INSTITUTO TECNOLÓGICO DE AERONÁUTICA

Bruno Monteiro Franzini

A *LIGHTGBM* MODEL FOR RUNWAY CONFIGURATION PREDICTION APPLIED TO CONGONHAS AIRPORT

Final Paper
2024

Course of Computer Engineering

CDU 681.3:16

Bruno Monteiro Franzini

A *LIGHTGBM* MODEL FOR RUNWAY CONFIGURATION PREDICTION APPLIED TO CONGONHAS AIRPORT

Advisor

Prof. Dr. Marcos Ricardo Omena de Albuquerque Máximo (ITA)

Co-advisor

Cap Eng Me. André Luiz Elias Melo (ICEA)

Co-advisor

Me. Jean Phelipe de Oliveira Lima (ICEA)

COMPUTER ENGINEERING

São José dos Campos
Instituto Tecnológico de Aeronáutica

2024

Cataloging-in-Publication Data
Documentation and Information Division

Franzini, Bruno Monteiro
A *LightGBM* Model for Runway Configuration Prediction Applied to Congonhas Airport / Bruno Monteiro Franzini.
São José dos Campos, 2024.
63f.

Final paper (Undergraduate Study) – Course of Computer Engineering– Instituto Tecnológico de Aeronáutica, 2024. Advisor: Prof. Dr. Marcos Ricardo Omena de Albuquerque Máximo. Co-advisors: Cap Eng Me. André Luiz Elias Melo and Me. Jean Phelipe de Oliveira Lima.

I. Instituto Tecnológico de Aeronáutica. II. Title.

BIBLIOGRAPHIC REFERENCE

FRANZINI, Bruno Monteiro. **A *LightGBM* Model for Runway Configuration Prediction Applied to Congonhas Airport**. 2024. 63f. Final paper (Undergraduate Study) – Instituto Tecnológico de Aeronáutica, São José dos Campos.

CESSION OF RIGHTS

Bruno M. Franzini

Bruno Monteiro Franzini
Rua H8A, Ap. 125
12.228-460 – São José dos Campos–SP

A *LIGHTGBM* MODEL FOR RUNWAY CONFIGURATION PREDICTION APPLIED TO CONGONHAS AIRPORT

This publication was accepted as Final Work of Undergraduate Study

Bruno M. Franzini

Bruno Monteiro Franzini

Marcos Ricardo Omena de Albuquerque Maximo

Marcos Ricardo Omena de Albuquerque Máximo (ITA)

Advisor

André Melo

André Luiz Elias Melo (ICEA)

Co-advisor

Jean Phelipe de O Lima

Jean Phelipe de Oliveira Lima (ICEA)

Co-advisor

Marcos Ricardo Omena de Albuquerque Maximo

Prof. Dr. Marcos Ricardo Omena de Albuquerque Máximo
Course Coordinator of Computer Engineering

São José dos Campos: November 08, 2024.

To my grandfather José, who unfortunately is not here to witness this achievement, but who was fundamental in helping me get to this point.

Acknowledgments

First and foremost, I would like to thank God, for without Him, none of this would be possible.

To my parents, for their unconditional support from the moment, ITA was just a dream until this final stage.

To my family, in general, for always encouraging and believing in me.

To my girlfriend, Giovanna, for being my support and companion at every moment.

To Prof. Dr. Marcos Máximo for guiding me in the best way possible in developing this work.

To Cap Melo, Jean Lima, and the entire ICEA team for always being willing to help and provide the necessary support.

To all my friends I made during my undergraduate studies, especially those in my apartment, without whom this journey would have been much more difficult.

Lastly, I also want to thank the Brazilian Air Force, which enabled me to pursue a top-level education and enjoy an unparalleled campus in Brazil.

"So that at the name of Jesus, every knee should bow, in heaven, on earth, and under the earth, and every tongue confess that Jesus Christ is Lord, to the glory of God the Father."

— PHILIPPIANS 2:10-11

Resumo

A complexidade e a criticidade do controle do espaço aéreo, juntamente com a crescente demanda por voos, destacam a necessidade de ferramentas avançadas para aprimorar a otimização e a segurança das operações. Dadas as inúmeras variáveis que influenciam o tráfego aéreo e a dificuldade de avaliar analiticamente o impacto de cada uma, as técnicas de Inteligência Artificial (IA) oferecem uma estratégia viável para ponderar cada fator com mais precisão. Considerando que a mudança na configuração de pista é uma das variáveis mais influentes na gestão do tráfego aéreo, este trabalho apresenta o desenvolvimento de um sistema para prever a configuração de pista no Aeroporto de Congonhas (SBSP) utilizando técnicas de *Gradient Boosting*, especificamente o LightGBM. O modelo de classificação baseado em LightGBM desenvolvido utiliza dados do *Meteorological Aerodrome Report* (METAR), *Terminal Aerodrome Forecast* (TAF) e *Weather Research and Forecasting* (WRF) para prever potenciais reconfigurações de pista, determinando em qual sentido a pista estará operando. As previsões são feitas para 12 instantes futuros, espaçados de 15 minutos, oferecendo previsões desde os próximos 15 minutos até 3 horas à frente. O modelo alcançou uma acurácia de 98,6% no conjunto de teste e, quando testado em um período inédito, fora do conjunto de treino e teste, manteve uma alta acurácia de 92,0% com um F1-Score de 91,8%. Esses resultados destacam a eficácia do modelo desenvolvido, superando os métodos tradicionais baseados em regras atualmente em uso no ICEA (Instituto de Controle do Espaço Aéreo), que alcançam aproximadamente 79% de acurácia. O objetivo é proporcionar uma quantificação mais precisa da influência das variáveis meteorológicas nas operações aeroportuárias, contribuindo para um planejamento mais eficiente e seguro das atividades no aeroporto.

Abstract

The complexity and criticality of airspace control, combined with the increasing demand for flights, underscore the need for advanced tools to enhance the optimization and safety of operations. Given the numerous variables that influence air traffic and the difficulty of analytically assessing the impact of each, Artificial Intelligence (AI) techniques provide a viable strategy for more accurately weighting each factor. Considering that runway configuration change is one of the most influential variables in air traffic management, this work presents the development of a system to predict runway configuration at Congonhas Airport (SBSP) using Gradient Boosting techniques, specifically LightGBM. The developed LightGBM-based classification model utilizes data from the Meteorological Aerodrome Report (METAR), Terminal Aerodrome Forecast (TAF), and Weather Research and Forecasting (WRF) to predict potential runway reconfigurations, determining which side of the runway will be in use. Predictions are made for 12 future time points, spaced 15 minutes apart, providing forecasts from the next 15 minutes up to 3 hours ahead. The model achieved a 98.6% accuracy on the test set and, when tested on an unseen data period outside the training and test set, maintained a high accuracy of 92.0% with an F1-Score of 91.8%. These results highlight the effectiveness of the developed model, outperforming traditional rule-based methods currently in use at ICEA (Institute of Airspace Control), which achieve approximately 79% accuracy. The objective is to provide a more precise quantification of meteorological variables' influence on airport operations, contributing to more efficient and safer airport activity planning.

List of Figures

List of Tables

List of Abbreviations and Acronyms

ACC	Area Control Center
ADC	Aerodrome Chart
APP	Approach Control
ATFM	Air Traffic Flow Management
CGH	Congonhas Airport (IATA)
CPU	Central Processing Unit
DECEA	Department of Airspace Control
GPU	Graphics Processing Unit
GRU	Guarulhos Airport (IATA)
IA	Artificial Intelligence (AI)
IATA	International Air Transport Association
ICA	Air Force Command Instruction
ICAO	International Civil Aviation Organization
ICEA	Airspace Control Institute
LightGBM	Light Gradient Boosting Machine
METAR	Meteorological Aerodrome Report
SBGR	Guarulhos Airport (ICAO)
SBSP	Congonhas Airport (ICAO)
SCW	Severe Convective Weather
SHAP	SHapley Additive exPlanations
TAF	Terminal Aerodrome Forecast
TWR	Tower Control
WRF	Weather Research and Forecasting

Contents

1 Introduction

In this section, we describe the current state of air traffic management, the challenges involved, and how this work aims to contribute to addressing one of these challenges.

1.1 Motivation

Airspace control is a significant engineering challenge due to its complexity, the numerous variables to consider, and the critical nature of the problem, given the associated risks. Additionally, the current trend shows a continuous increase in demand for flights: it is estimated that the demand for air transportation will grow by approximately 4.3% per year over the next 20 years (WORLD, 2023a), driven both by societal development and the reduced distances brought about by globalization.

This growing demand from the population necessitates that regulatory agencies and relevant authorities worldwide adapt to manage an increasing volume of aircraft flying simultaneously (WORLD, 2023b). However, one of the primary challenges in airspace control is balancing two often conflicting factors: safety and efficiency. Generally, an extremely safe operation with high safety margins tends to be less efficient, while a flexible operation that allows more simultaneous flights can be risky.

Another aspect to consider in this context is the economic chain within the civil aviation market, given that many individuals can be positively or negatively affected depending on whether airspace is used efficiently. In addition to the more than 112 million passengers who flew in Brazil in 2023 (PODER360, 2023), nearly 1 million people are employed in this sector (IATA, 2018). Greater efficiency in airspace usage leads to lower operational costs for airlines and, consequently, lower ticket prices for consumers. Conversely, inefficient use — where delays and holding patterns are frequent — increases costs, negatively impacting both airlines, whose operations are hindered, and consumers, who face higher ticket prices. It is estimated that the average cost of in-flight delays is 100 euros per minute (EUROCONTROL, 2015). Thus, considering both the rising demand and the economic impacts caused by inefficient airspace usage, it is evident that optimizing its use is essential to maximize efficiency.

As previously described, numerous variables can cause a flight to deviate from the planned schedule, resulting in delays, for example. Weather conditions, runway reconfigurations, and capacity degradations at adjacent airports are just a few examples of factors that impact flight punctuality. Ideally, all these elements should be considered in air traffic management decision-making. However, this is challenging for a human, as quantifying the influence of each of these factors would require massive, real-time data processing. Furthermore, the mathematical modeling needed to solve this problem analytically is highly complex, as demonstrated by Prandini *et al.* (2011), due to the difficulty of capturing dependencies among various variables while accounting for variability and uncertainty.

In this context, an approach that provides a better quantification of each variable's influence is artificial intelligence (AI), specifically machine learning. Through training with historical data, machine learning algorithms can extract statistical patterns from the data and consequently predict, with greater accuracy and foresight than humans, when these issues might affect a flight. Recent studies have utilized AI algorithms in this context to address specific problems, which will be detailed in Section 1.4, demonstrating that these methods can enhance the performance of Air Traffic Flow Management (ATFM).

At this point, it has been identified that one of the main challenges in air traffic management, and one of the most influential factors in determining whether an aircraft will land as scheduled, is runway reconfiguration, commonly known as "runway configuration change". Figure 1.1 displays the Aerodrome Chart (ADC) for Congonhas, showing the two runway directions (17 and 35), which are essentially the two directions on the runway that aircraft can use for takeoff or landing, depending on the prevailing weather conditions.

Given this scenario, the importance of having a forecast that indicates, in advance, when runway reconfiguration will occur becomes evident. Such a forecast would allow airport operations to prepare accordingly, minimizing negative impacts on flights, such as holding patterns, delays, and congestion in the terminal area.

1.2 Background

The Airspace Control Institute (ICEA), a division under the Department of Airspace Control (DECEA), aims to support the continuous advancement of this sector through its Research Division. In response to the ongoing increase in demand from the market and the public, ICEA, as the state's representative in this context, has been working on solutions to improve the efficiency of airspace usage.

As mentioned, dealing with runway reconfiguration is one of the most significant challenges in air traffic management. For this reason, ICEA has already developed a runway configuration predictor for several aerodromes, which is currently operational. However, this prediction is based solely on rules that relate values obtained from the WRF (Weather Research and Forecasting), as shown in Figure 1.2. These rules follow the standard set by DECEA to ensure operational safety, as outlined in the Aeronautical Command Instruction (ICA) 100-37/2020 (DECEA, 2020a).

This current predictor achieves approximately 79% accuracy, based on its performance over ten months. Nevertheless, this level of accuracy is still insufficient to provide the necessary support to air traffic controllers, as this is a high-risk activity. Therefore, a more accurate predictor is required. Additionally, as discussed in the Motivation section, it is challenging for humans to quantify the impact of each variable (wind, visibility, clouds, etc.) on the outcome of which runway direction will be used. This highlights the importance of developing a predictor based on machine learning techniques that can better understand the influence of each factor and outperform the current method.

1.3 Objectives

In line with the discussions in the previous sections, the objective of this study is to develop a model based on AI techniques to accurately predict, up to three hours in advance, which runway configuration will be in use at the Congonhas Airport.

To achieve this general objective, the following specific objectives are set:

- Collect WRF, METAR (Meteorological Aerodrome Report), TAF (Terminal Aerodrome Forecast), weather station data, and the historical records of runway configuration changes (target variable);
- Process these data so they can be used as input for the model;
- Evaluate the performance of different AI models and select the most suitable one;
- Forecast the runway configuration up to three hours ahead in 15-minute intervals, providing predictions for 12 future time points, which correspond to predictions from 15 minutes up to three hours.

1.4 Literature Review

As highlighted in the previous sections, air traffic management is a complex task due to the multitude of variables involved, and AI can be a powerful tool in addressing such problems. In this regard, numerous studies have explored the application of AI techniques in this context, employing different approaches.

