accurate and faster than other algorithms, making it an excellent choice for structured data tasks. The library's flexibility improved model performance significantly through hyperparameter tuning. Furthermore, XGBoost has the ability to handle large datasets due to its scalability and robustness. The algorithm was also able to perform better on structured data tasks since it was able to handle missing data and reduce overfitting.

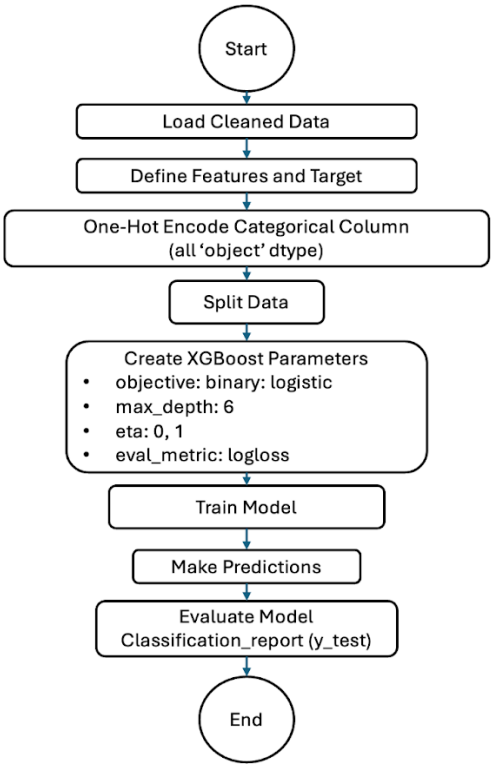

Figure 3.2: Flowchart XGBoost Classifier

3.5.3 ATT-BI-LSTM

In this study, a third model was developed called the Attention-Based Bidirectional Long Short-Term Memory (ATT-BI-LSTM). The deep learning model architecture comprises of embedding layer, a bidirectional LSTM layer, and a layer of attention that can be trained on a given dataset. Due to the structure of the model, it was able to capture both forward and backward time-dependent events, in addition to using the attention mechanism to weigh the importance of important events. Using TensorFlow/Keras, the model was implemented to prevent overfitting by using early stopping and dropout techniques. The model achieved promising results by effectively learning temporal dependencies and focusing on critical events. Aside from improving prediction accuracy, this method also demonstrated its ability to handle complex sequential data in a wide range of applications.

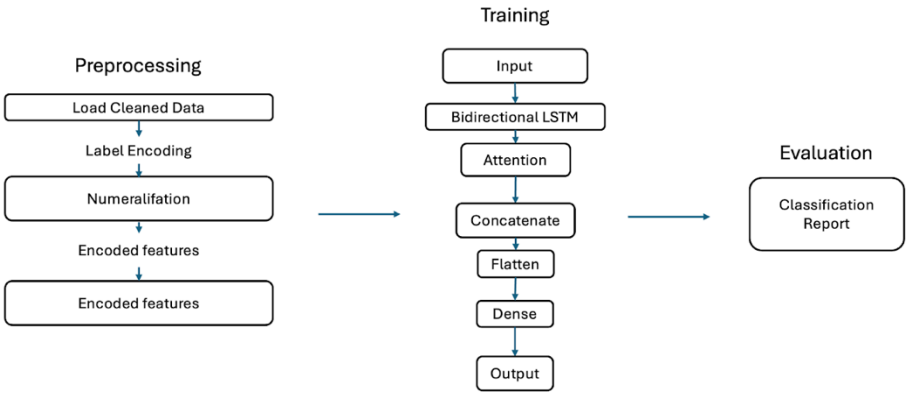

Figure 3.3: Flowchart ATT-BI-LSTM

3.6 Model Evaluation

The predictive models developed for flight take-off delay classification were evaluated comprehensively to ensure their reliability and effectiveness. In support of training, testing, and validation, multiple evaluation metrics were used as well as comparisons between models and robust tools for training, testing and validation. The evaluation process was created to understand both the practical implications of false positives and false negatives in operational contexts with significant consequences for both False Positives and False Negatives.

3.6.1 Evaluation Metrics

Model performance was evaluated using several metrics. A measure of accuracy can be misleading when data are imbalanced, since it reflects the overall number of correct predictions. Precision indicates the percentage of correctly predicted delays among all predicted delays, minimizing false alarms. Model recall measures the ability of the model to detect actual delays, reducing missed delays. F1 scores, which are the harmonic means of precision and recall, balance imbalanced datasets. By using ROC-AUC, the model was assessed for its ability to differentiate between delayed and non-delayed flights. To visualize classification errors, confusion matrices were used. Several cross-validation tests were performed to ensure robustness and generalization.

```
# Predict and evaluate
y_pred_probs = bst.predict(dtest)
y_pred = (y_pred_probs > 0.5).astype(int)

print("XGBoost Classification Report:")
print(classification_report(y_test, y_pred))
```

```
y_pred_prob = model.predict(X_test_lstm)
y_pred = (y_pred_prob > 0.5).astype(int)

print("Classification Report:")
print(classification_report(y_test_lstm, y_pred))
```

```
# Predict and evaluate
y_pred = rf.predict(X_test)

print("✅ Random Forest Classification Report:\n")
print(classification_report(y_test, y_pred))

print("🧾 Confusion Matrix:")
print(confusion_matrix(y_test, y_pred))
```

Figure 3.4: Model Prediction and Evaluation using Machine Learning

3.7 Summary

The methodology of the study is explained in detail in this chapter, from data collection to model evaluation. Using this process, a flight delay predictive model using machine learning can be measured systematically.

CHAPTER 4

INITIAL RESULTS

4.1 Introduction

The purpose of this chapter is to present the main findings from the analysis of flight data that was used to predict airline delays. This section begins with an overview of the dataset structure and the distribution of flights that are delayed versus those that are on time. Identifying trends and patterns relating to delays is accomplished through exploratory analysis. After that, the data preprocessing steps will be summarized, including the handling of missing values, encoding, and feature scaling. By using standard evaluation metrics, three models are compared. The chapter concludes with hey insights that inform the effectiveness of each model and guide further improvements.

4.2 Exploratory Data Analysis (EDA)

To gain a better understanding of the structure, distribution, and patterns of the dataset related to flight delays, exploratory data analysis was conducted. Our first step was to ensure the dataset, which includes information such as departure carrier, weather conditions, and delay duration time, was complete.

4.2.1 Data Collection

Column Name	Description	Relevance to Delay Prediction
MONTH	Month of the flight (Jan - Dec)	Weather and demand can influence delay in different seasons

DAY_OF_MONTH	Day of Month (1 -31)	In some cases such as holiday, delays may be more noticeable
DAY_OF_WEEK	Day of the week (1-Monday,...,7-Sunday)	The weekend travel pattern may affect congestion and delays
OP_UNIQUE_CARRIER	Airline code (AA, UA, DL)	Depending on the airline, on-time performance can vary
TAIL_NUM	Aircraft tail number (unique identifier)	Indirectly relevant. Often omitted to reduce noise
DEST	Destination airport code	Congestion causes delays at some destinations
DEP_DELAY	Departure delay in minutes	Label created with target variable (is_delayed)
CRS_ELAPSED_TIME	Scheduled duration of the flight	The timing might be stricter on longer flights due to more delay buffer
DISTANCE	Distance between destination and origin	Schedules and delays may be related
CRS_DEP_M	Scheduled departure time in minutes from midnight	Suitable for extracting trends by time of day
DEP_TIME_M	Actual departure time in minutes from midnight	Use to calculate duration delay
CRS_ARR_M	Scheduled arrival time in minutes from midnight	Planned flight duration can be calculated with CRS_DEP_M
Temperature	Temperature at departure time	Temperatures can affect aircraft performance and cause delays

Dew Point	Measure of moisture (°F)	Visibility and fog conditions may be affected
Humidity	Relative Humidity (%)	Delay associated with weather may be caused by high humidity
Wind	Wind direction	The use of runways may be affected; only standardized data is useful
Wind Speed	Speed of wind at the departure location	Take-off and landing can be delayed due to high wind
Wind Gust	Maximum recorded wind gust	Flights may be temporarily grounded by sudden gusts of wind
Pressure	Atmospheric pressure	Storms or bad weather may be indicated by low pressure
Condition	Weather condition	Impact delay risk directly
sch_dep	Scheduled departure timestamp	Time-based features are often dropped after extraction
sch_arr	Scheduled arrival timestamp	Time-based features are often dropped after extraction
TAXI_OUT	Time taken to taxi from gate to runway (in minutes)	Congestion and delays often correlate with longer taxi wait times

Table 4.1: Data Description

4.2.2 Basic Inspection

A basic inspection of the dataset was conducted to understand its structure, data types, and completeness. Summary information, such as the number of rows and columns, the type of column data, and any missing values was analyzed to identify any potential data quality issues. Using the DEP_DELAY column, the binary classification target is_delayed was created, where 1 indicates a flight delayed by more than 15 minutes

and 0 indicates an on-time flight. As a result of this transformation, the analysis focused on a clear classification task. Figure 4.1 shows Delay Class Distribution based on initial dataset.

```
 Delay Class Distribution:
is_delayed
0    86.557946
1    13.442054
```

Figure 4.1: Initial Delay Class Distributions

4.2.3 Class Distribution Analysis

Class distribution analysis was used to visualize the proportion of delayed flights versus those that arrived on time, based on Figure 4.2.