In an early, broader exploration, Gosling (1987) proposes various areas within the scope of airspace control where AI could contribute and yield positive outcomes. The author identifies seven possible control strategies, ranging from visual and electronic collision prevention to proposed improvements in the U.S. air traffic control systems, in which aircraft follow predefined, non-conflicting trajectories. AI applications are categorized into seven functional areas, discussing ways to incorporate these applications into different control strategies while also considering implementation issues that may arise during the application of AI techniques in air traffic control.

Since the publication of this work, the aviation industry has evolved significantly, with an increasing amount of data now available on flights. This has led some studies to attempt to extract insights from this data to assist in air traffic management. In this regard, Murça e Hansman (2019) presents a data-driven framework to identify, characterize, and predict traffic flow patterns in the terminal area of multiple airport systems, known as metroplexes. This framework employs machine learning methods applied to historical flight trajectories, weather forecasts, and airport operational data. Through a multi-layer clustering analysis, the study identifies recurring runway and airspace usage patterns, along with relevant determining factors, and develops descriptive models to predict metroplex configuration and capacity estimation. Applied to the New York metroplex, this framework demonstrates high accuracy in forecasting traffic flow patterns for planning horizons of up to eight hours, underscoring the importance of metroplex configuration prediction for better flow rate planning and, consequently, improved traffic regulation.

Additionally, other studies emphasize the substantial influence that external factors have on flight planning, further complicating air traffic management. Oliveira *et al.* (2021) investigates the impacts of weather conditions on the punctuality of domestic flights in Brazil. Using historical data on flight schedules and weather, the authors estimate a logit model to analyze the effects of variables such as visibility, cloud ceiling height, wind gusts, and precipitation on the likelihood of delays. The study concludes that adverse weather conditions significantly increase the probability of delays, contributing to a better understanding and management of weather impacts on flight punctuality.

In the context of runway conditions, Midtfjord *et al.* (2021) developed a decision support system for safer aircraft landings, using XGBoost to predict these conditions. This system combines classification and regression models trained with weather and runway data, achieving high accuracy and providing explainable predictions through techniques such as SHAP (SHapley Additive exPlanations).

Another significant study is that of Herrema *et al.* (2019), which also uses Gradient Boosting to predict runway exit times in Vienna, specifically how long an aircraft will take to vacate the runway after landing. By applying the technique to 54,679 arriving flights and analyzing scenarios that impact runway usage, this model achieved an accuracy of 79%, demonstrating the applicability of Gradient Boosting in airport contexts for predicting operational changes and improving efficiency.

Finally, Lau (2021) focuses specifically on predicting transition times during runway reconfiguration, using ensemble methods such as Random Forest Regressor, AdaBoost, and Gradient Tree Boosting. The study showed that these methods could achieve R^2 scores of at least 0.8, providing accurate predictions for runway configuration changes based on dynamic conditions like wind and cloud cover.

1.5 Contributions

During the development of this work, we decided to consolidate part of it as a research paper, which was published at the 2024 edition of the Air Transport Symposium (XXI SITRAER), held in Fortaleza-CE. In this article, Franzini *et al.* (2024) contributed to the field by developing a predictive system based on a Gradient Boosting technique, specifically LightGBM, to forecast runway configuration at Congonhas Airport (SBSP). The model, trained on historical data from the Meteorological Aerodrome Report (METAR), Terminal Aerodrome Forecast (TAF), and Weather Research and Forecasting (WRF), demonstrated the ability to accurately predict which side of the runway would be in use in the coming hours. Achieving a prediction accuracy of 98% on the test dataset and 88% on an unseen validation period outside the training data, the model outperformed traditional rule-based methods, currently used by ICEA (Institute of Airspace Control), which typically achieve around 79% accuracy. This research provides a more precise quantification of the influence of meteorological variables on airspace operations, as indicated by the feature importance analysis. Such insights can support more effective and safer planning for airspace activities, showcasing the potential of Machine Learning to enhance complex operational scenarios in air traffic management.

1.6 Scope

This work focuses on developing a predictive model based on artificial intelligence (AI) techniques to accurately forecast which runway configuration will be in use within a three-hour horizon at Congonhas Airport (SBSP). The proposed model will use current meteorological data (METAR) and forecast data (TAF and WRF) and will be trained to identify patterns that precede the need for change in runway configuration.

1.7 Organization of the Work

The subsequent chapters of this work are organized as follows:

- Chapter 2 explains the dynamics of Congonhas Airport and runway configuration change;
- Chapter 3 provides the theoretical background necessary to understand the AI implementation;
- Chapter 4 describes how we modeled the problem and the main tasks we undertook to address it;
- Chapter 5 presents the results obtained from various perspectives;
- Chapter 6 details the conclusions drawn and outlines possibilities for future work.

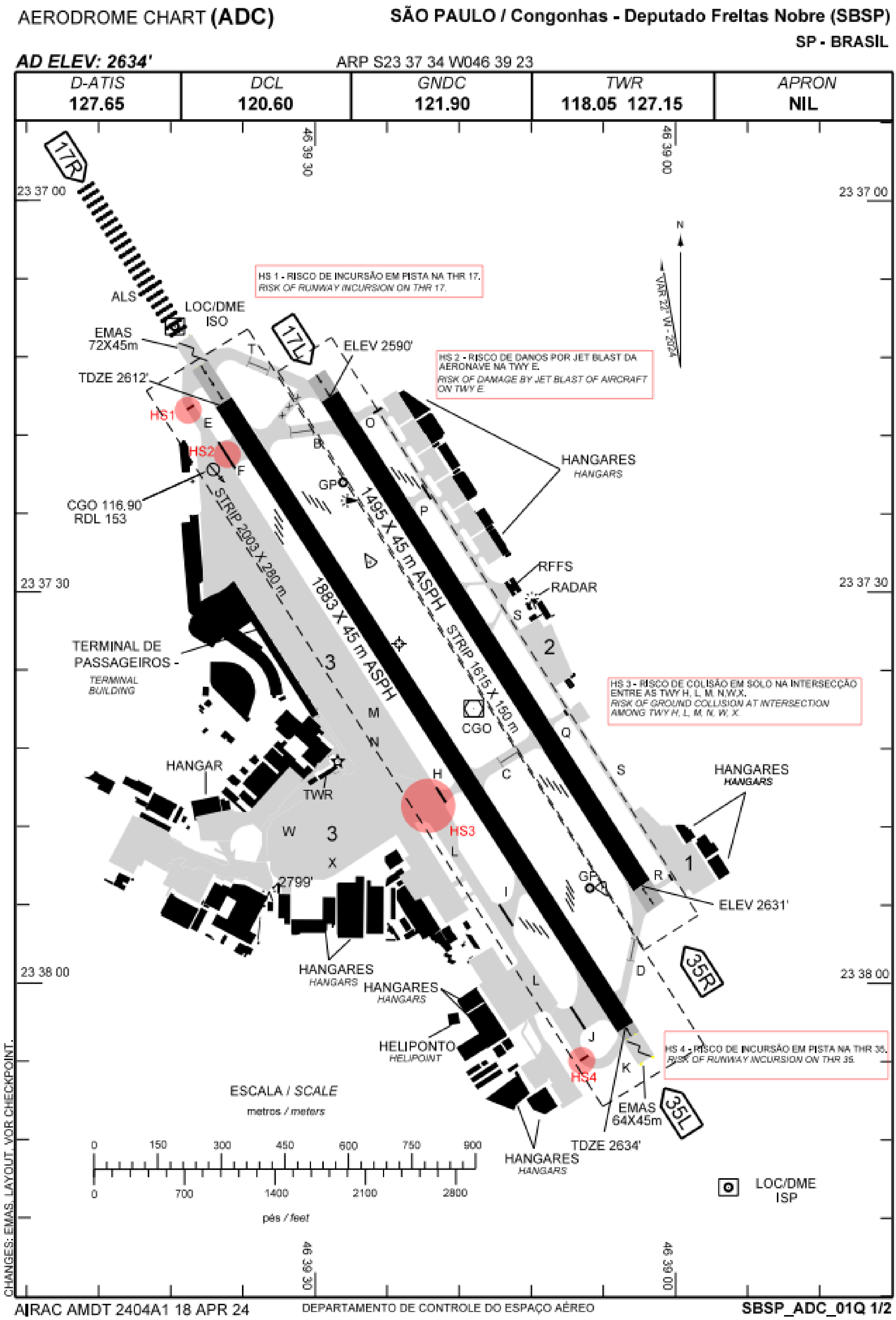

FIGURE 1.1 – Aerodrome Chart (ADC) of Congonhas showing the configuration of runway directions (17 and 35) and their respective runways (R and L). Source: AISWEB/DECEA.

FIGURE 1.2 – Current rule-based runway configuration predictor developed by ICEA. Source: (ICEA, 2024).

2 Dynamics of Congonhas Airport and Runway Configuration Change

In this section, we provide context on airport operations at Congonhas, detailing how runway configuration change, also known as runway reconfiguration, occurs and the agents involved.

2.1 History and Importance of Congonhas Airport

The São Paulo/Congonhas – Deputado Freitas Nobre Airport (IATA: CGH, ICAO: SBSP), popularly known as Congonhas Airport, is one of the oldest and most important airports in Brazil. Inaugurated in 1936, Congonhas has played a crucial role in developing commercial aviation in the country. Located in the southern part of São Paulo, approximately 9 km from the city center, the airport is known for its proximity to the urban area, facilitating passenger access and contributing to its popularity.

Over the years, Congonhas has undergone several expansions and modernizations to keep pace with growing air traffic. It is currently the second busiest airport in São Paulo, surpassed only by São Paulo/Guarulhos International Airport (IATA: GRU, ICAO: SBGR). Congonhas is a strategic hub for domestic flights, connecting São Paulo to various capitals and major cities across Brazil.

Its economic importance is undeniable, as it is a key facilitator for business, tourism, and cargo transport. The airport generates thousands of direct and indirect jobs and significantly contributes to the local and national economy. Its proximity to São Paulo's central region makes Congonhas a preferred choice for executives and business travelers seeking speed and convenience.

2.2 Physical and Operational Structure

Congonhas Airport has two main runways: runway 17R/35L and runway 17L/35R. These runways are oriented north-south and are used alternately depending on weather and operational conditions. Runway 17R/35L is 1,883 meters long, while runway 17L/35R is 1,345 meters long and 45 meters wide (AENA BRASIL, 2024). Figure 2.1 shows a map of Congonhas Airport, highlighting the runways and runway directions.

FIGURE 2.1 – Runway and direction configuration at Congonhas Airport. Source: Google Maps.

The airport infrastructure includes a modern passenger terminal with various amenities for travelers, such as restaurants, shops, banking services, and comfortable waiting areas. Additionally, Congonhas has facilities for aircraft maintenance, hangars, and designated areas for air taxis and general aviation.

From an operational perspective, Congonhas Airport is known for its intensive operations and high flight turnover. On average, the airport handles around 600 flights daily, serving more than 70,000 passengers per day and over 20 million per year (AENA BRASIL, 2024). This intense activity requires meticulous coordination and efficient planning to ensure flight safety and punctuality.

Aena manages the airport and oversees its administration, maintenance, and operation. It collaborates with various regulatory agencies and civil aviation authorities to ensure that Congonhas operates according to the highest safety and efficiency standards.

2.3 Agents Involved in Airspace Control

Before detailing the runway reconfiguration process in the next section, we must explain the primary agents involved in air traffic control operations and their respective responsibilities, according to DECEA (2020a).

2.3.1 ACC

The ACC (Area Control Center) controls traffic over a vast region of airspace, generally above a certain altitude where the main airways are located. Its goal is to ensure safe separation between aircraft in cruising flights, both within national borders and on international routes. This is often achieved by instructing aircraft on flight level, direction, and speed to maintain the desired spacing.

2.3.2 APP

The APP (Approach Control) manages air traffic near an airport, covering approximately 40 nautical miles around the aerodrome. It coordinates the arrival and departure phases of flights, ensuring that aircraft align correctly for landing or depart safely after takeoff.

2.3.3 TWR

The TWR (Tower Control) controls air traffic directly within the airport area, including runways and taxiways. The tower oversees takeoff and landing operations and aircraft movement on the ground.

2.4 Runway Reconfiguration

Runway reconfiguration - or runway configuration change - is essential in air traffic management, especially at busy airports like Congonhas. This process involves changing the direction of landing and takeoff operations to the opposite end of the runway due to shifts in weather conditions, such as wind direction. Runway reconfiguration is critical to maintaining the safety of air operations, as aircraft ideally take off and land into the wind to achieve adequate lift and ensure safe maneuvers. Therefore, one of the main triggers for runway reconfiguration is tailwind — wind blowing in the same direction as the aircraft.

2.4.1 Runway Reconfiguration Process

1. **Monitoring Weather Conditions**:

   Air traffic controllers and airport operators continuously monitor weather conditions, particularly wind direction and speed, using an anemometer, an essential instrument for these measurements. Aircraft typically land or take off into the wind unless operational safety conditions, runway configuration, weather conditions, the availability of instrument approach procedures, or traffic conditions indicate that a different direction is preferable (DECEA, 2020a).

2. **Decision-Making**:

   When selecting the runway in use, the tower control (TWR) must consider factors such as aerodrome traffic patterns, runway length, and available approach and landing aids. When surface wind is below 10 km/h (6 knots), aircraft are generally instructed to use the runway with greater operational advantages, such as larger dimensions or shorter taxiing distance. Surface wind direction and speed are always communicated to aircraft (DECEA, 2020a).

3. **Communication**:

   Tower control (TWR) must keep the approach control (APP) continuously informed about the runway selection. During instrument approach operations, TWR should not change the runway in use without proper coordination with APP (DECEA, 2020a). TWR and APP communicate any runway reconfigurations to the aircraft.

2.4.2 Challenges and Impacts

Runway reconfiguration can be problematic due to several factors:

1. **Operational Delays**:

   Changing the direction of operations can cause delays, as reconfiguration requires time for communication and operational adjustments (SILVA *et al.*, 2023). At a busy airport like Congonhas, this can lead to congestion and increased delay times for takeoffs and landings.

2. **Impact on Efficiency**:

   Runway reconfiguration can reduce operational efficiency, as aircraft need to reposition, and takeoff and landing intervals may need to be rescheduled. This can affect flight punctuality (SILVA *et al.*, 2023) and consequently increase the operational costs for airlines (EUROCONTROL, 2015).

3. **Safety**:

   Although runway reconfiguration enhances safety, the process itself must be managed precisely to avoid risks during the transition.

4. **Coordination**:

   Coordination among TWR, APP, pilots, ground crews, and airlines is crucial and challenging. A lack of communication or delays in information exchange can exacerbate operational issues.

3 How the LightGBM Algorithm Works

This chapter explains the algorithm used in this study: LightGBM ("Light Gradient Boosting Machine"). However, before describing it in detail, it is necessary to explain prior algorithms, such as Decision Trees and Gradient Boosting in general, which will be discussed in the following sections.

3.1 Decision Trees

Decision Trees are supervised learning models used for both classification and regression tasks. They function by splitting data into subsets based on a decision criterion at each internal node, creating a tree-like structure with leaves representing the final decisions or predictions. Figure 3.1 shows an example of a Decision Tree for a classification task of deciding whether or not to leave the house to help illustrate the basic structure of a Decision Tree.

3.1.1 Tree Structure

Decision Trees are composed of nodes and branches:

- **Root Node**: The initial node representing the entire dataset.
- **Internal Nodes**: Each internal node represents a decision attribute that splits the data based on an impurity criterion, such as entropy or the Gini index.
- **Leaves**: The leaves are the final nodes that represent the ultimate decision or prediction after all splits.

The tree construction involves the following steps:

1. Select the attribute that best splits the data at each node based on impurity measures.
2. Split the data into subsets according to the selected attribute.

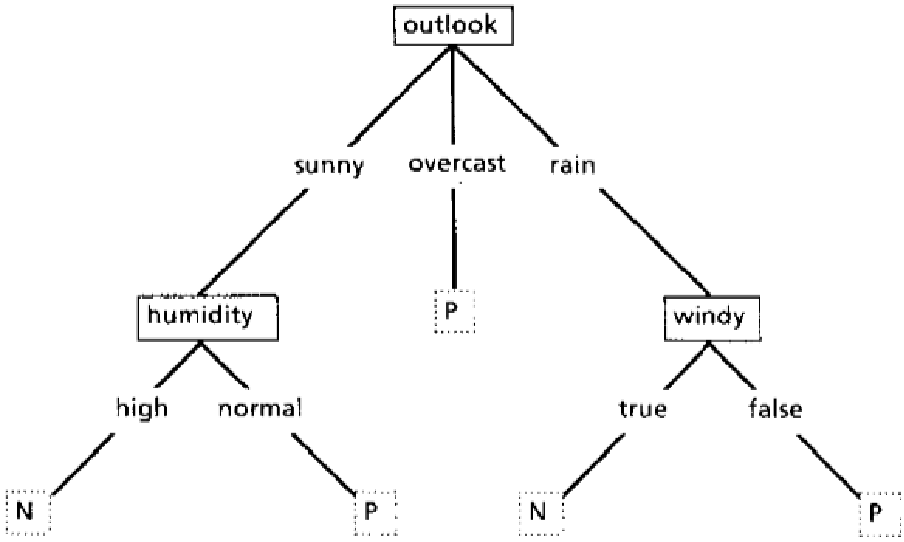

FIGURE 3.1 – Example of a Decision Tree for classifying whether an individual will leave the house. Source: (QUINLAN, 1986).

3. Repeat the process recursively for each subset until a stopping criterion is reached, such as maximum tree depth or the minimum number of samples in a node (BREIMAN *et al.*, 1984).

3.1.2 Impurity Calculation

To decide which attribute to use for splitting data at each node, impurity measures are applied:

- **Entropy**: Measures a dataset's degree of disorder or impurity. It is calculated using the formula:

$$\text{Entropy}(S) = -\sum_{i=1}^{c} p_i \log_2 p_i,$$

  where c is the number of classes in the set S and p_i is the proportion of instances belonging to class i in S (QUINLAN, 1986).

- **Gini Index**: Measures the probability that a sample will be incorrectly classified if it is randomly labeled according to the class distribution in the dataset. It is computed using the formula:

$$\text{Gini}(S) = 1 - \sum_{i=1}^{c} p_i^2,$$

  where c is the number of classes in the set S and p_i is the proportion of instances belonging to class i in S (BREIMAN *et al.*, 1984).

3.1.3 Usage Example

Let us consider a classification problem where we aim to predict which runway direction will be used at a given time. The process can be described as follows:

- Each node in the tree represents a condition that could affect the decision, such as "is the wind speed above 6 kt?" or "is it a tailwind?".
- The tree branches represent the answers to these questions ("yes" or "no").
- The order of questions is determined to maximize entropy reduction or information gain at each decision.
- By traversing the tree and answering these questions, we reach a leaf node representing the final prediction (e.g., runway direction 17 or 35).

3.2 Gradient Boosting

Gradient Boosting is a machine learning technique for regression and classification problems that produces a predictive model in the form of an ensemble of weak models, typically decision trees. It is an ensemble learning method that builds the model in stages, training new models to correct the errors of the previous ones (FRIEDMAN, 2001).

3.2.1 Initial Model

The Gradient Boosting process begins by creating a simple model, such as a small decision tree, which provides an initial prediction for all the training data (FRIEDMAN, 2001).

3.2.2 Residual Calculation

After the initial model makes its predictions, the residuals are calculated. The residuals are the errors made by the current model, defined as the difference between the model's predictions and the actual values of the training data, as described by Friedman (2001):

$$\text{Residual} = \text{Actual Value} - \text{Prediction}.$$

3.2.3 Fitting Sequential Models

Each new model is trained to predict the residuals (errors) of the previous model. The goal is to fit the latest model to minimize the ensemble's loss function. This is achieved using a gradient descent method, where the direction and magnitude of the updates are determined by the derivative of the loss function concerning the model's predictions (FRIEDMAN, 2001).

3.2.4 Combining the Models

The models are combined by weighting them according to their performance. The final combination is the weighted sum of all models, given by:

$$\text{Final Prediction} = \sum_{m=1}^{M} \gamma_m \cdot h_m(x),$$

where M is the number of models, γ_m is the weight of the m-th model, and $h_m(x)$ is the prediction of the m-th model (HASTIE *et al.*, 2009).

3.2.5 Usage Example

Again, consider a classification problem where we aim to predict the runway direction used at a given time based on weather conditions.

- **Step 1**: The initial model predicts the runway direction (e.g., the most frequently used direction).
- **Step 2**: The residuals are calculated, i.e., the difference between the actual runway direction used and the one predicted by the initial model.
- **Step 3**: A new model is trained to predict these residuals. This model considers wind speed and direction, visibility, and rain conditions.
- **Step 4**: The residual predictions are added to the initial model's predictions to improve prediction accuracy.
- **Step 5**: This process is repeated, adding sequential models that correct the errors of previous models until the ensemble achieves satisfactory accuracy (HASTIE *et al.*, 2009).

3.2.6 Advantages and Disadvantages

- **Advantages**:
  - High predictive accuracy.
  - Flexibility to use different loss functions.
  - Robust against overfitting, provided that appropriate parameters are chosen.
- **Disadvantages**:
  - Can be computationally expensive and slow to train, especially with large datasets.
  - Requires careful parameter tuning to avoid overfitting.

3.3 LightGBM

LightGBM is an implementation of gradient boosting designed to be efficient in speed and memory usage. It is widely used in data science competitions and industrial applications due to its high performance, as demonstrated by Machado *et al.* (2019), Zhang *et al.* (2019), and Wang *et al.* (2021).

3.3.1 Histogram-Based Learning

Instead of using all data to build trees, LightGBM employs a histogram-based technique that groups data into discrete bins. This significantly reduces training time and memory usage.

- Continuous values are converted into discrete bins, forming histograms.
- Tree construction is based on these histograms, making the process faster and more efficient.

3.3.2 Leaf-Wise Growth

LightGBM grows decision trees in a leaf-wise manner, expanding the most promising leaf nodes instead of growing the tree level-wise. This results in deeper and potentially more accurate trees.

- Leaves with the highest loss reduction are chosen for growth.
- This can lead to better performance, especially with large datasets.

3.3.3 Handling Large Datasets

LightGBM is designed to efficiently handle large volumes of data by using sampling and optimization techniques.

- Supports parallel learning, allowing the use of multiple CPU (Central Processing Unit) cores.
- Offers support for GPU (Graphics Processing Unit) learning, further accelerating the training process.

3.3.4 Hyperparameters

The choice of hyperparameters can significantly influence the performance of models based on LightGBM. Table 3.1 describes some of the main hyperparameters used in training the model.

TABLE 3.1 – Main hyperparameters of LightGBM.

Hyperparameter	Description
`num_leaves`	Defines the maximum number of leaves in each tree. A higher number can increase model accuracy but also raises the risk of overfitting.
`max_depth`	Limits the maximum depth of the trees. Setting an appropriate value for depth can help control overfitting.
`learning_rate`	Also known as the learning rate, it determines the contribution of each tree to the final model. Lower learning rates tend to produce more robust models but require more iterations.
`feature_fraction`	Specifies the feature fraction to consider when building each tree. This can help reduce overfitting and speed up training.
`bagging_fraction`	Determines the fraction of training data to be used to train each tree. Similar to `feature_fraction`, it can help prevent overfitting.
`lambda_l1`	Applies L1 regularization to the weights of the tree leaves, promoting sparsity in the model.
`lambda_l2`	Applies L2 regularization to the weights of the tree leaves, helping to avoid large coefficients and, consequently, overfitting.

3.3.5 Usage Example

Consider a classification problem where we aim to predict the runway direction in use at a given time based on weather conditions.

- **Step 1**: The data are initially converted into histograms, grouping continuous values into discrete intervals.
- **Step 2**: The initial model predicts the runway direction (e.g., the most frequently used direction).
- **Step 3**: The residuals are calculated, i.e., the difference between the actual runway direction used and the one predicted by the initial model.
- **Step 4**: A new model is trained to predict these residuals. This model considers variables such as wind speed and direction, visibility, and rain conditions.
- **Step 5**: The residual predictions are added to the initial model's predictions to improve prediction accuracy.
- **Step 6**: This process is repeated, adding sequential models that correct the errors of previous models until the ensemble achieves satisfactory accuracy (KE *et al.*, 2017).

3.3.6 Advantages and Disadvantages

Here we list the main positive and negative points of the LightGBM algorithm, based primarily on Ke *et al.* (2017).

- **Advantages**:
  - High efficiency in terms of time and memory.
  - Excellent performance on large datasets.
  - Support for parallel and GPU learning.
  - Better predictive performance due to leaf-wise growth.
- **Disadvantages**:
  - Can be more complex to implement and tune than other algorithms.
  - Leaf-wise growth may lead to overfitting if not well-regularized.

4 Methodology

In this section, we describe how the problem was approached in relation to the implementation carried out up to this point so that the method is reproducible for readers who wish to use it.

4.1 Data Collection

The data used in developing the runway configuration predictive model, from training to testing, were collected from different sources, which will be described below. It is worth noting that the period selected for training and testing was from 28/12/2023 to 30/09/2024, limited by the available WRF (Weather Research and Forecasting) dataset.

4.1.1 WRF

The WRF data were collected from the ODIN (Operational Data Integrator) database provided by the ICEA (Airspace Control Institute) Research Division. These data formed the basis of the table that combines all the data, which will be described in more detail in the following sections. This was done because it is the data with the highest frequency of forecasts among those used: its projections are made at 15-minute intervals.

4.1.2 METAR

The METAR (Meteorological Aerodrome Report) data were collected through the REDEMET (Aeronautics Command Meteorology Network) API (Application Programming Interface), provided by DECEA (2024a). These data represent the average measurements from the aerodrome's Surface Meteorological Station (EMS) over a one-hour interval.

4.1.3 TAF

The TAF (Terminal Aerodrome Forecast) data, like the METAR data, were collected through the REDEMET API. However, these are forecast data, which are generated every 6 hours.

4.1.4 Runway Configuration Change History

Finally, obtaining the history of runway direction usage was necessary, as the method used was supervised learning. The ODIN database of the ICEA Research Division provided the runway configuration history table to determine which runway direction (17 or 35) was in use during each time period within the selected interval.

4.2 Data Format

Each data source has a specific format, and understanding each one will be important to comprehend the following sections on how the data were processed.

4.2.1 WRF

The format of the WRF data provided in the ODIN database by ICEA is similar to that shown in Table 4.1, already filtered for the SBSP aerodrome (Congonhas).

TABLE 4.1 – Sample of WRF data collected and separated into columns.