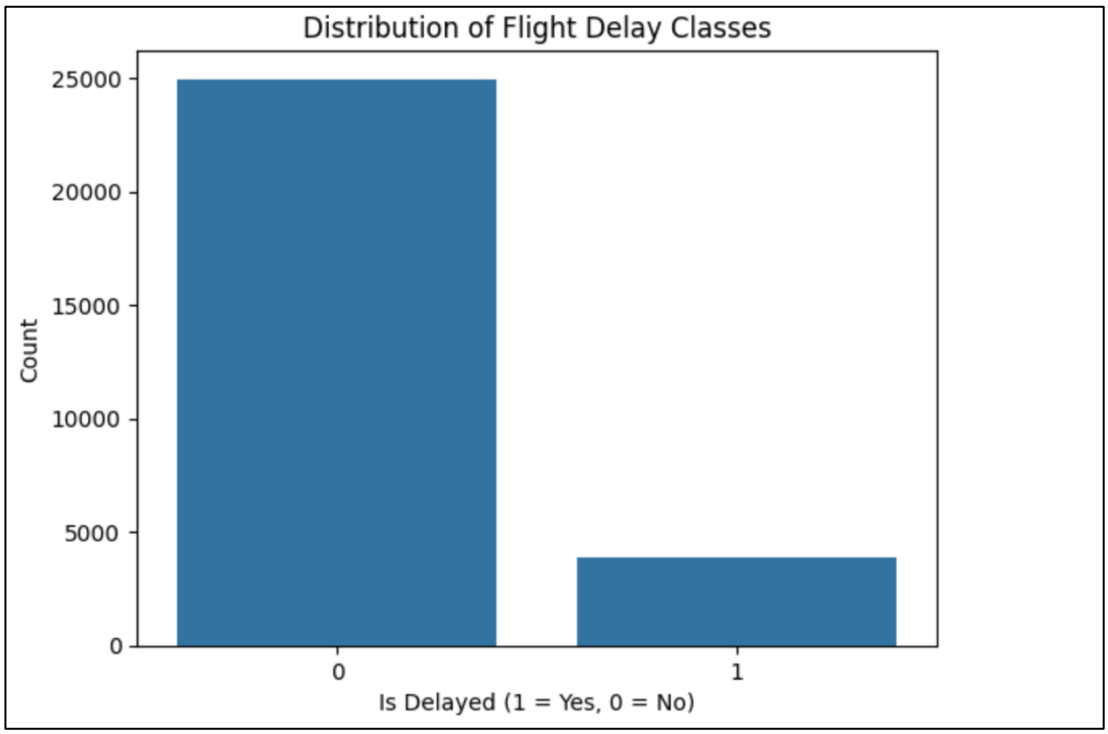

Figure 4.2: Bar Chart Distribution of Flight Delay Classes

There was a noticeable imbalance in the results, with more on-time flights, suggesting further evaluation should consider F1 score and recall metrics. Data from the DEP_TIME column was used to plot the frequency of delays throughout the day to

examine temporal patterns. Traffic congestion or weather conditions may have contributed to delays earlier in the morning and later in the evening.

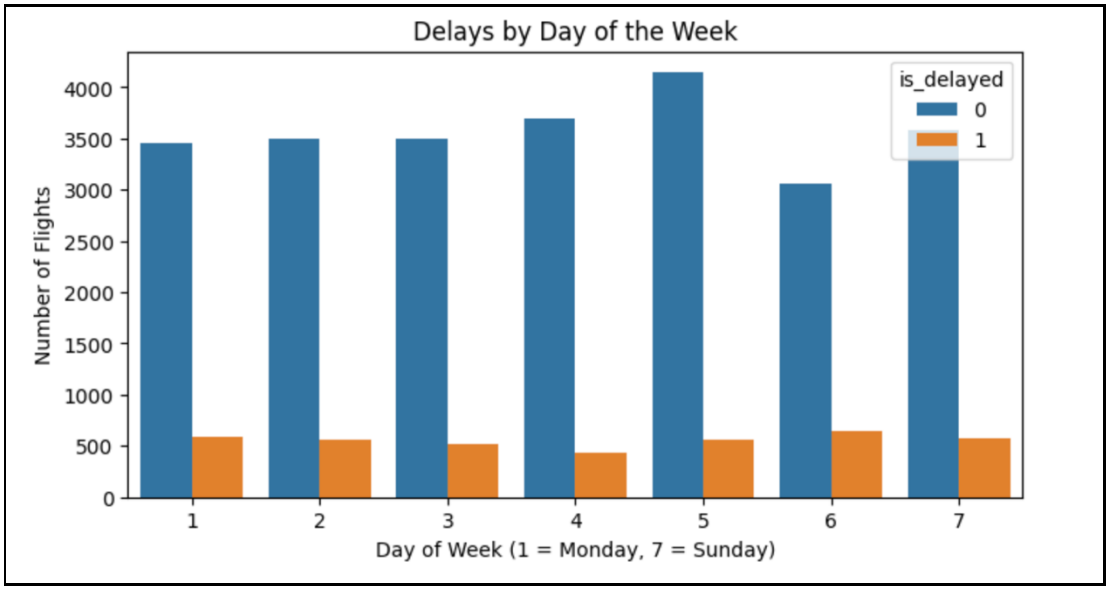

Figure 4.3: Clustered Bar Chart Delays by Day of the Week

Based on Figure 4.3, it was also observed that delays were more frequent on Saturdays and Sundays, which was possibly due to heavier traffic on weekends.

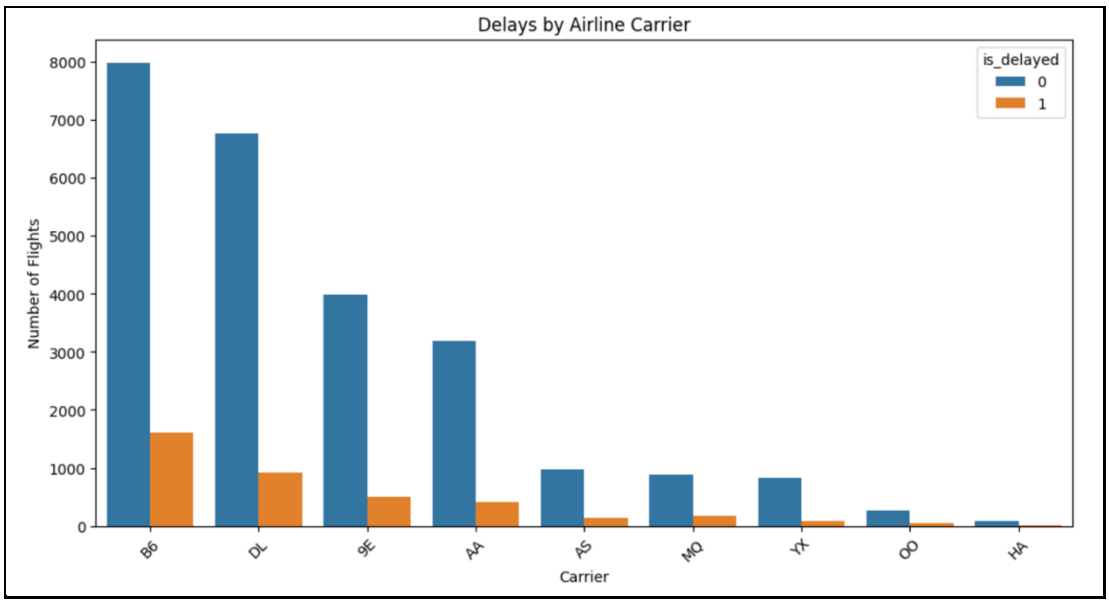

Figure 4.4: Clustered Bar Chart Delays by Airline Carrier

Moreover, categorical trends were analysed, such as delays by airline carrier. There were significantly higher delays with some carriers compared to others in the plot based on Figure 4.4, which suggests that carrier performance and scheduling efficiency may contribute to this.