Forecast Date	**Wind Speed**	**Wind Direction**	**Tailwind**	**Ref Date**
2023-12-28 06:00:00	5	100	0	2023-12-28 06:00:00
2023-12-28 06:15:00	5	100	0	2023-12-28 06:00:00
2023-12-28 06:30:00	4	80	0	2023-12-28 06:00:00
2023-12-28 06:45:00	4	90	0	2023-12-28 06:00:00
2023-12-28 07:00:00	4	100	0	2023-12-28 06:00:00

As previously described, these data are forecasts generated every 6 hours at 15-minute intervals. Wind Speed is measured in knots (kt), and Wind Direction is measured in degrees, counted from geographic north (0°) in a clockwise direction, so 90° represents East. It is worth noting that this structure, with "Forecast Date" as the table's primary key, will form the basis for data integration, which will be described later.

4.2.2 METAR

The METAR data are provided as a single string representing various aggregated information. An example of METAR data is as follows:

```
METAR SBSP 101200Z 14008KT 3000 R17R/2500U -DZ SCT020 BKN050 22/20 Q1015 TEMPO
```

In this case, the data indicate that, at Congonhas Airport (SBSP), at 12:00 UTC on the 10th (101200Z), the wind is blowing from 140° at 8 knots (14008KT), with visibility of 3000 meters (3000). Additionally, the runway visual range on runway 17R is 2500 meters, with an upward trend (R17R/2500U), indicated by "U" (from "upward"). There is also an indication of drizzle (-DZ). Clouds are scattered at 2000 feet (SCT020) and broken at 5000 feet (BKN050). Finally, the temperature is 22°C, the dew point is 20°C (22/20), sea-level pressure is 1015 hPa (Q1015), and temporary (TEMPO) changes in weather conditions are expected in the coming hours.

Some fields in the METAR data have additional classifications, such as clouds and expected changes. However, detailing each of these is beyond the scope of this work, and the example provided only illustrates how the data were fragmented to integrate the table that will serve as input to the model. For readers interested in more details, DECEA (2024b) explains each possible element of the METAR message. After the data are separated into relevant columns, they are arranged as shown in Table 4.2.

4.2.3 TAF

Finally, the TAF data, like the METAR, are provided as a character string in which each part represents a specific piece of information. However, this representation differs somewhat from that of the METAR. Consider this example:

```
TAF SBSP 121100Z 1212/1312 14010KT 5000 -RA BKN020
     TEMPO 1212/1218 3000 RA SCT010 BKN015
     BECMG 1220/1222 16008KT 9999 NSW SCT020
     BECMG 1300/1302 14010KT 4000 BR SCT005 BKN010.
```

TABLE 4.2 – Sample of METAR data collected and separated into columns.

Date	**Wind**	**Visibility**	**Clouds**	**Max Temp**	**Min Temp**	**Pressure**
2024-04-13 20:00:00	16012KT	1320	BKN046	24	19	Q1018
2024-04-13 21:00:00	16011KT	1321	SCT008	22	19	Q1018
2024-04-13 22:00:00	16011KT	1322	SCT009	22	19	Q1019
2024-04-13 23:00:00	15008KT	1323	SCT007	21	19	Q1019
2024-04-14 00:00:00	13007KT	1400	SCT009	21	18	Q1019

In this case, the data indicate that, at Congonhas Airport (SBSP), the weather forecast for the period from 12:00 UTC on the 12th to 12:00 UTC on the 13th (1212/1312) is wind blowing from 140° at 10 knots (14010KT), with visibility of 5000 meters (5000) and light rain (-RA), with broken clouds at 2000 feet (BKN020). Between 12:00 and 18:00 UTC on the 12th, temporary conditions with visibility of 3000 meters and rain (3000 RA) are forecasted, with scattered clouds at 1000 feet (SCT010) and broken clouds at 1500 feet (BKN015). Between 20:00 and 22:00 UTC on the 12th, a gradual change is expected, with wind from 160° at 8 knots (16008KT), visibility of 10 km or more (9999), and no significant weather (NSW), with scattered clouds at 2000 feet (SCT020). Between 00:00 and 02:00 UTC on the 13th, another gradual change is expected, with wind from 140° at 10 knots (14010KT), visibility of 4000 meters (4000), and mist (BR), with scattered clouds at 500 feet (SCT005) and broken clouds at 1000 feet (BKN010).

As explained after the METAR example, detailed explanations of each possible element in the TAF message are not provided here, but interested readers can refer to DECEA (2020b). After this TAF data is separated into columns, it appears as exemplified in Table 4.3.

4.3 Data Integration

After collecting each of the three datasets, we organized them into three separate tables — one for each dataset — and then integrated them into a single table by following these steps:

TABLE 4.3 – Sample of TAF data collected and separated into columns.

Date	Wind	Visibility	Clouds	Changes	Weather
2024-04-13 00:00:00	14010KT	1221	SCT006	[BECMG]	BR
2024-04-13 06:00:00	13005KT	1304	FEW015	[BECMG]	BR
2024-04-13 12:00:00	10008KT	1309	FEW018	[BECMG]	None
2024-04-13 18:00:00	07008KT	1316	SCT035	[BECMG]	RA
2024-04-14 00:00:00	14008KT	1322	BKN009	[BECMG]	None

1. We used the WRF table as the foundation for the integrated table, as it contains more data instances for the same time interval due to its 15-minute forecast intervals.

2. For each row in the WRF table, we retrieved the corresponding METAR data by matching the METAR reading to the specific date and time. Since the new table includes measurements every 15 minutes, and METAR data is provided hourly, we repeated each METAR entry across the four 15-minute intervals within the same hour.

3. We applied a similar approach for the TAF data, retrieving the relevant TAF forecast for the specified time. In this case, forecasts are only available every 6 hours, resulting in repeated values, similar to the METAR data.

4.4 Exploratory Data Analysis

We conducted an exploratory data analysis to investigate potential seasonality in the data, examining runway usage in different time units. Specifically, we considered the time of day, day of the week, and month as the main units to evaluate possible seasonal patterns.

Figures 4.1, 4.2, and 4.3 provide a detailed visualization of runway usage patterns at different timescales. Figure 4.1 shows the distribution of runway usage across various hours of the day, highlighting any potential hourly trends. Figure 4.2 illustrates how runway preference varies throughout the days of the week, potentially revealing specific weekday patterns. Finally, Figure 4.3 presents a month-by-month analysis of runway usage, allowing for observing any seasonal trends or long-term variations in runway selec-

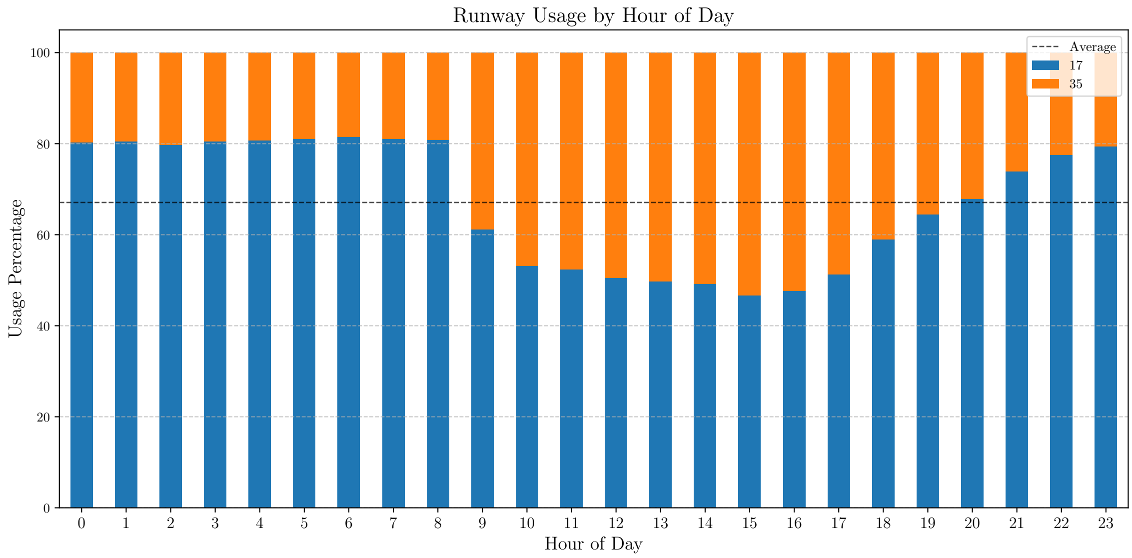

FIGURE 4.1 – Percentage usage of each runway (17 and 35) as a function of the hour of the day.

tion. Together, these figures offer a comprehensive view of how runway usage fluctuates across different temporal dimensions.

4.5 Data Preprocessing

After integrating the three data types into a single table, we performed data preprocessing to prepare the table for input into the model. The steps taken are as follows:

1. **Time Difference**:

   To handle date information more effectively, we focused on extracting the difference between the two date columns ("Forecast Date" and "Reference Date") rather than directly using the day and time. This approach allows the model to account for the fact that the farther a forecast is from the reference time, the less accurate it tends to be. By including this difference, we enabled the model to weigh forecasts less as the time gap increases.

2. **Cleaning Numerical Columns**:

   At this stage, we cleaned the numerical data in columns that should contain only numeric values, such as wind and pressure in METAR data and wind in TAF data, by removing any associated units that were still attached to the numbers. At this point, we also removed the R/L information from the runway used, considering that the practical difference between them is minimal, and this work focuses on predicting the runway direction (17 or 35), given that the impact of changing the runway direction is much greater than the changing between the right and left runways.

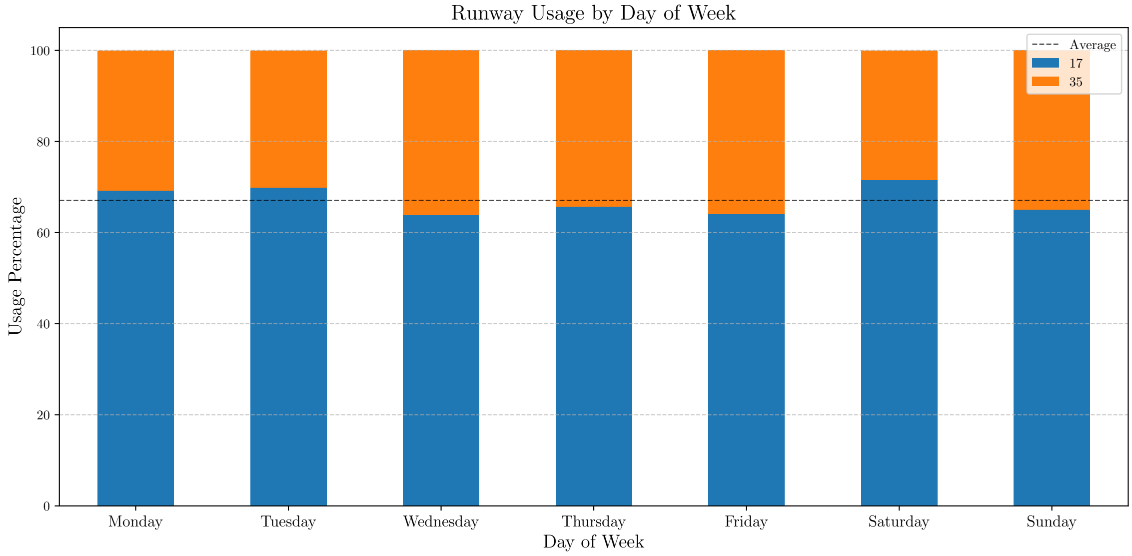

FIGURE 4.2 – Percentage usage of each runway (17 and 35) as a function of the day of the week.

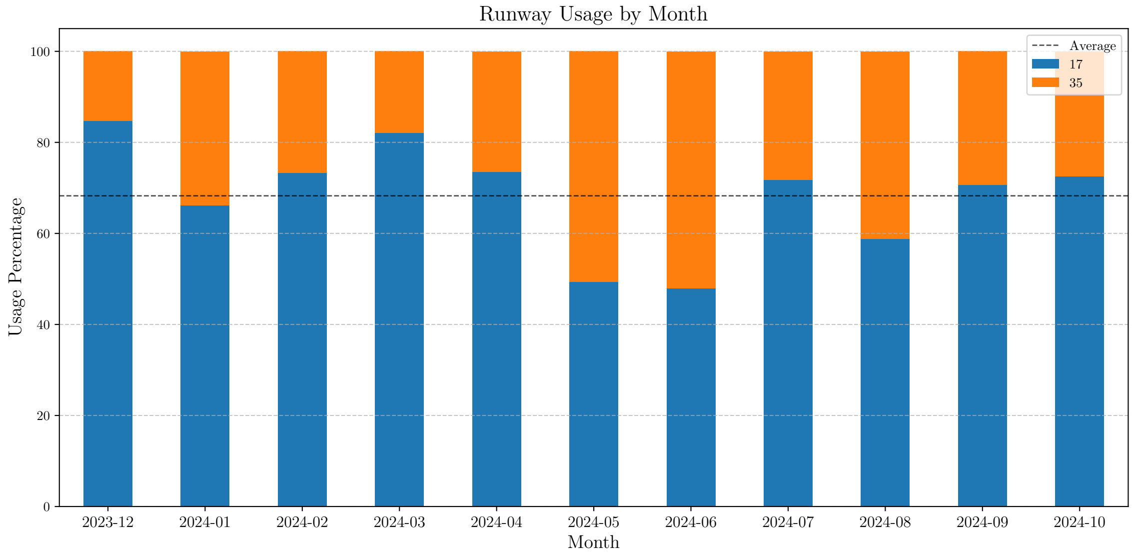

FIGURE 4.3 – Percentage usage of each runway (17 and 35) as a function of the month.

3. **Handling Missing Values**:

   We replaced all missing values with *"NoInfo"* ("No Information") for categorical variables. This approach not only highlights the absence of information, which the model may interpret as useful, but also avoids the need for artificial or imprecise imputations and ensures that we can still use the column despite occasional missing values. To make this missing value imputation analysis more rigorous, we tested the use of KNN (K-Nearest Neighbors) for this purpose, using Python's `KNNImputer` from the `sklearn` library. However, this approach did not yield any improvements in model accuracy and F1-score in our tests, leading us to retain the initial *"NoInfo"* imputation approach, as it produced the same practical result as the KNN method but in a simpler manner.

4. **Manually Adding Categories**:

   To prepare categorical data for encoding, we manually added all possible categories. This ensures that later automatic encoding remains consistent and independent of the dataset used. With this approach, we maintain the same columns across all datasets, even if some categories do not appear in certain specific instances.

5. **Encoding Categorical Variables**:

   Initially, we encoded categorical variables using One-Hot Encoding, as it is a common approach to help machine learning models handle such data. Given that LightGBM can effectively process categorical variables directly, we tested the model without One-Hot Encoding, which led to an increase in both average accuracy and F1-score. Consequently, the final version of the models simply flagged certain variables as categorical to LightGBM, without any prior encoding.

6. **Normalizing Numerical Columns**:

   We initially applied normalization using the `StandardScaler` from Python's `sklearn` library, as this is typically a beneficial step when working with tabular data that include variables of differing scales. During development, however, we tested the models without normalization, given that LightGBM can handle numerical variables without requiring normalization to achieve strong performance. These tests showed improved average accuracy and F1-score without normalization, so this approach was adopted in the final version of the models.

7. **Including Future Runway Information**:

   Since the goal was to predict the runway in use up to 3 hours ahead in 15-minute intervals, we adapted the dataset to include annotations of the runway actually used in each 15-minute interval in the future. For instance, the row corresponding to 10:00:00 now contains columns indicating the actual runway used not only at

10:00:00 but also at 10:15:00, 10:30:00, 10:45:00, and so on, up to 13:00:00. This setup allows the model to learn the relationship between each input and the runway in use at various future time intervals.

8. **Temporal Information**:

   To help the models capture any potential daily, weekly, or monthly seasonal patterns, as shown in Figures 4.1, 4.2, and 4.3, we included certain temporal features related to the instant for which we aimed to predict the runway in use. Specifically, we used the hour of the day, day of the week, and month as input features.

4.6 Model Selection

After preparing the data for model input, we selected the classification technique. For this purpose, we utilized the `LazyPredict` (PANDALA, 2024) library, which compares various classification models and their respective performances. Upon running it on the processed dataset, the model that achieved the highest accuracy was the `LGBMClassifier`, or LightGBM. Additionally, some teams that performed well in data science challenges organized by ICEA used LightGBM, such as the winning team from 2023 (TELES, 2024). Therefore, these factors, combined with the other advantages described in previous sections, led us to choose this model to predict the runway configuration in this study.

4.7 Dataset Splitting

To perform the hyperparameter optimization and model training steps, which will be explained in Sections 4.10 and 4.11, respectively, we split the dataset (containing 25,380 instances) as follows:

- **Training Set (64%)**: Used to adjust the model parameters. This set contains most data (16,243 examples) to allow the model to learn existing patterns.
- **Validation Set (16%)**: Used during cross-validation, which will be explained in Section 4.10. For every four parts used for training, one was reserved to validate the model, totaling 4,061 examples.
- **Test Set (20%)**: Used for the final evaluation of the model. This set contains 5,076 examples and is not used during training or hyperparameter tuning, ensuring a fair and unbiased assessment of model performance.

4.8 Independent Temporal Classifications

Since the model needed to predict the runway configuration for 12 different future time intervals (from the next 15 minutes up to the next 180 minutes, or 3 hours) for each instance, it essentially had to perform multiple classifications simultaneously. To isolate the problems, considering that each one would have its specific characteristics, we trained 12 distinct models, each based on the LightGBM classifier and using the same input data (WRF, METAR, and TAF), but each one aimed at predicting one of the 12 columns, meaning the runway configuration at a specific future time interval. This approach allowed each model to focus on understanding the relationship between the input variables and a single output variable, namely, the runway configuration at a particular future moment, enabling better specialization for each model in its task.

4.9 Dataset Imbalance

We employed class weighting to adjust the model's sensitivity to each class to mitigate the class imbalance between runways 17 and 35, where runway 17 appears approximately 2.6 times more frequently than runway 35. This technique ensures that the model does not overly favor the majority class (runway 17) simply because of its higher occurrence rate, thus maintaining a balanced performance across both classes.

We implemented this approach using the `compute_sample_weight` function from the `sklearn` library. This function assigns specific weights to each sample based on the distribution of classes in the training data, effectively increasing the importance of the minority class (runway 35) during model training. These calculated weights were then incorporated directly into the model training process by passing them as the `sample_weight` parameter in the `fit` method of the `random_search`. Consequently, while the randomized search optimized the hyperparameters, the class weights ensured that each model treated both classes more equitably throughout the training process. We repeated this process during the final retraining of the models on the full training dataset to reinforce balanced performance.

This approach allowed the model to maintain balanced performance, even with significant class imbalance. The integration of sample weighting into both the randomized search and retraining processes aligns the model's learning strategy with the dataset's imbalanced nature, thus enhancing its robustness and accuracy when predicting the runway configuration at Congonhas Airport.

4.10 Hyperparameter Selection

To optimize the performance of the LightGBM model, we performed a hyperparameter search using the `RandomizedSearchCV` technique for each of the 12 classifier models due to its effectiveness in optimization (BERGSTRA *et al.*, 2012). This process involved defining a distribution for the hyperparameters, including the number of leaves (‘`num_leaves`’), the fraction of features used (‘`feature_fraction`’), the fraction of samples for aggregation (‘`bagging_fraction`’), the maximum depth of trees (‘`max_depth`’), and the regularization parameters (‘`lambda_l1`’ and ‘`lambda_l2`’). We conducted the search using 5-fold cross-validation to ensure the robustness of the results. After training and evaluating the accuracy of all hyperparameter combinations, we selected the set with the highest accuracy on the validation set. The distributions used in the search for each model can be seen in Table 4.4, and the values chosen for each model are shown in Table 4.5.

TABLE 4.4 – Table of hyperparameters with distributions used in the `RandomizedSearchCV` method.

Hiperparâmetro	**Distribuição**
`num_leaves`	`randint(20, 150)`
`feature_fraction`	`uniform(0.7, 0.3)`
`bagging_fraction`	`uniform(0.7, 0.3)`
`max_depth`	`randint(3, 20)`
`lambda_l1`	`uniform(0.0, 1.0)`
`lambda_l2`	`uniform(0.0, 1.0)`

4.11 Model Training

With the optimized hyperparameters, we trained the LightGBM model again, this time using the entire training set rather than just 80% of it (i.e., 64% of the total data), as was done in the 5-fold cross-validation. This approach allows the optimized model to learn even more by training on the complete training set.

It is worth noting that the metric used during training was binary logarithmic loss (`binary_logloss`), which iteratively adjusts the model's parameters to improve its performance. We chose this metric over others (such as binary error) due to its ability to weigh not only the decision made by the model but also the confidence involved, penalizing incorrect decisions made with high confidence, for example.

TABLE 4.5 – Hyperparameters of the 12 models selected by the `RandomizedSearchCV` method.

Modelo	`num leaves`	`feature fraction`	`bagging fraction`	`max depth`	`lambda l1`	`lambda l2`
Model 1 (15 min)	137	0.72	0.77	13	0.13	0.13
Model 2 (30 min)	147	0.73	0.79	13	0.46	0.22
Model 3 (45 min)	137	0.72	0.77	13	0.13	0.13
Model 4 (60 min)	137	0.72	0.77	13	0.13	0.13
Model 5 (75 min)	137	0.72	0.77	13	0.13	0.13
Model 6 (90 min)	137	0.72	0.77	13	0.13	0.13
Model 7 (105 min)	137	0.72	0.77	13	0.13	0.13
Model 8 (120 min)	137	0.72	0.77	13	0.13	0.13
Model 9 (135 min)	137	0.72	0.77	13	0.13	0.13
Model 10 (150 min)	137	0.72	0.77	13	0.13	0.13
Model 11 (165 min)	137	0.72	0.77	13	0.13	0.13
Model 12 (180 min)	148	0.83	0.78	11	0.08	0.03

4.12 Decision Confidence

To enhance the credibility of the model's decisions, we calculated the confidence it has in each prediction in addition to the classification. This was done using the predicted probabilities for each class during the prediction phase. Specifically, we recorded the highest probability for each instance among the possible classes, representing the model's confidence in its decision. This confidence measure indicates how certain the model is when classifying a given instance, where values close to 1 suggest high certainty in the prediction, while lower values indicate greater uncertainty. This approach enables a more robust and transparent analysis of the predictions, providing an additional layer of reliability to the model's results.

5 Results and Discussions

In this section, we present the results obtained and discuss the insights derived from them.

5.1 Model Performance

During the search for the best set of hyperparameters, conducted using cross-validation as explained in Section 4.10, the average accuracy achieved by each model on its respective test set (20% of the data) was 93.7%, with a standard deviation of 0.4%. This result is illustrated in Figure 5.1, which shows the accuracies for each fold in the cross-validation process for the best hyperparameter combination found for each model. In other words, this figure displays the results of all 5 folds and the respective average for the best model found for each of the 12 future time intervals to be predicted.

After completing the cross-validation and finding the best set of hyperparameters for each model, we conducted a new training process that included all available training data rather than only 80% of it. The average accuracy obtained by the models following this process was 98.6%, with a standard deviation of 0.2%, highlighting the models' ability to predict runway configuration at Congonhas Airport accurately. During this retraining process, we used `early_stopping` from the LightGBM library, with a value of 100 for the `stopping_rounds` parameter, meaning that the process stops when there is no reduction in binary log loss for 100 consecutive iterations. The training curve for each of the 12 models is shown in Figure 5.2, where it is possible to observe early stopping, as each curve ends at different points.

To further assess the models' generalization capability and test them with minimal bias, we evaluated them outside the training and test range of 28/12/2023 to 30/09/2024. We selected the period from 01/10/2024 to 25/10/2024 for this test, during which the models achieved an average accuracy of 92.0% and an F1-score of 91.8%. These results are displayed in Figure 5.3, which shows the accuracy and F1-score values for each future period.

Additionally, we computed the consolidated confusion matrix shown in Figure 5.4 for

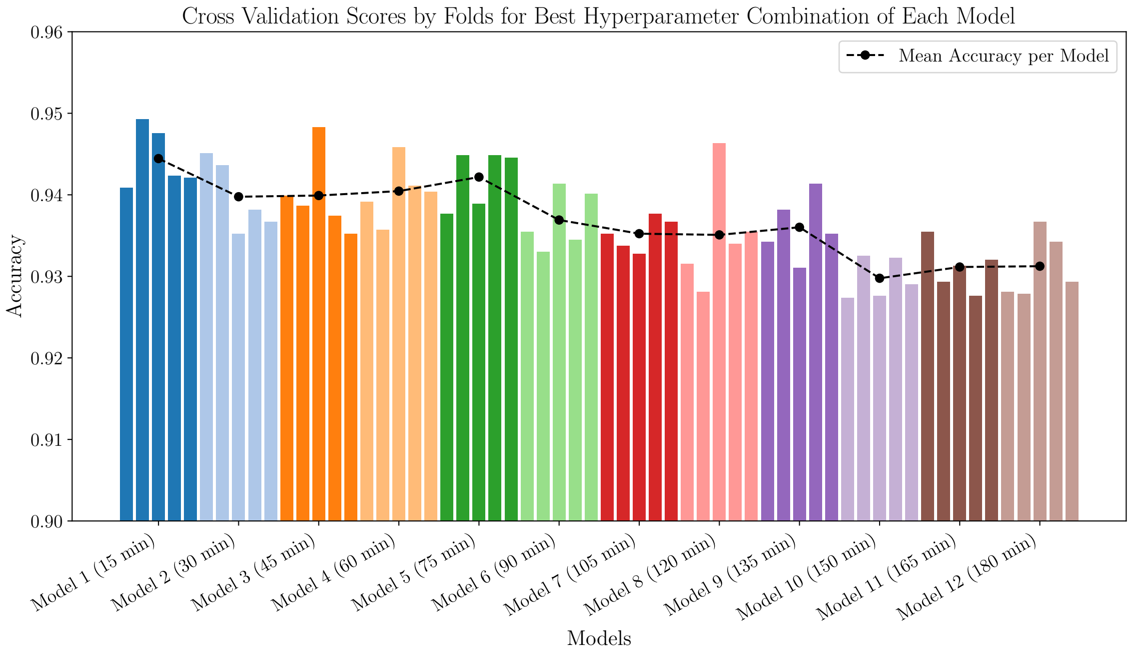

FIGURE 5.1 – Accuracy across the different folds in the cross-validation of `RandomizedSearchCV`, along with the average of these values for each of the 12 models. It is important to note that the y-axis does not start at 0 to highlight the differences in accuracy values between folds for each future time interval.

the test on the new dataset for the model, outside the set used for training, validation, and testing. This matrix aggregates all classifications made by the 12 models on this data to facilitate visualization.

Finally, as described in Section 4.12, the model's confidence in each classification prediction was calculated in addition to the predictions themselves to increase operators' confidence in the predictive model. This was done using the model's calculated probability for each class, as detailed in the same section. To provide a consolidated view of the data, the average confidence, along with the median and interquartile range, was calculated for each of the 12 models across all classifications made on the new dataset, as shown in Figure 5.5.