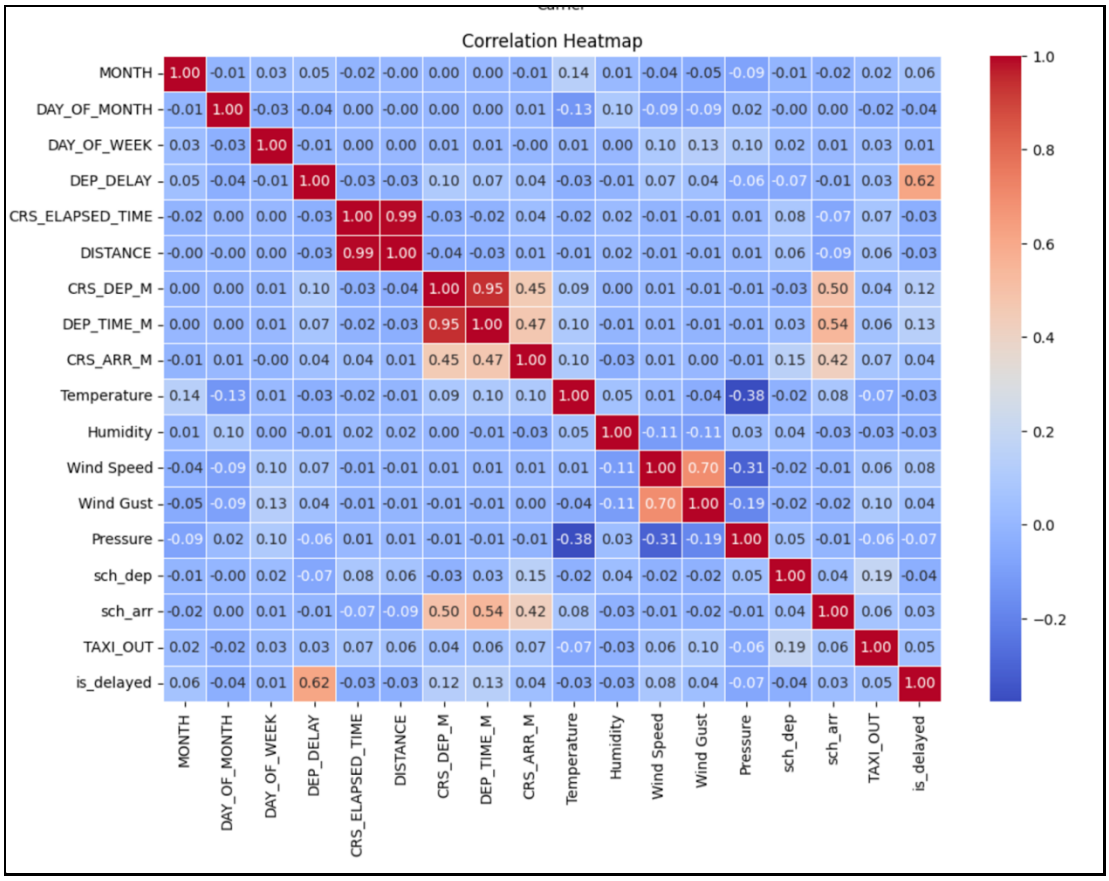

Figure 4.5: Correlation Heatmap

Figure 4.5 shows correlation heatmaps were generated to analyse relationships between numerical features. Some features, such as taxi-out time, humidity, and wind speed, showed moderate correlations with delays, suggesting their potential to be used in predictive modelling.

4.2.4 Boxplot

Boxplots were used to visualize the relationship between weather variables and flight delays.