5.2 Feature Importance

Feature importance was calculated using the `feature_importance` function from the LightGBM library, `lgb`. This calculation is based on a sensitivity analysis of the trained model's loss function concerning each input variable. This analysis helps to understand the model's decision-making process better. To streamline and condense the information, Figure 5.6 shows the average importance of each variable across the 12 classifier models. Additionally, the black H-shaped line shows the minimum and maximum importance values assigned by any of the 12 models for these variables, providing an understanding

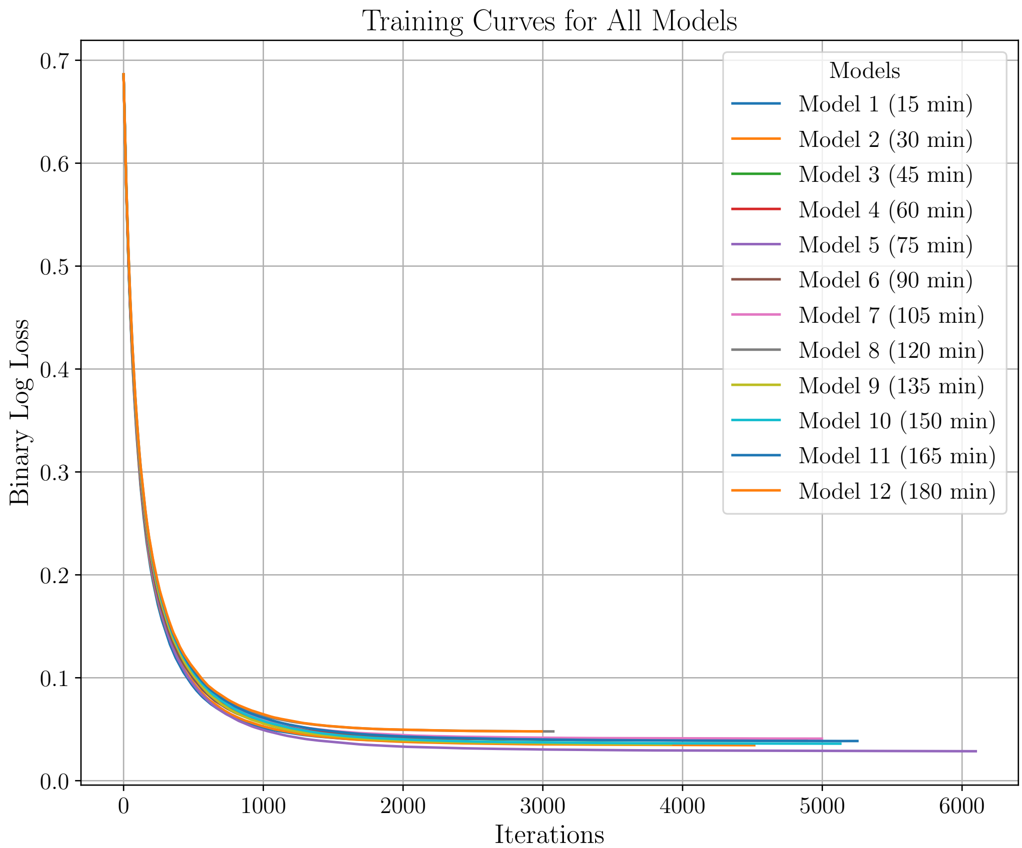

FIGURE 5.2 – Training curve for the best models resulting from `RandomizedSearchCV`, showing the evolution of binary log-loss over iterations.

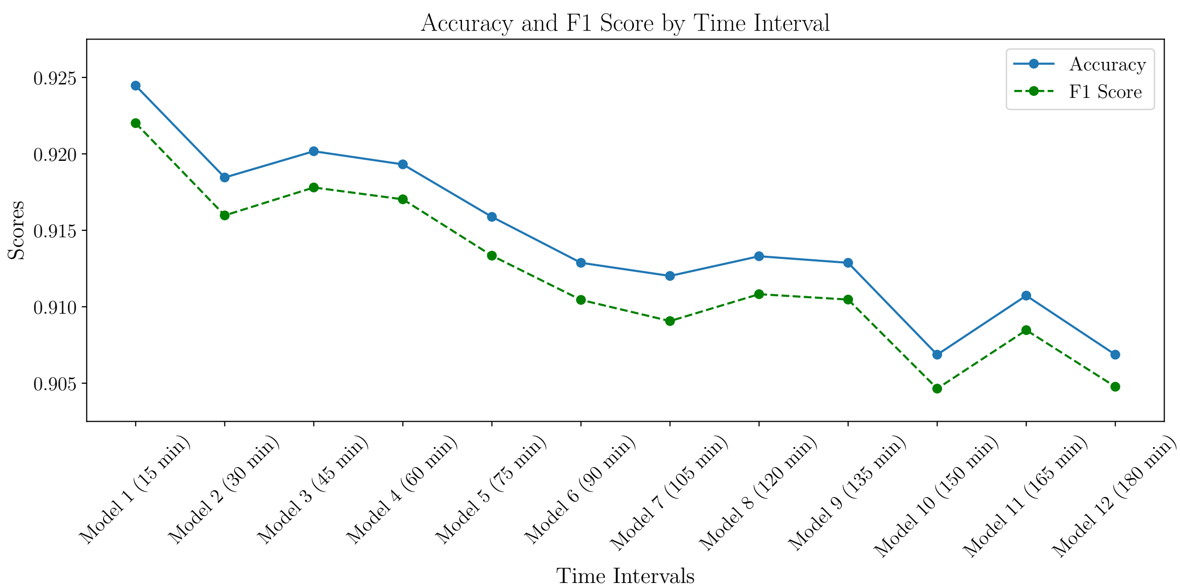

FIGURE 5.3 – Accuracy and F1-score for each future instance predicted by each model on the new dataset, outside the training and test data range. It is important to note that the y-axis does not start at 0 to highlight the differences in accuracy and F1-score values for each future time interval.

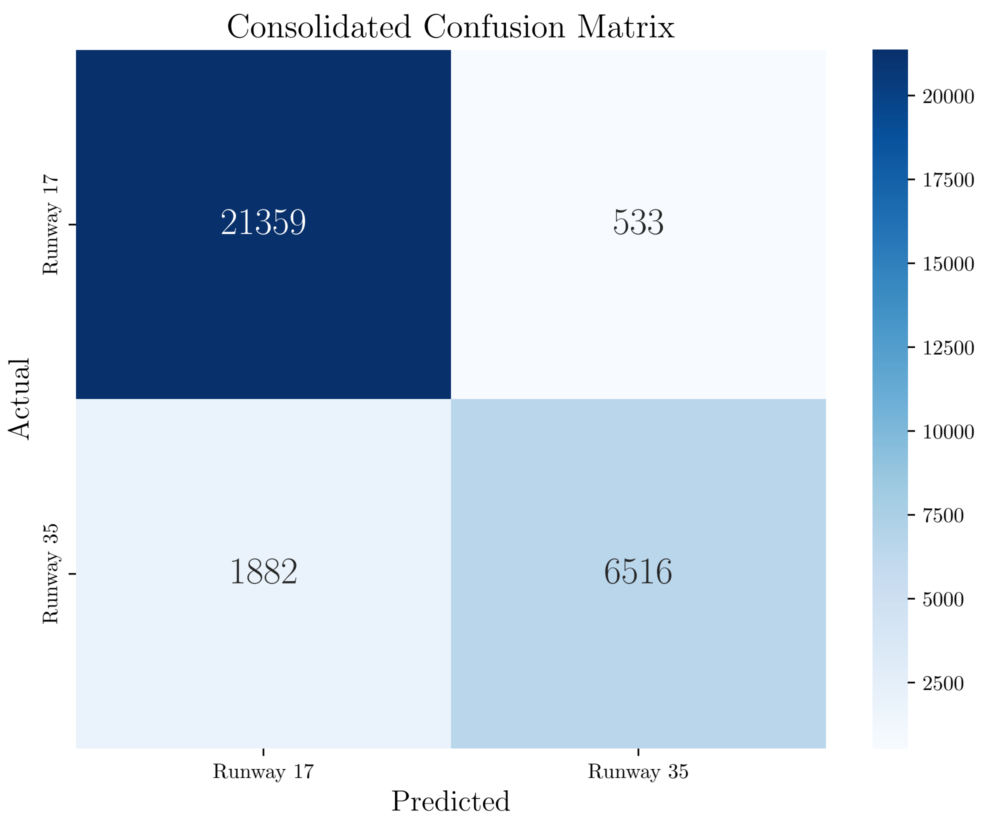

FIGURE 5.4 – Consolidated confusion matrix for the 12 models on the new dataset.

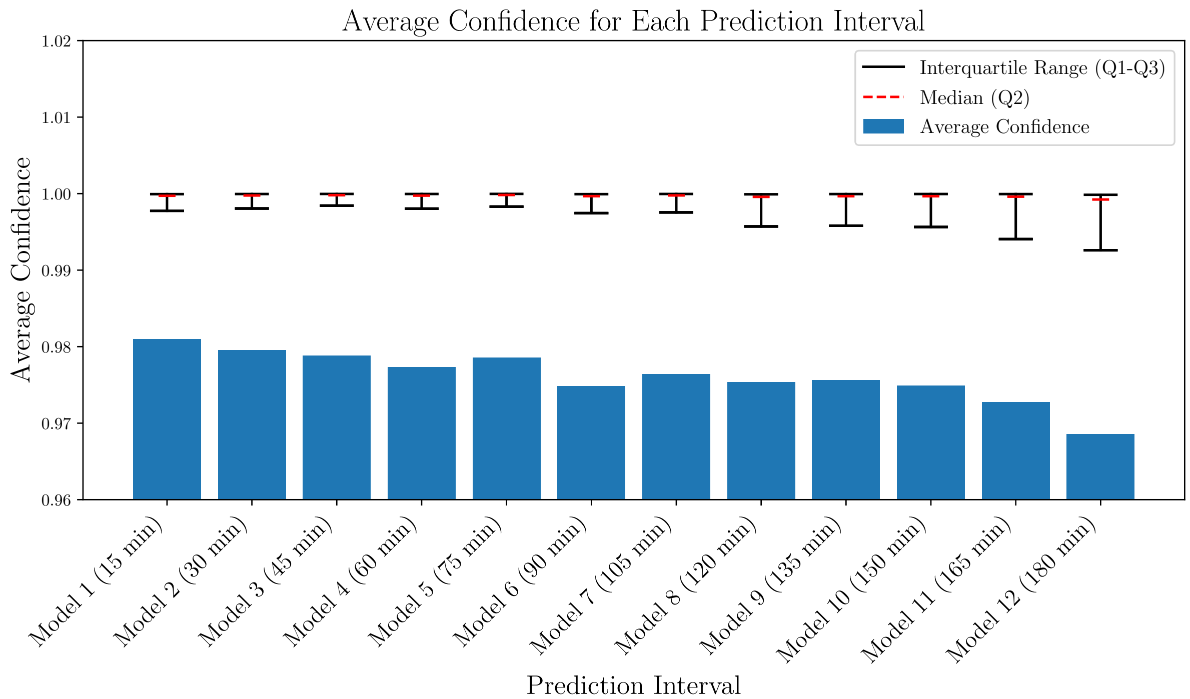

FIGURE 5.5 – Average confidence levels of model predictions across different forecast intervals on the new dataset. It is important to note that the y-axis scale starts at 0.96 to allow for the visualization of quartiles.

of the variability of these importances.

5.3 Evaluating Weather Prediction as Feature

One possibility evaluated during the development of the predictive model was to disregard forecast-related variables, such as TAF and WRF, to assess whether the model could independently understand the historical data patterns and infer these forecasts using only measurement data, specifically METAR. To enable a comparison with the final chosen model, whose results were presented in Section 5.1, we show analogous results for this model that only considered METAR data.

Figure 5.7 displays the average accuracy and F1-Score of the 12 models trained similarly to the previously described final model but using only METAR data. Figure 5.8 shows the consolidated confusion matrix of the classifications made by the 12 models trained exclusively with METAR.

5.4 Discussion of Results

Analyzing Figure 5.1, we observe that the random search process with cross-validation worked as expected, given the accuracy results for each of the five folds across all models. This is evident because, in addition to indicating significant learning from the data by the models, with an average accuracy of 93.7%, the variability between the folds is low, with a standard deviation around 0.4%. Thus, the primary objective of the random search with cross-validation was achieved, as the best hyperparameters were chosen, resulting in low variability across the folds, indicating no overfitting to any specific fold.

Moreover, we notice a gradual decrease in the average accuracy across the 5 folds from model 1 to model 12. This is in line with expectations, as the greater the time interval between the data's reference date and the date we aim to predict the runway configuration, the more challenging and less precise the prediction becomes. Nevertheless, it is worth mentioning that even for the most extended interval of 180 minutes, the model still achieved a respectable level of accuracy, with an average accuracy of approximately 92.3% across the folds.

Examining the training curve in Figure 5.2, we see that the learning process during retraining proceeded as expected for all models, with the binary log-loss decreasing at each iteration and stabilizing. Additionally, the results obtained on the test set after retraining demonstrate the predictive capability of these models, achieving an accuracy of 98.6%. Furthermore, the standard deviation of 0.2% across the results of each model on this test data indicates that the models performed consistently well, with minimal

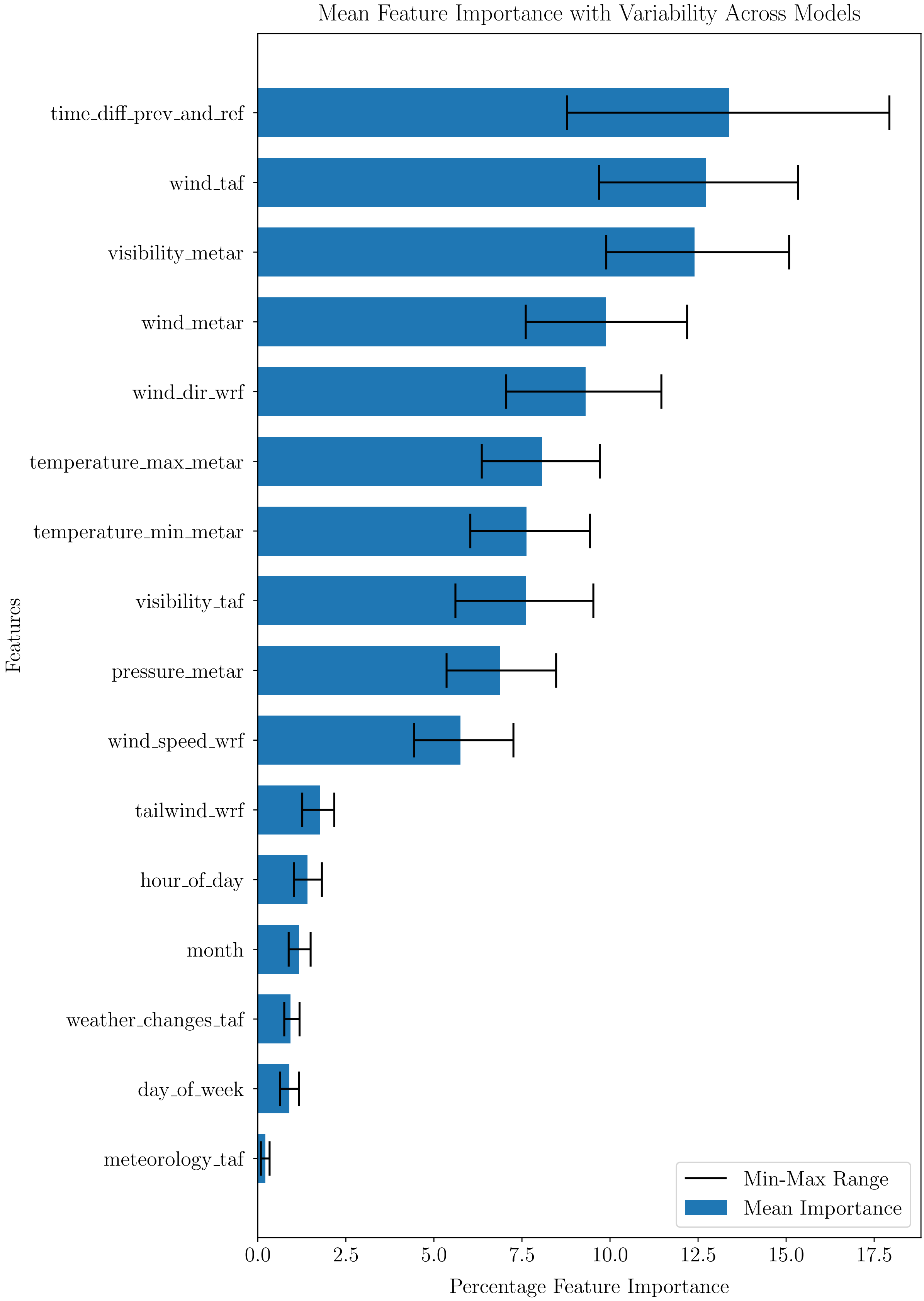

FIGURE 5.6 – Average importance of each variable (feature) for the models and their variability, filtered by features with percentage importance greater than 0.1%.

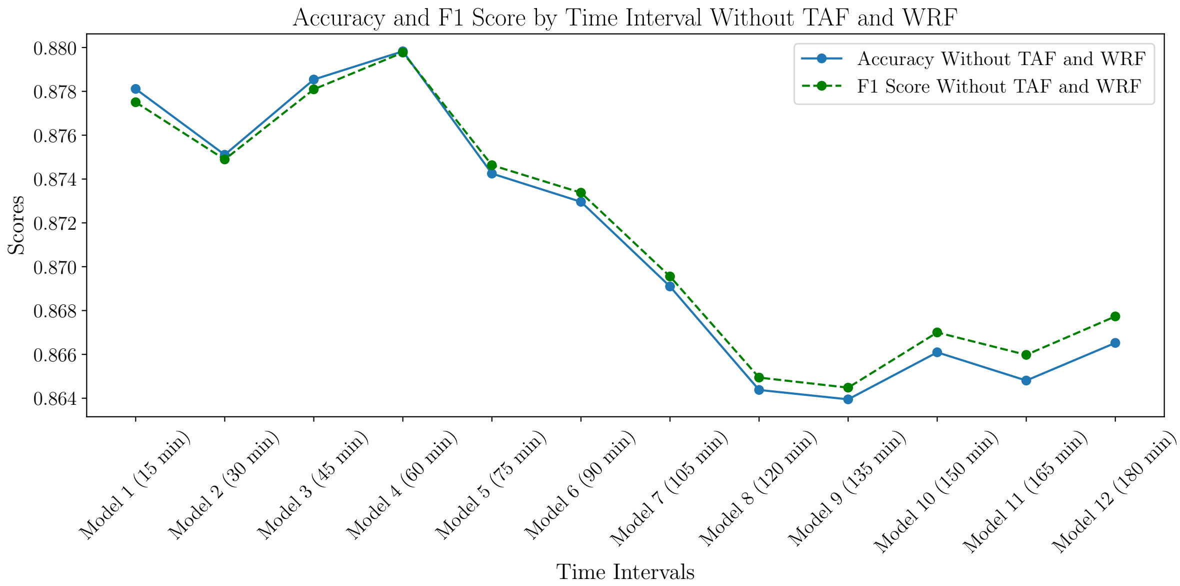

FIGURE 5.7 – Accuracy and F1-score for each future instance predicted by each model on the new dataset, outside the training and test data range, when trained only with METAR data. It is important to note that the y-axis does not start at 0 to highlight the accuracy and F1-score value differences for each future time interval.

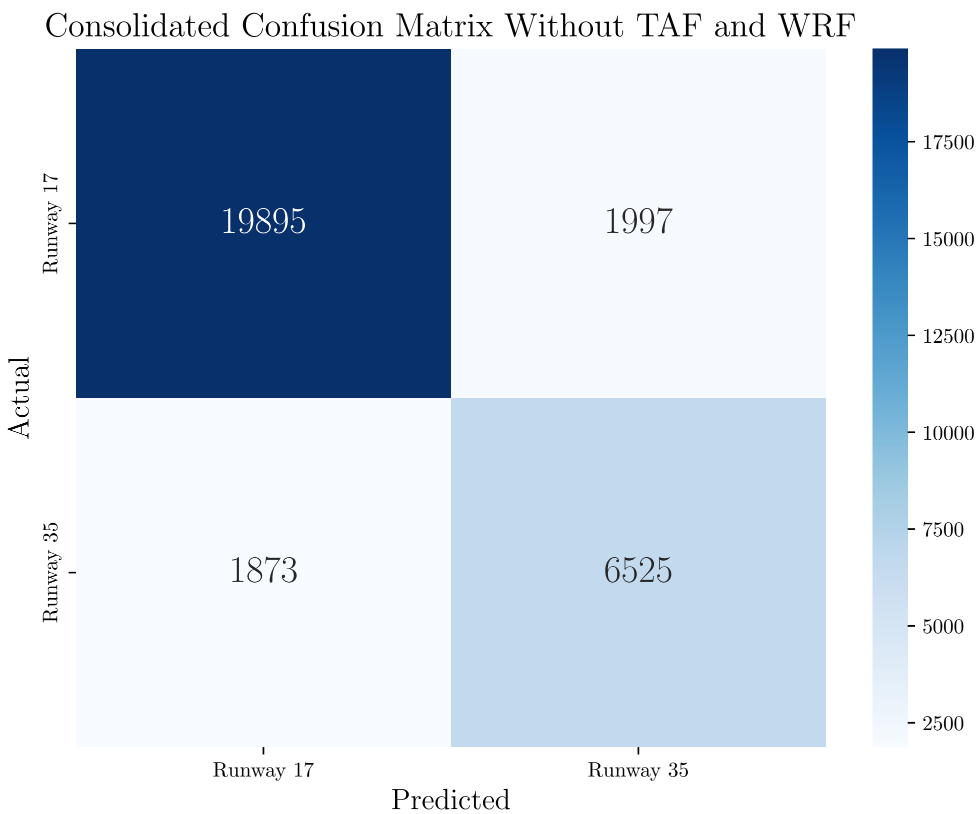

FIGURE 5.8 – Consolidated confusion matrix for the 12 models on the new dataset when trained only with METAR data.

variability among them.

When analyzing the models' performance on the new dataset, shown in Figure 5.3, with an average accuracy of 92.0%, we see excellent generalization capability, as the models had never been exposed to this data from October 2024. Complementarily, the average F1-score of 91.8% suggests that classifications (between runway 17 or 35) are balanced and accurate, not favoring one class over the other. Finally, as seen in Figure 5.1, it is again reasonable that accuracy (and F1-score) decreases as the interval between the reference and prediction instances increases.

This performance drop on the new dataset compared to the test set can be explained by the data exhibiting different patterns across months, as shown in Figure 4.3. Unfortunately, due to limitations in available WRF data, it was impossible to use even more training and test data while reserving a set for entirely new testing. Consequently, we chose to use October 2024 for this purpose, with the remaining data allocated for training, validation, and testing, as previously explained in section 4.7. One reason for this choice was that, as shown in Figure 4.3, October's usage percentages for each runway closely align with the average, making it a good reference month without strange behavior in this aspect for completely new testing.

However, despite the runway usage percentages being close to the average, various other variables may exhibit specific behaviors in October, especially meteorological variables. Nevertheless, due to the limitation of available WRF data, the models did not encounter this behavior during training and may not be making predictions as accurately as possible. Despite this limitation, it is worth noting that since almost all months of the year are represented in the available data, the models demonstrated excellent generalization ability, as evidenced by their high accuracies.

Another interesting analysis in this classification context is provided by the confusion matrix shown in Figure 5.4. This matrix helps us understand if the models are classifying correctly and the main errors when they make mistakes. In this sense, one issue with imbalanced data, as is the case here, is that runway 17 is used approximately 72% of the time, while runway 35 is used only 28% of the time. However, based on the results shown in the matrix, it is evident that the model is not biased toward one class over the other, as its precision is approximately 91.9% for class 17 and 92.4% for class 35. Therefore, we conclude that assigning different weights to each class, as discussed in section 4.9, ensured stability in the classifications.

Analyzing Figure 5.5, we can observe that the average confidence of the 12 models in their decisions on the new data is considerably high, remaining above 0.97 for all models. Additionally, by examining other statistical measures shown in the graph, such as the interquartile range and the median, we see that at least 75% of decisions are made with more than 99% confidence. This indicates an even higher level of confidence than the average might suggest. This is because some decisions are occasionally made with low confidence — when the model is uncertain between the two classes, and the confidence hovers around 50%. These outliers have a much greater impact on the average, significantly lowering its value. This effect does not impact measures like the median and quartiles as much, as they are less sensitive to outliers, which, in cases like this, helps us understand where most of the data is concentrated. This analysis of the confidence levels associated with classifications indicates that the models are making their decisions very reliably, further reinforcing the robustness of the model.

Additionally, the analysis of feature importance, shown in Figure 5.6, suggests that meteorological data significantly impacts the predictions across all time intervals. In particular, wind forecasts (TAF wind) proved to be highly important, which was expected, as it aligns with the Brazilian Air Force Command Instruction (ICA) 100-37/2020, where wind is one of the primary factors to consider when selecting the runway in use (DECEA, 2020a). We also observe the high importance of visibility measurements (METAR visibility), which is logical given that visibility is a crucial meteorological condition in the airport context. The most important feature is the temporal difference between the reference date and the target date for WRF forecast data. This result aligns with expectations, as it indicates the reliability of these forecasts, and the figure shows that the models appropriately assigned significant importance to this feature.

It is also noteworthy that the variability in importance each model assigns to a given variable, illustrated by the black H-shaped bar, indicates that, across different time intervals, the variables have varying degrees of relevance, reinforcing that each model should learn independently. Nevertheless, the most significant variables remain approximately the same, underscoring that, despite the unique aspects of each classification task, the problems also share some similarities.

Finally, analyzing the importance of weather forecasts through the results presented in section 5.3, we can observe that excluding TAF and WRF data generally worsened the models' performance. In Figure 5.7, it is evident that both the accuracy and F1-Scores of the models decreased compared to the model trained with WRF and TAF, whose results were shown in Figure 5.3. For models trained only with METAR data, the average accuracy was 87.2%, while the F1-Score was 87.3%. This alone shows that the predictions in this case are generally worse on average, but it does not reveal a significant imbalance in the classifications. However, analyzing the consolidated confusion matrix shown in Figure 5.8, we can calculate the precision for each class, resulting in 91.4% for class 17 and 76.6% for class 35, indicating that the model is making imbalanced errors in this case. Thus, we can conclude that the model's performance was significantly degraded when weather forecasts were removed from the feature set. Notice also that this does not rule out using only measurement data to achieve the proposed goal of having the model infer the forecasts. Instead, it suggests that, in this context and with the available data, it makes more sense to continue using forecasts as inputs for the model, as this improves its accuracy.