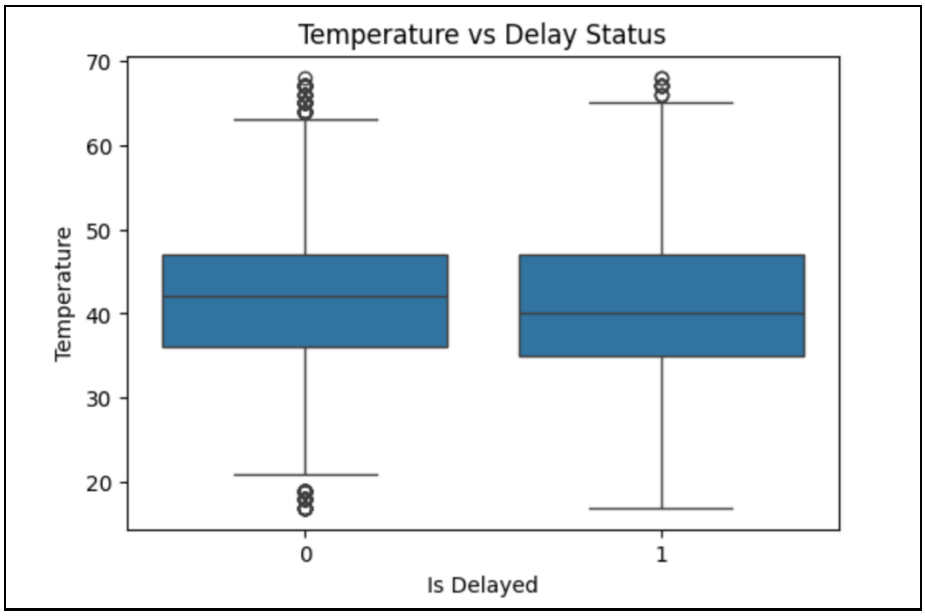

Figure 4.6: Boxplot Temperature vs Delay Status

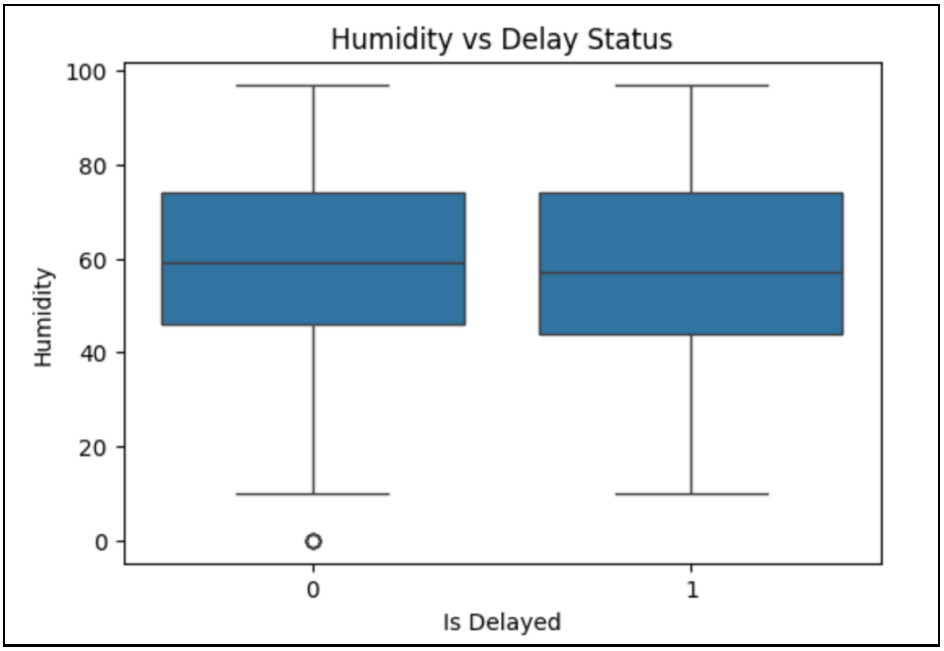

Figure 4.7: Boxplot Humidity vs Delay Status

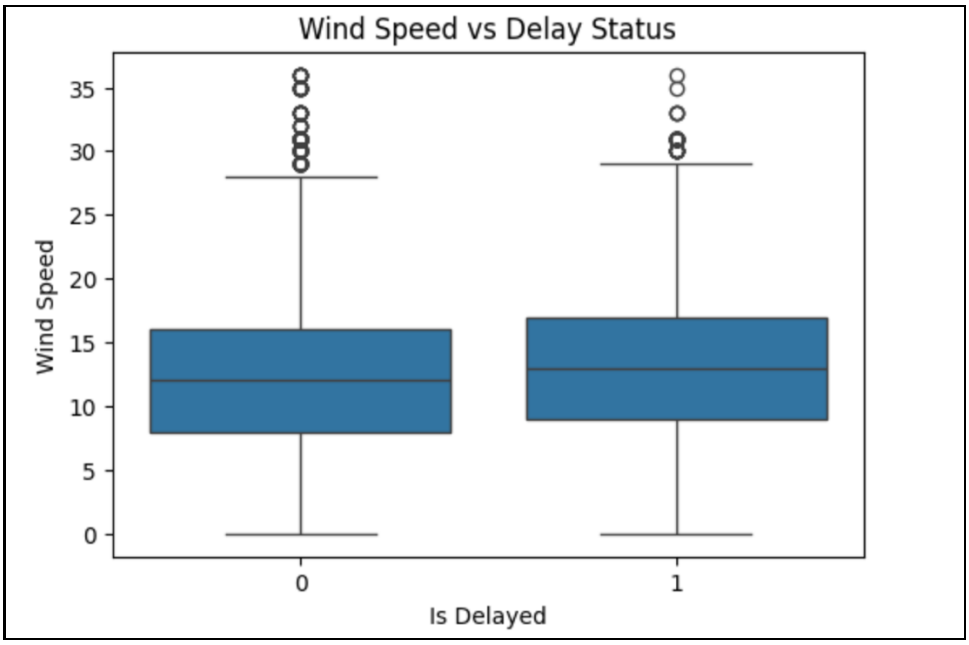

Figure 4.8: Boxplot Wind Speed vs Delay Status

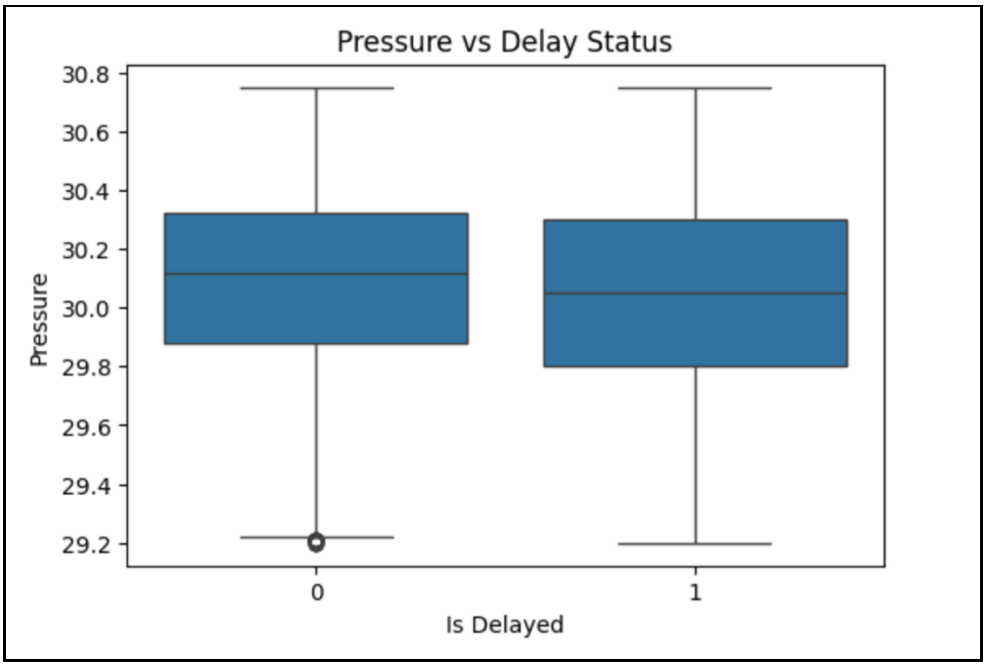

Figure 4.9: Boxplot Pressure vs Delay Status

According to comparisons of temperature, humidity, wind speed, and pressure between delayed flights and non-delayed flights, delays are more likely to occur in cool, humid, and windy environments. However, these differences were not pronounced, suggesting that while weather plays a role in flight delays, it is likely only one of several contributors. A key outcome of the EDA was the identification of

relevant features and patterns that could be employed to enhance the accuracy of prediction models.

4.3 Data Preparation and Cleaning Data

Initially, we examined the raw dataset's structure and dimensions to confirm its structure and dimensions. Next, both numerical and categorical features were cleaned and prepared for model development. We converted key numerical columns, such as temperature, dew point, humidity, wind speed, wind gust, pressure, and taxi-out time, to a numeric format to handle any nonstandard values. In these columns, missing values were filled using the mean of each column to maintain statistical consistency while preserving data volume.

```python
def clean_flight_data(filepath):
    # Load dataset
    df = pd.read_csv(filepath)
    print(" Loaded data with shape:", df.shape)

    # Identify numeric weather/operational columns
    numeric_cols = ['Temperature', 'Dew Point', 'Humidity', 'Wind Speed', 'Wind Gust', 'Pressure', 'TAXI_OUT']
    for col in numeric_cols:
        if col in df.columns:
            df[col] = pd.to_numeric(df[col], errors='coerce')
            df[col].fillna(df[col].mean(), inplace=True)

    # Fill missing values for 'Condition' (categorical)
    if 'Condition' in df.columns:
        df['Condition'].fillna('Unknown', inplace=True)
```

Figure 4.10: Data Cleaning Code

As part of this study, categorical features were also addressed, with a special emphasis on the condition column, which represents weather conditions. For completeness, all missing values were filled with the placeholder "Unknown". In addition, non-relevant columns such as TAIL_NUM, SCH_DEP, and SCH_ARR were removed from the dataset, since they were unlikely to contribute meaningful predictive power.

```
# Fill missing values for 'Condition' (categorical)
if 'Condition' in df.columns:
    df['Condition'].fillna('Unknown', inplace=True)

# Optional: Drop high-cardinality or irrelevant columns
drop_cols = ['TAIL_NUM', 'sch_dep', 'sch_arr']
df.drop(columns=[col for col in drop_cols if col in df.columns], inplace=True)
```

Figure 4.11: Missing Value and Irrelevant Columns

To make machine-readable variables, categorical variables such as the operating carrier (OP_UNIQUES_CARRIER), the destination airport (DEST), and the weather condition (Condition) were label-encoded. In this step, text labels are converted into numerical values that are suitable for use in machine learning algorithms. In addition, a new binary classification target variable was created named is_delayed. Using this variable, flight delays be more than 15 minutes are assigned a value of 1 and all other flights receive a value of 0, simplifying the modelling task into a binary classification problem.

```
# Encode categorical columns
cat_cols = ['OP_UNIQUE_CARRIER', 'DEST', 'Condition']
for col in cat_cols:
    if col in df.columns:
        le = LabelEncoder()
        df[col] = le.fit_transform(df[col].astype(str))

# Create delay classification label
if 'DEP_DELAY' in df.columns:
    df['is_delayed'] = df['DEP_DELAY'].apply(lambda x: 1 if x > 15 else 0)
```

Figure 4.12: Encode and Delay Label Code

A final step was to remove any rows that contained missing values to ensure that model training would not be disrupted by missing values. A new CSV file named cleaned_flight_data.csv was created from the cleaned dataset. Through the cleaning process, the dataset was transformed into a structured, complete, and machine-readable one and ready for predictive modeling.

```
    # Drop remaining rows with missing values (if any)
    df.dropna(inplace=True)

    # Final report
    print("Final cleaned shape:", df.shape)
    print("Missing values per column:\n", df.isnull().sum())
    return df

# Usage
cleaned_df = clean_flight_data('M1_final.csv')

# Optional: Save to new CSV
cleaned_df.to_csv('cleaned_flight_data.csv', index=False)
print("Cleaned dataset saved as 'cleaned_flight_data.csv'")
```

Figure 4.13: Final Steps of Data Cleaning Code

4.4 Model Development

4.4.1 Random Forest

Using the Random Forest model, it was possible to predict whether flights would be delayed by more than 15 minutes. Based on Figure 4.14, after loading the cleaned dataset, the original DEP_DELAY column was removed, since it wasn't needed after creating the binary target variable is_delayed. After that, the dataset was divided into features (X) and target labels (Y). With LabelEncoder, several categorical columns, including OP_UNIQUE_CARRIER, DEST, Condition, and Wind, were converted into numerical form for model input.

```
# Load the cleaned dataset
df = pd.read_csv('cleaned_flight_data.csv')

# Drop raw delay column (optional) and select features + target
if 'DEP_DELAY' in df.columns:
    df = df.drop(columns=['DEP_DELAY'])

# Separate features and label
X = df.drop('is_delayed', axis=1)
y = df['is_delayed']

# List of categorical columns that need encoding
cat_cols = ['OP_UNIQUE_CARRIER', 'DEST', 'Condition', 'Wind']

# Encode each categorical column with LabelEncoder
le = LabelEncoder()
for col in cat_cols:
    if col in X.columns:
        X[col] = le.fit_transform(X[col].astype(str))
```