6 Conclusions and Future Works

In this chapter, we present conclusions about the work carried out and possible implementations to be explored in future research.

6.1 Conclusions

Based on the results presented and discussed in the previous chapter, we conclude that this work demonstrated the effectiveness of LightGBM models in predicting runway usage at Congonhas Airport up to 3 hours in advance, confirming that machine learning approaches can be successfully applied in complex operational contexts. After hyperparameter tuning, the model achieved an average accuracy of 98.6% on test data, highlighting its capacity for precise runway configuration predictions. In tests with new, unseen data, which simulated a completely fresh scenario for the model, the average accuracy of 92.0% and F1-Score of 91.8% demonstrate its robustness and generalization ability.

The confusion matrix analysis indicated that the model maintained balanced predictions between the two runways, even in an imbalanced data scenario, ensuring stable and unbiased performance thanks to using different class weights. The feature importance analysis highlighted the significant impact of meteorological conditions, especially wind forecasts, reflecting real-world practices in runway selection within the aeronautical context.

In summary, the developed model provides a solid and adaptable foundation for enhancing runway prediction in the airport environment. It serves as a reference for future applications that could benefit from new data and more specific adjustments. The approach demonstrated high accuracy and adaptability, even amid operational challenges and data limitations, establishing an effective and promising solution for similar scenarios.

6.2 Future Work

Although significant progress was made in this work, improvements to further enhance the predictive model for runway configuration at Congonhas Airport are still possible.

Acquiring more WRF data would be an immediate improvement, allowing the models to learn in an even more comprehensive way. This would enable them to capture specific trends and seasonalities for each time of day, week, and year more precisely.

Another future enhancement that could likely improve the model would be the inclusion of Severe Convective Weather (SCW) data, adding more meteorological information, specifically about potential weather severities. With these variables, which likely influence runway configuration, the model could produce more accurate predictions.

Additionally, a promising possibility would be to include data from the airport's weather station, which measures more frequently than METAR. Due to the difficulties in obtaining this data, it was not used in the present work. However, a future study incorporating it could potentially improve model accuracy.

Finally, we consider the possibility of an inertial behavior in the decision regarding runway configuration. This would mean that once the runway direction is changed, it is unlikely to change again in the near future. If this proves to be true, incorporating the classifications of previous time intervals as input to the model could be a way to enhance its accuracy in future work.

We envision these potential improvements to develop an even more accurate and assertive model. This would enhance runway predictability at Congonhas Airport and, in the future, at other airports, making airspace usage increasingly efficient and safe.

Bibliography

AENA BRASIL. **Características do Aeroporto de Congonhas**. 2024. Available at: https://www.aenabrasil.com.br/pt/aeroportos/aeroporto-de-congonhas/caracteristicas-.html. Accessed on: 5 jun. 2024.

BERGSTRA, J.; BARDENET, R.; BENGIO, Y.; KÉGL, B. Random search for hyper-parameter optimization. **Journal of Machine Learning Research**, JMLR.org, v. 13, n. Feb, p. 281–305, 2012.

BREIMAN, L.; FRIEDMAN, J.; STONE, C. J.; OLSHEN, R. A. **Classification and regression trees**. Belmont, CA: CRC press, 1984.

DECEA. **ICA 100-37/2020 - Serviços de Tráfego Aéreo**. Rio de Janeiro, RJ, 2020. 147 p. Seção 6.5.

DECEA. **ICA 105-8/2020 - Métodos de Avaliação de Previsões Meteorológicas**. Rio de Janeiro, RJ, 2020. 19 p. Seção 4.

DECEA. **API REDEMET: O que é?** 2024. Acessado em: 12-junho-2024. Available at: https://ajuda.decea.mil.br/base-de-conhecimento/api-redemet-o-que-e/.

DECEA. **Como decodificar o METAR e o SPECI**. 2024. Acessado em: 12 de junho de 2024. Available at: https://ajuda.decea.mil.br/base-de-conhecimento/como-decodificar-o-metar-e-o-speci/.

EUROCONTROL. **European Airline Delay Cost Reference Values: Final Report**. [*S.l.*], December 2015. Available at: https://www.eurocontrol.int/sites/default/files-/publication/files/european-airline-delay-cost-reference-values-final-report-4-1.pdf.

FRANZINI, B. M.; DOMINGUES, T.; SILVA, I. M.; SILVA, P. E. da; MELO, A. L. E.; FORSTER, C. H. Q.; LIMA, J. P. O.; MáXIMO, M. R. O. A. Predicting runway configuration in the context of the congonhas airport using machine learning. *In*: ACCAER. **XXI SITRAER - Air Transportation Symposium**. **Proceedings** [...]. Fortaleza, CE, Brazil, 2024. p. 9. PAPER ID: SIT904960.

FRIEDMAN, J. H. Greedy function approximation: a gradient boosting machine. **Annals of statistics**, p. 1189–1232, 2001.

GOSLING, G. D. Identification of artificial intelligence applications in air traffic control. **Transportation Research Part A: General**, Pergamon Journals Ltd., v. 21, n. 1, p. 27–38, 1987.

HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. **The Elements of Statistical Learning: Data Mining, Inference, and Prediction**. New York, NY: Springer Science Business Media, 2009.

HERREMA, F.; CURRAN, R.; HARTJES, S.; ELLEJMI, M.; BANCROFT, S.; SCHULTZ, M. A machine learning model to predict runway exit at vienna airport. **Transportation Research Part E: Logistics and Transportation Review**, 2019.

IATA. **Brazil: Value of Aviation**. Montreal, QC, 2018. Available at: https://www.iata-.org/en/iata-repository/publications/economic-reports/brazil–value-of-aviation/.

ICEA. **Sistema de Previsão de Reconfiguração de Pista**. 2024. Acessado em: 12-junho-2024. Available at: https://pesquisa.icea.decea.mil.br/cabeceira/.

KE, G.; MENG, Q.; FINLEY, T.; WANG, T.; CHEN, W.; MA, W.; YE, Q.; LIU, T. Y. Lightgbm: A highly efficient gradient boosting decision tree. *In*: **Advances in Neural Information Processing Systems**. **Proceedings** [...]. Long Beach, CA, USA: Curran Associates, Inc., 2017. p. 3146–3154.

LAU, M. E. C. **Prediction of Runway Configuration Change Transition Timings Using Machine Learning Approach**. 2021. Nanyang Technological University. Available at: https://hdl.handle.net/10356/150292.

MACHADO, M. R.; KARRAY, S.; SOUSA, I. T. D. Lightgbm: An effective decision tree gradient boosting method to predict customer loyalty in the finance industry. *In*: IEEE. **2019 14th International Conference on Computer Science and Education (ICCSE)**. **Proceedings** [...]. Toronto, Canada: IEEE, 2019. p. 1111–1116.

MIDTFJORD, A. D.; BIN, R. D.; HUSEBY, A. B. A decision support system for safer airplane landings: Predicting runway conditions using xgboost and explainable ai. **Cold Regions Science and Technology**, p. 103556, 2021.

MURÇA, M. C. R.; HANSMAN, R. J. Identification, characterization, and prediction of traffic flow patterns in multi-airport systems. **IEEE Transactions on Intelligent Transportation Systems**, IEEE, v. 20, n. 5, p. 1683–1696, 2019.

OLIVEIRA, M. d.; EUFRÁSIO, A. B. R.; GUTERRES, M. X.; MURÇA, M. C.; GOMES, R. d. A. Analysis of airport weather impact on on-time performance of arrival flights for the brazilian domestic air transportation system. **Journal of Air Transport Management**, Elsevier, v. 91, p. 101974, 2021.

PANDALA, S. **Lazypredict: A Tool to Simplify the Training and Comparison of Several Models with a Single Line of Code**. [*S.l.*]: GitHub, 2024. `https://github.com/shankarpandala/lazypredict`. Acessado em: 12-junho-2024.

PODER360. **Setor aéreo movimentou 112,6 mi de passageiros no país em 2023**. 2023. Available at: https://www.poder360.com.br/infraestrutura/setor-aereo-movimentou-1126-mi-de-passageiros-no-pais-em-2023/.

PRANDINI, M.; PIRODDI, L.; PUECHMOREL, S.; BRáZDILOVá, S. L. Toward air traffic complexity assessment in new generation air traffic management systems. **IEEE Transactions on Intelligent Transportation Systems**, v. 12, n. 3, p. 809 – 818, cited by:

116, 2011. Available at:
https://www.scopus.com/inward/record.uri?eid=2-s2.0-80052360551doi=10-.1109%2fTITS.2011.2113175partnerID=40md5=fab60d399543b125cd2b36331778ccf3.

QUINLAN, J. R. Induction of decision trees. **Machine learning**, v. 1, n. 1, p. 81–106, 1986.

SILVA, I. M. da; FREITAS, A. S. de; LIMA, J. P. O.; SILVA, P. E. da; ALMEIDA, R. D. A.; SILVA, A. D. da; MELO, A. L. E. Influência da reconfiguração de pistas na ocorrência de esperas em voo: Um estudo de caso no aeroporto internacional de guarulhos. *In*: UFSC. **XX SITRAER - Air Transportation Symposium. Proceedings** [...]. Joinville, SC, Brazil, 2023. p. 129. PAPER ID: SIT2284. Available at: https://sitraer2023.paginas.ufsc.br/files/2023/12/XXSITRAER-Anais.pdf.

TELES, P. **Predicting Estimated Landing Time**. 2024. `https://github.com/pedroteles17/predicting-estimated-landing-time`. Accessed: 2024-06-12.

WANG, Y.; CHEN, J.; CHEN, X.; ZENG, X.; KONG, Y.; SUN, S.; GUO, Y.; LIU, Y. Short-term load forecasting for industrial customers based on tcn-lightgbm. **IEEE Transactions on Power Systems**, IEEE, v. 36, n. 3, p. 1984–1997, 2021. Available at: https://ieeexplore.ieee.org/document/9210509.

WORLD, A. **ACI World Airport Traffic Forecasts 2023–2052**. Montreal, 2023. Disponível em: `https://store.aci.aero/wp-content/uploads/2024/02/WATF-Executive-Summary.pdf`. Accessed on: 3 jun. 2024.

WORLD, A. **What to expect: Latest air travel outlook reveals short- and long-term demand**. Montreal, 2023. Disponível em: `https://aci.aero`. Accessed on: 3 jun. 2024.

ZHANG, J.; MUCS, D.; NORINDER, U.; SVENSSON, F. Lightgbm: An effective and scalable algorithm for prediction of chemical toxicity-application to the tox21 and mutagenicity data sets. **Journal of Chemical Information and Modeling**, ACS Publications, 2019. Available at: https://pubs.acs.org/doi/10.1021/acs.jcim.9b00284.

FOLHA DE REGISTRO DO DOCUMENTO

1. CLASSIFICAÇÃO/TIPO	2. DATA	3. DOCUMENTO Nº	4. Nº DE PÁGINAS
TC	25 de novembro de 2024	DCTA/ITA/TC-123/2024	63

5. TÍTULO E SUBTÍTULO:

A *LightGBM* Model for Runway Configuration Prediction Applied to Congonhas Airport

6. AUTOR(ES):

Bruno Monteiro Franzini

7. INSTITUIÇÃO(ÕES)/ÓRGÃO(S) INTERNO(S)/DIVISÃO(ÕES):

Instituto Tecnológico de Aeronáutica – ITA

8. PALAVRAS-CHAVE SUGERIDAS PELO AUTOR:

Runway Configuration; Air Traffic Control; LightGBM; Meteorological Data

9. PALAVRAS-CHAVE RESULTANTES DE INDEXAÇÃO:

Aprendizagem (inteligência artificial); Aeroportos; Capacidade aeroportuária; Pistas (de pouso e decolagem); Desenvolvimento de software; Trafego aéreo; Computação.

10. APRESENTAÇÃO: **(X) Nacional** ( ) **Internacional**

Trabalho de Graduação, ITA, São José dos Campos, 2024. 63 páginas. Orientador: Marcos Ricardo Omena de Albuquerque Máximo; coorientadores: Cap André Luiz Elias Melo e Me. Jean Lima, Publicado em 2024.

11. RESUMO:

The complexity and criticality of airspace control, combined with the increasing demand for flights, underscore the need for advanced tools to enhance the optimization and safety of operations. Given the numerous variables that influence air traffic and the difficulty of analytically assessing the impact of each, Artificial Intelligence (AI) techniques provide a viable strategy for more accurately weighting each factor. Considering that runway configuration change is one of the most influential variables in air traffic management, this work presents the development of a system to predict runway configuration at Congonhas Airport (SBSP) using Gradient Boosting techniques, specifically LightGBM. The developed LightGBM-based classification model utilizes data from the Meteorological Aerodrome Report (METAR), Terminal Aerodrome Forecast (TAF), and Weather Research and Forecasting (WRF) to predict potential runway reconfigurations, determining which side of the runway will be in use. Predictions are made for 12 future time points, spaced 15 minutes apart, providing forecasts from the next 15 minutes up to 3 hours ahead. The model achieved a 98.6% accuracy on the test set and, when tested on an unseen data period outside the training and test set, maintained a high accuracy of 92.0% with an F1-Score of 91.8%. These results highlight the effectiveness of the developed model, outperforming traditional rule-based methods currently in use at ICEA (Institute of Airspace Control), which achieve approximately 79% accuracy. The objective is to provide a more precise quantification of meteorological variables' influence on airport operations, contributing to more efficient and safer airport activity planning.

12. GRAU DE SIGILO:

(X) OSTENSIVO ( ) **RESERVADO** ( ) **SECRETO**