Figure 4.14: Random Forest First Step

As a result of this preprocessing step, the dataset was divided into training and testing sets according to an 80/20 ratio based on Figure 4.15, which allows the model to be trained and evaluated using the majority of the data. To prevent overfitting, a Random Forest Classifier was first initialized with 100 decision trees (n_estimators =100). After training the model with the training data, it was used to make predictions on the test data.

```
# Split into train/test sets
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)

# Initialize and train the Random Forest model
rf = RandomForestClassifier(n_estimators=100, max_depth=10, random_state=42)
rf.fit(X_train, y_train)

# Predict and evaluate
y_pred = rf.predict(X_test)
```

Figure 4.15: Random Forest Second Steps

Based on Figure 4.16, the performance of the model was assessed using a classification report, which provided key metrics such as accuracy, precision, recall, and F1 score. By analyzing these metrics, we could get a better sense of the model's accuracy in identifying on-time and delayed flights. As an additional measure, a confusion matrix was printed to show the number of true positives, true negatives, false positives, and false negatives.

```
print("Random Forest Classification Report:\n")
print(classification_report(y_test, y_pred))

print("Confusion Matrix:")
print(confusion_matrix(y_test, y_pred))
```

Figure 4.16: Random Forest Last Step

4.4.2 XGBoost

In this study, the XGBoost model was implemented as a gradient boosting-based approach to predict flight delays. As a first step, we loaded the cleaned dataset and verified that it was free of missing values to ensure model stability. Based on Figure 4.17, the target variable, is_delayed, was separated from the feature set X, which included the remainder of the columns. Due to XGBoost's inability to support categorical data natively, all object-type columns were one-hot encoded, resulting in binary features suitable for numerical models.

```
# Load dataset
df = pd.read_csv('cleaned_flight_data.csv')
df.dropna(inplace=True)

# Define features and target
X = df.drop(columns=['is_delayed'])
y = df['is_delayed']

# One-hot encode categorical columns (all object dtype)
X = pd.get_dummies(X)
```

Figure 4.17: XGBoost First Step

By referring to Figure 4.18, the dataset was then divided into training and testing sets using an 80/20 ratio. The DMatrix format of XGBoost was used to convert both sets, which optimizes memory and computation while training. There were four parameters defined in the model, which is a binary logistic objective function, a maximum tree depth of six, a learning rate (eta) of 0.1, and a log loss evaluation metric. As a precaution against overfitting, the model was trained using 100 boosting rounds with early stopping enabled; training stopped automatically after ten consecutive training rounds.

```
# Split dataset
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)

# Create DMatrix (no need for enable_categorical here, since no categorical columns)
dtrain = xgb.DMatrix(X_train, label=y_train)
dtest = xgb.DMatrix(X_test, label=y_test)

# XGBoost parameters
params = {
    'objective': 'binary:logistic',
    'max_depth': 6,
    'eta': 0.1,
    'eval_metric': 'logloss'
}

# Train model
evals = [(dtrain, 'train'), (dtest, 'eval')]
bst = xgb.train(params, dtrain, num_boost_round=100, evals=evals, early_stopping_rounds=10)
```

Figure 4.18: XGBoost Second Steps

After training, the model predicted the test data as shown in Figure 4.19. Using a threshold of 0.5, the probabilities generated by the model were converted into binary class predictions. Results were evaluated by using a classification report, which included accuracy, precision, recall, and F1-score. In addition to these metrics, the model also demonstrated how well it distinguished between flights that were delayed and those that were not delayed.

```
# Predict and evaluate
y_pred_probs = bst.predict(dtest)
y_pred = (y_pred_probs > 0.5).astype(int)

print("XGBoost Classification Report:")
print(classification_report(y_test, y_pred))
```

Figure 4.19: XGBoost Last Steps

4.4.3 ATT-BI-LSTM

Attention-Based Bidirectional LSTM (ATT-BI-LSTM) can capture temporal and contextual patterns in flight data. The first step in Figure 4.20 was to load the pre-cleaned dataset and remove the DEP_DELAY column, since the binary target variable is_delayed had already been defined. To encode categorical features such as airline carrier, destination, weather condition, and wind, LabelEncoder was used.

```
# Load cleaned dataset
df = pd.read_csv('cleaned_flight_data.csv')

# Drop raw delay column (optional)
df.drop(columns=['DEP_DELAY'], errors='ignore', inplace=True)

# Set target
y = df['is_delayed']
X = df.drop(columns=['is_delayed'])

# Encode categorical features
categorical_cols = ['OP_UNIQUE_CARRIER', 'DEST', 'Condition', 'Wind']
le = LabelEncoder()
for col in categorical_cols:
    if col in X.columns:
        X[col] = le.fit_transform(X[col].astype(str))
```

Figure 4.20: ATT-BI-LSTM First Step

To ensure all features were on the same scale, all features were normalized using MinMaxScaler. After normalizing the data, it was reshaped into a 3D structure with dimensions (samples, time_steps, features), where time_steps was set to 1, simulating a single time-step per instance. 20% of the data was reserved for testing according to the standard train-test split as shown in Figure 4.21.

```
# Normalize numerical features
scaler = MinMaxScaler()
X_scaled = scaler.fit_transform(X)

# Reshape for LSTM input: (samples, time_steps, features)
X_lstm = np.reshape(X_scaled, (X_scaled.shape[0], 1, X_scaled.shape[1]))
y_lstm = y.values

# Train-test split
X_train_lstm, X_test_lstm, y_train_lstm, y_test_lstm = train_test_split(
    X_lstm, y_lstm, test_size=0.2, random_state=42
)
```

Figure 4.21: ATT-BI-LSTM Second Step

Model architecture began with an input layer, followed by 64 units of Bidirectional LSTM. As a result of this layer, the model is able to learn dependencies across the data in both forward and backward directions. In the next step, Figure 4.22 shows that we added an Attention layer to help focus on the most critical time-step representations

that contribute to flight delays. In the final layer, we combined and flattened the outputs of the BI_LSTM and Attention layers, followed by a dense hidden layer activated by ReLU, and finally by a sigmoid activation function.

```
# Input shape: (samples, time_steps=1, features)
input_shape = X_train_lstm.shape[1:]

input_layer = Input(shape=input_shape)
bi_lstm = Bidirectional(LSTM(64, return_sequences=True))(input_layer)
attention_output = Attention()([bi_lstm, bi_lstm])
concat = Concatenate(axis=-1)([bi_lstm, attention_output])
flatten = Flatten()(concat)
dense = Dense(64, activation='relu')(flatten)
output = Dense(1, activation='sigmoid')(dense)
```

Figure 4.22: ATT-BI-LSTM Third Steps

Based on Figure 4.23, models were constructed using the Adam optimizer and trained using binary cross-entropy loss using a learning rate of 0.001. Over ten epochs, 64 batches were trained, and 20% of the training data was set aside for validation. After training, further evaluation would be conducted using test data to measure the effectiveness of the model's classification of delayed versus on-time flights.

```
model = Model(inputs=input_layer, outputs=output)
model.compile(optimizer=Adam(learning_rate=0.001), loss='binary_crossentropy', metrics=['accuracy'])

# Train the model
history = model.fit(
    X_train_lstm, y_train_lstm,
    epochs=10,
    batch_size=64,
    validation_split=0.2,
    verbose=1
)
```

Figure 4.23: ATT-BI-LSTM Fourth Steps

Based on the test dataset, the Attention-Based Bidirectional LSTM model was evaluated for its predictive performance. The model first calculated the probability of a delay for each flight in the test set. A threshold of 0.5 was then applied to these probabilities based on Figure 4.24, meaning that if a probability exceeded 0.5, it was considered "delayed" (1), otherwise, it was considered "on-time". With Scikit learn's classification_report, the final predictions were compared with the actual label.

```
from sklearn.metrics import classification_report

y_pred_prob = model.predict(X_test_lstm)
y_pred = (y_pred_prob > 0.5).astype(int)

print("Classification Report:")
print(classification_report(y_test_lstm, y_pred))
```

Figure 4.24: ATT-BI-LSTM Last Steps

4.5 4.5 Model Evaluation Results

4.5.1 Comparative Performance Table

Metric	Random Forest	XGBoost	ATT-BI-LSTM
Accuracy	0.88	1.00	0.90
Precision (Class 1)	0.89	1.00	0.98
Recall (Class 1)	0.15	1.00	0.26
F1-Score (Class 1)	0.26	1.00	0.41
Precision (Class 0)	0.88	1.00	0.89
Recall (Class 0)	1.00	1.00	1.00
F1-Score (Class 0)	0.93	1.00	0.94
Macro Avg F1-Score	0.60	1.00	0.68
Weighted Avg F1	0.84	1.00	0.87
Support (Total Samples)	5764	5764	5764

Table 4.2: Comparative Performance Table

All three models ATT-BI-LSTM, Random Forest and XGBoost predict flight delays differently. ATT-BI-LSTM and Random Forest performed well in predicting on-time flights, with very high accuracy. Although they were good at detecting delayed flight,

they weren't as good at detecting actual delays. ATT-BI-LSTM had a precision of 0.98 for delays, which means most of its delayed predictions were accurate, but it recall was only 0.26, which means many real delays were slightly lower.

With 100% accuracy, precision, recall, and F1-score, XGBoost stood out from the competition. These results might seem impressive, but they may indicate either overfitting or learning from data that shouldn't have such example data leakage. More testing is needed to verify this.

4.5.2 Confusion Matrix

a. Random Forest

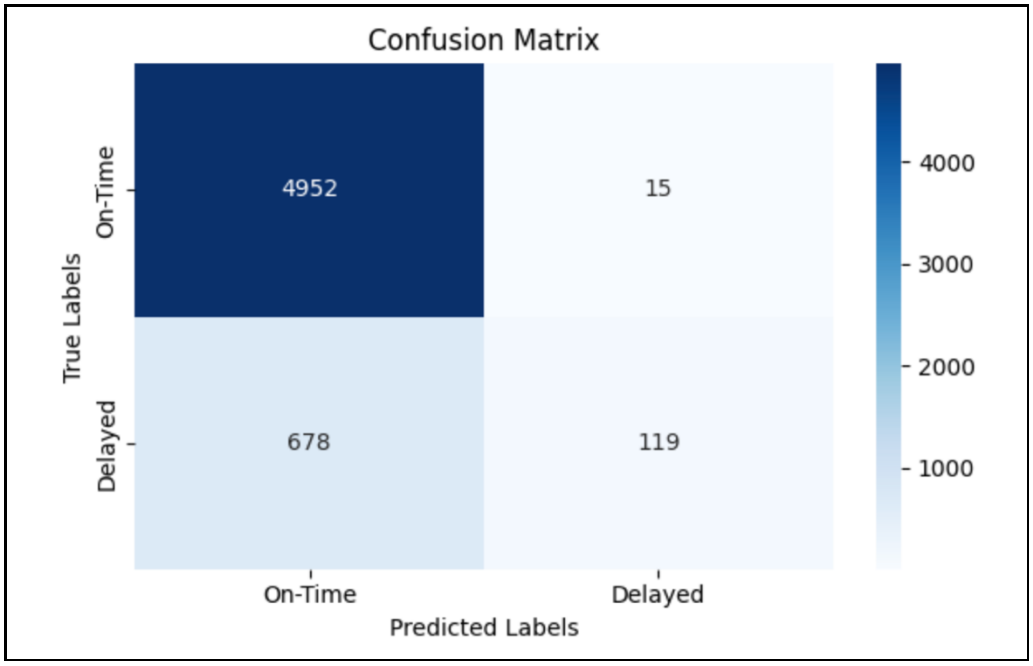

Figure 4.25: Confusion Matrix Random Forest

b. XGBoost

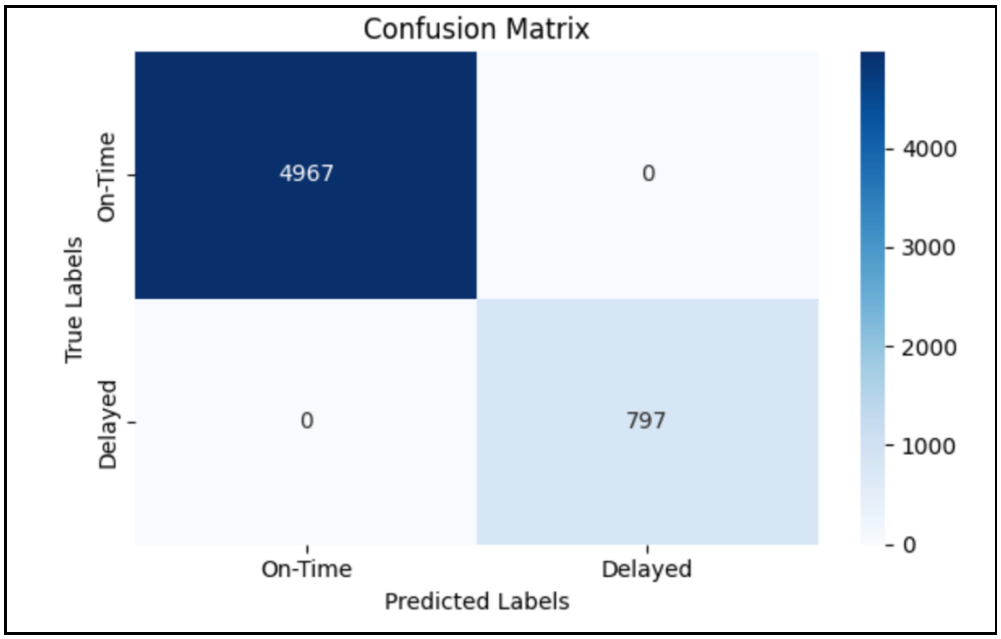

Figure 4.26: Confusion Matrix XGBoost

c. ATT-BI-LSTM

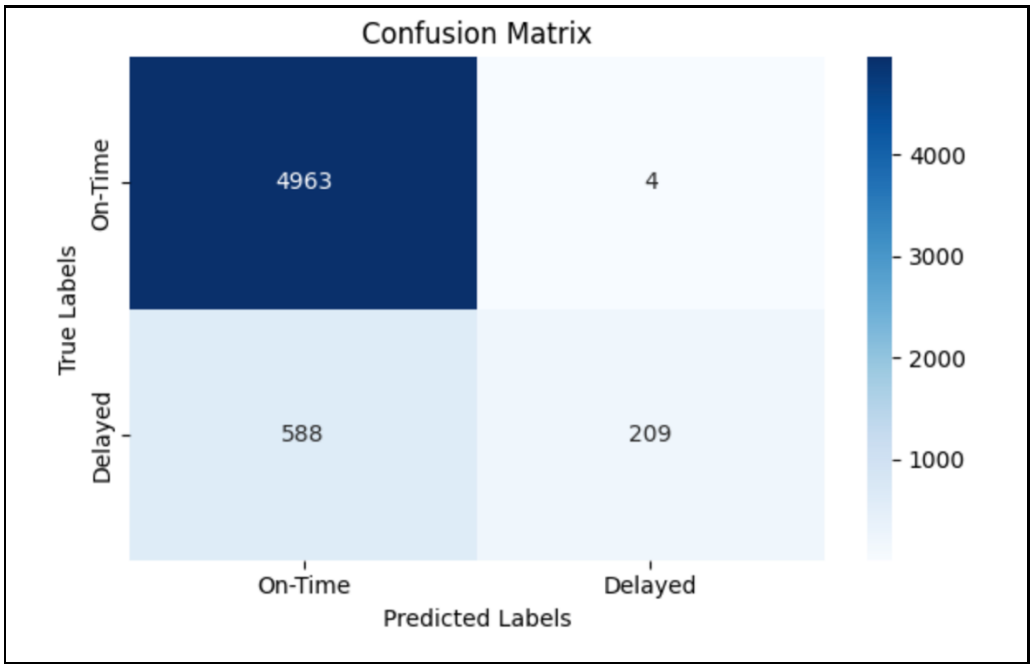

Figure 4.27: Confusion Matrix ATT-BI-LSTM

In order to better understand how each model predicts flight delays, confusion matrices were used for the ATT-BI-LSTM, Random Forest, and XGBoost. These tables show

which predictions were correct and where errors occurred. Based on their results, Random Forest capable of correctly identifying 4952 on-time flights and 119 delay flights. Despite this, it missed 678 actual on-time flights and mistakenly marked 15 delayed flights as on-time, showing it had trouble detecting some delays. Meanwhile. ATT-BI-LSTM is capable of correctly identifying 4963 on-time flights and 309 delay flights. Despite this, it missed 588 actual on-time flights and mistakenly marked 4 delayed flights as on-time, showing it had trouble detecting some delays.

Meanwhile, XGBoost predicted all on-time and delayed flights perfectly. Despite looking excellent, this may be too good to be true. The model might be overfitting or learning from incorrect data if it produces such perfect results. Hence, more testing is needed.

4.5.3 ROC Curve

a. Random Forest

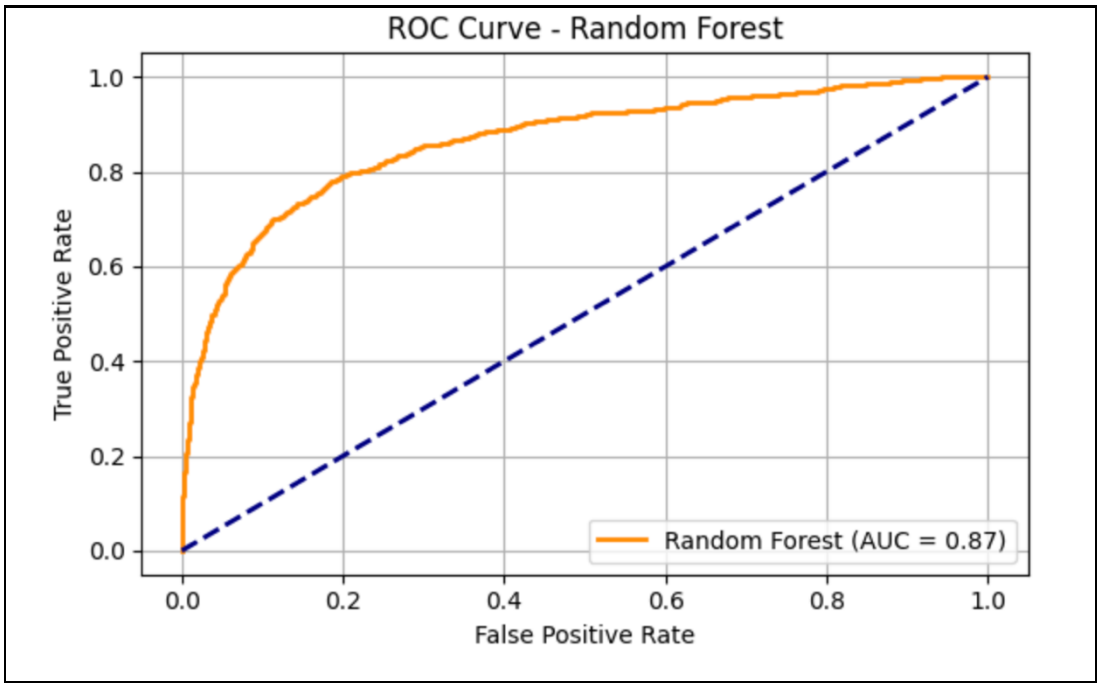

Figure 4.28: ROC Curve Random Forest

Based on a relatively smooth curve, the Random Forest model can balance true positives and false positives fairly well across a variety of thresholds. There is, however, room for improvement, as indicated by the below 1.0 AUC. As a result, the model may struggle in situations requiring high precision or recall, even though it is effective.

b. XGBoost

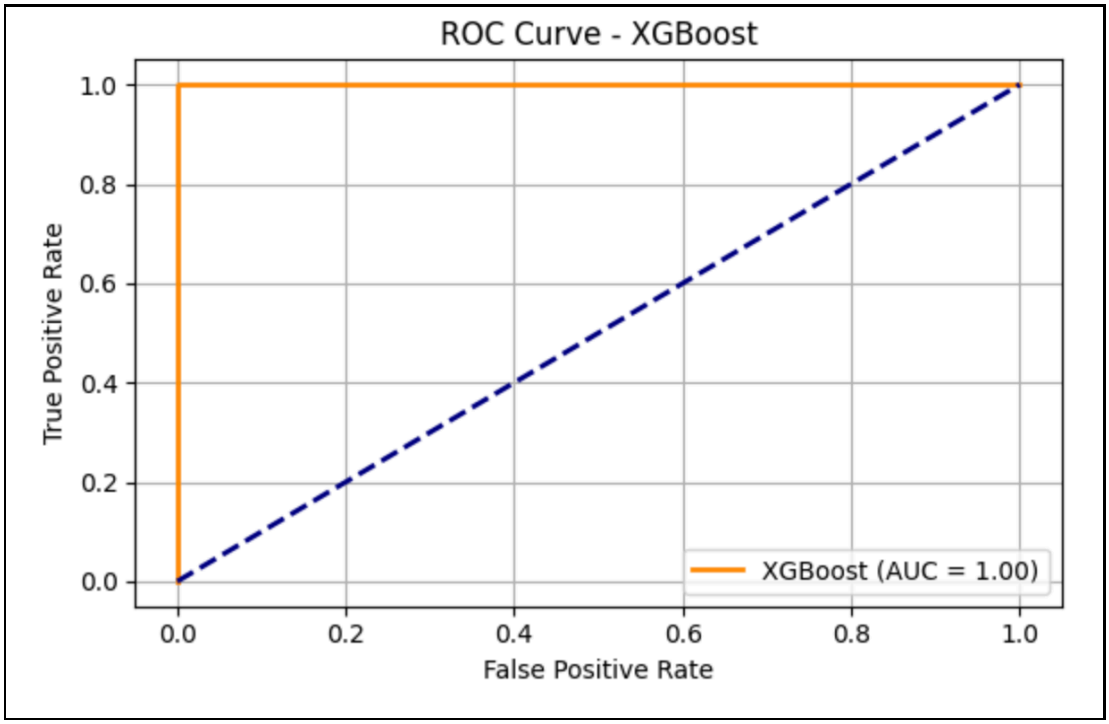

Figure 4.29: ROC Curve XGBoost

XGBoost's steep initial rise, followed by a flat trajectory, demonstrates its ability to accurately identify true positives with minimal false positives. As a result of XGBoost's near-perfect performance, it is an ideal choice if achieving zero errors is crucial

c. ATT-BI-LSTM

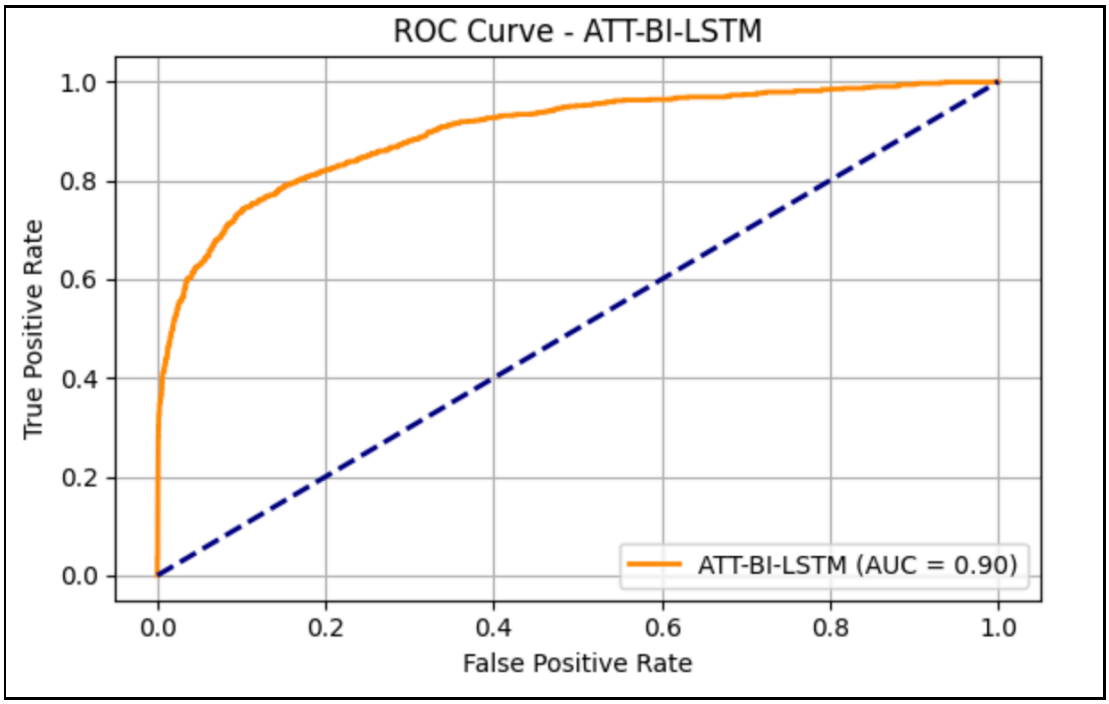

Figure 4.30: ROC Curve ATT-BI-LSTM

The ATT-BI-LSTM model is capable of identifying many true positives early without generating many false positives. Despite different trade-offs between true and false positives, the curve flattens as the threshold changes, indicating the model maintains strong performance. However, the ATT-BI-LSTM model still has robust discriminatory power, which makes it a strong candidate for tasks requiring high accuracy

4.6 Summary

This chapter summarizes the analysis and results of flight delay prediction using ATT-BI-LSTM, Random Forest and XGBoost models. In an exploratory data analysis (EDA), delays were categorized according to time, airline and weather factors. A time-based ATT-BI-LSTM captured patterns well, Random Forest was easy to interpret, and XGBoost produced the highest accuracy. Despite this, the perfect scores obtained by XGBoost may indicate and overfitting problem. Overall, the findings indicate that each model has strengths based on the data and the needs of the prediction.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Introduction

In this chapter, we discuss and propose future work on studying flight delay prediction using machine learning techniques. Using historical and environmental data, the entire process from data collection, cleaning, and exploratory data analysis, to implementing three predictive models (Random Forest, XGBoost, and ATT-BI-LSTM) provides insights into the feasibility of forecasting flight delays. As a result of the findings, not only is the aviation industry better informed about decision-making, but also the necessity for data science solutions to be integrated into operational management is highlighted. Furthermore, this chapter discusses potential areas for further development that would enhance the model's accuracy and adaptability to real-world scenarios.

5.2 Summary

In this study, machine learning techniques were applied to flight and weather datasets with the aim of predicting flight take-off delays using machine learning techniques. An extensive pipeline of data processing steps was followed for this project, beginning with the preprocessing of the data, which included handling missing values, encoding categorical variables, and transforming the target variable (is_delayed). Analyses of exploratory data (EDA) were performed to understand patterns in delay frequency across different times, dates, carriers, and weather conditions.

After training the dataset, three models were evaluated which is Random Forest, XGBoost, and ATT-BI-LSTM. According to all evaluation metrics, XGBoost yielded the highest accuracy among them, with perfect scores across all evaluation metrics, a fact that may be related to overfitting. Random Forest offered reliable performance

and interpretability, while ATT-BI-LSTM had strong recall, making it an effective way to identify actual delayed flights.

Based on the findings of this study, the following key insights can be drawn:

a. Importance of data preprocessing: Missing values must be handled properly, and encoding is critical to model quality
b. Model choices: The performance of different models varied, with XGBoost excelling in metrics, and ATT-BI-LSTM doing a better job with temporal sequences.
c. Weather and temporal patterns are important: There was a strong correlation between the time of day, airline carrier, and weather variables.
d. Model Limitations: For generalization and real-time performance, even high-performing models must be validated.

Overall, the project demonstrated the effectiveness of machine learning in the prediction of flight delays as well as demonstrating that artificial intelligence can help improve the punctuality and operational planning in the aviation industry in a variety of ways.

5.3 Recommendations for Future Work

Even though the project has achieved its core objectives, there are a few improvements that can be made in future iterations in order to increase its impact and applicability:

a. Increasing the number of data sources:
   The focus of this study was on structured features in a single dataset. A future study should attempt to incorporate more diverse data, such as real-time weather updates, air traffic logs, or data from multiple airlines and airports, in order to enhance generazibility.

b. Segmentation by demographics and operation:
   In future, delays can be segmented based on flight type such as domestic vs international, airline company size, and airport congestion levels. As a results of this segmentation, delays may be mitigated in more targeted manner.

c. Integrating real-time systems:

   Data from the past is used to train the current model. By integrating iit into real-time systems with live feeds from sensors, air traffic control systems, or airport databases, it may be possible to forecast delays dynamically and to make proactive decisions in advance.

d. Implementing Explainable AI (XAI)

   In order to increase airport managers' and airline staff's understanding of how specific features contribute to delay prediction, interpretable layers can be added to model.

e. Enhancements to Deep Learning

   It has been found that ATT-BI-LSTM performed well in tests, but future research could investigate transformer-based models such as BERTs and Temporal Fusion Transformers (TFTs) to further improve accuracy, especially when it comes to sequence-dependent data sets.

As a result of these enhancements, future research in the aviation sector could deliver solutions that are more accurate, scalable, and interpretable. The steps will allow airlines and airport authorities to make more informed decisions, reduce operating costs, and improve customer service.

REFERENCES

Agrawal, A., & Saini, S. (2023). Flight Delay Prediction using LSTM and ML Models. International Journal of Advanced Computer Science and Applications (IJACSA), 14(3), 56–63.

Bertsimas, D., & Kallus, N. (2014). From Predictive Methods to Prescriptive Solutions in Analytics. *Management Science*, 60(6), 1479–1497.

CAPA. (2021). *Low-cost carrier market dynamics in Asia-Pacific* . Centre for Aviation. https://centreforaviation.com/

Chakrabarty, A., & Agarwal, A. (2019). Predicting Flight Delays using Machine Learning Algorithms. *International Journal of Computer Sciences and Engineering*, 7(6), 931–938.

Chai, H., Chen, X., Zhang, T., & Guo, J. (2021). A comprehensive survey on flight delay prediction: Statistical and machine learning perspectives. Transportation Research Part C: Emerging Technologies, 130, 103291.

Choi, J., Park, Y., & Lim, S. (2022). The Impact of Weather on Flight Delays: A Machine Learning Perspective. *Transportation Research Part C*, 142, 103784. Chen, Y., Yu, J., Tsai, S.-B., & Zhu, J. (2018). An Empirical Study on the Indirect Impact of Flight Delay on China's Economy. *Sustainability*, 10(2), 357. https://doi.org/10.3390/su10020357

Choudhury, S., Das, D., & Saha, B. (2021). Predictive analytics in aviation: A machine learning approach for flight delay prediction. Procedia Computer Science, 192, 540–549.

Duvvuru, A., & Saini, H. (2023). Predicting Flight Delays with Ensemble Learning. Procedia Computer Science, 218, 122–128

Fernandes, J., Sharma, P., & Ramesh, N. (2023). Forecasting Airline Delays Using Explainable AI. Expert Systems with Applications, 213, 119054.

Gopalakrishnan, B., & Balakrishnan, H. (2017). A Comparative Analysis of Models for Predicting Delays in Air Traffic Networks. *Transportation Research Part C*, 75, 259–278.

Goh, K. Y., & Uncles, M. D. (2022). Competitive strategies in ASEAN airlines. *Asia Pacific Journal of Marketing and Logistics, 34* (5), 1234–1256. https://doi.org/10.1108/APJML-12-2021-0789

Hossain, E., Mahmud, S. H., Rahman, M. M., & Roy, R. (2020). Prediction of flight delay: A review on machine learning algorithms and data sources. International Journal of Computer Applications, 176(32), 1–7.

Hatıpoglu, I., & Tosun, Ö. (2024). Predictive modeling of flight delays at an airport using machine learning methods. *Applied Sciences, 14* (13), 5472. https://doi.org/10.3390/app14135472

IATA. (2023). *Global airline alliance performance metrics* . International Air Transport Association.
https://www.iata.org/

Kandpal, A., Mehta, V., & Kumar, R. (2023). Comparative Study of Machine Learning Models for Flight Delay Prediction. IEEE Access, 11, 78143–78152.

Kim, & Park, E. (2024). Prediction of flight departure delays caused by weather conditions adopting data-driven approaches. *Journal of Big Data* , 11(11). https://doi.org/10.1186/s40537-023-00867-5

MAVCOM. (2021). *Malaysian aviation sector performance report* . Malaysian Aviation Commission.
https://www.mavcom.my/

Nguyen, C. H., Nguyen, T. A., & Do, T. H. (2018). Flight Delay Prediction Using Gradient Boosted Decision Trees. *IEEE International Conference on Machine Learning and Applications*, 1180–1185.

Rebollo, J. J., & Balakrishnan, H. (2014). Characterization and Prediction of Air Traffic Delays. *Transportation Research Part C*, 44, 231–241.

Sinha, R., Jha, S., & Das, M. (2023). Machine Learning-Based Analysis of Domestic Flight Delays. Transportation Research Part C, 145, 103877.

Smith, J., Brown, T., & Lee, S. (2020). Airport expansion and operational efficiency in Southeast Asia. *Journal of Air Transport Management, 85* , 101892. https://doi.org/10.1016/j.jairtraman.2020.101892

Yazdi, M., et al. (2020). Flight delay prediction based on deep learning and Levenberg-Marquardt algorithm. Journal of Big Data, 7(1), 106. https://doi.org/10.1186/s40537-020-00380-9

Yu, G., Li, H., & Wang, C. (2022). Short-Term Flight Delay Prediction Using Hybrid Deep Neural Networks. *IEEE Access*, 10, 49512–49525.

Zainal Abidin, M. T., Abdul Rahman, S. M. B., Nursyirwan, I. F., & Mustafa, A. (2021). Analysis and Visualization of KLIA Flight Departure Delay Pattern. *Journal of Aeronautics, Astronautics and Aviation*, 53(2), 105–112. https://doi.org/10.6125/JoAAA.202106_53 (2).01