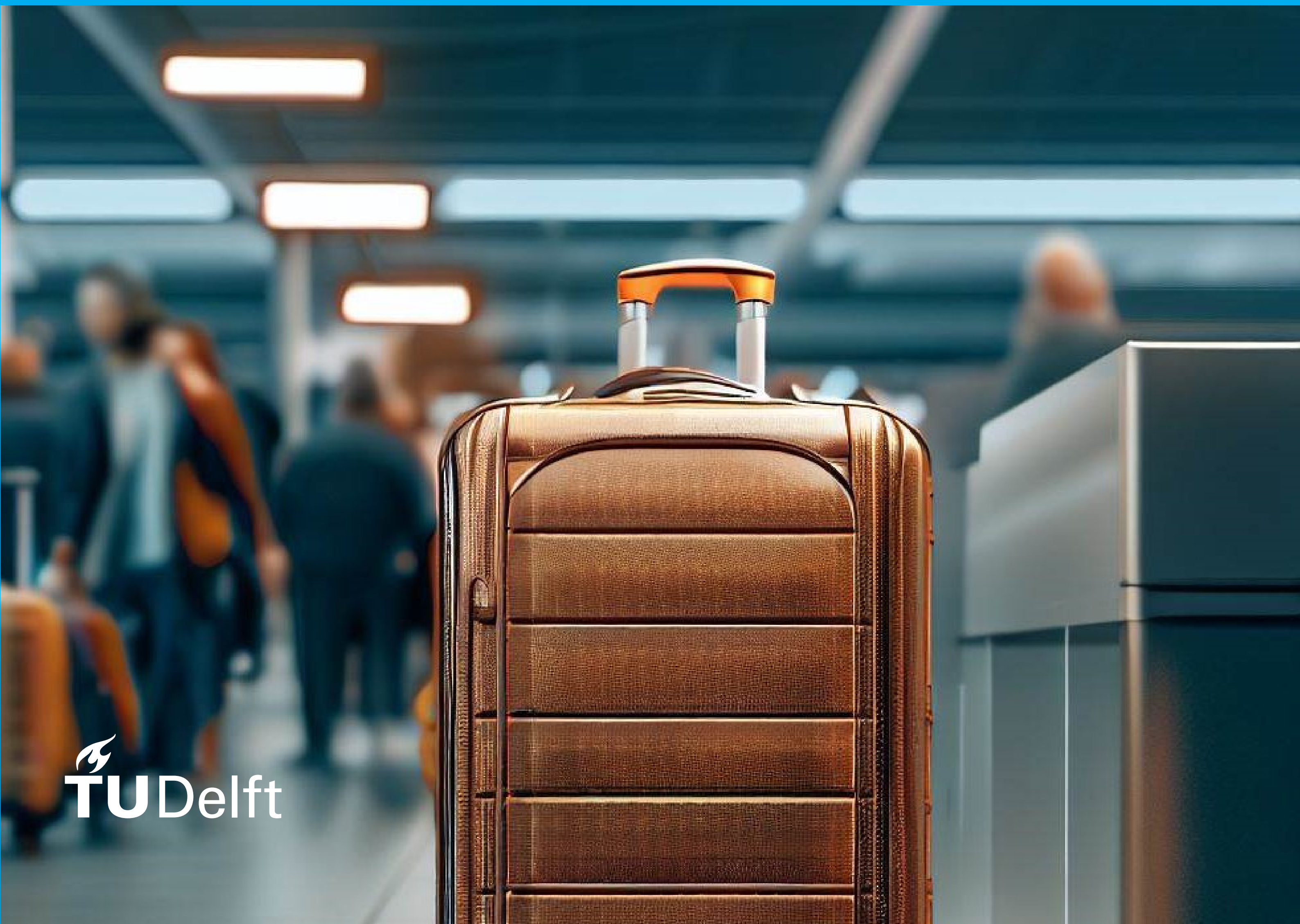
Enhancing Baggage Handling Operations at Hub Airports through Sequential Forecasting and Resource Allocation

Master Thesis Aerospace Engineering

Mitchell de Keijzer



TU Delft

Enhancing Baggage Handling Operations at Hub Airports through Sequential Forecasting and Resource Allocation

Master Thesis Aerospace Engineering

by

Mitchell de Keijzer

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Thursday July 13, 2023 at 10:00.

Student number:	4496701	
Project duration:	November 2022 – July 2023	
Thesis committee:	Dr. M.D. Pavel	TU Delft, Chair
	Dr. A. Bombelli	TU Delft, Supervisor
	ir. P.C. Roling	TU Delft, Examiner
	drs. M.A. van Gaalen	Schiphol Airport, Supervisor

TU Delft

Acknowledgements

I hereby present my final thesis project for the completion of my master's degree at the Faculty of Aerospace Engineering, Delft University of Technology. As I reflect on my journey, having embarked on the study of Aerospace Engineering during my bachelor's degree, I am filled with a mix of excitement and nostalgia as my time as a student draws to a close. My experience in Delft and at the University has been immensely enjoyable, and I will fondly cherish the days spent wandering these halls. Nevertheless, I am eagerly anticipating the new challenges and opportunities that await me, and I will always look back on my time at TU Delft with a wide smile and countless beautiful memories. During my thesis project at Schiphol Airport, I had the privilege of concluding my student career in an environment where complex challenges abound, yet everyone collaborates diligently to deliver the best possible services. Therefore, before I encourage you to dive into the details of my work, I would like to express my gratitude to the key individuals who played a crucial role in this project.

First and foremost, I extend my heartfelt appreciation to Alessandro. Your genuine curiosity and insightful questions demonstrated a deep understanding of my research. Your guidance in navigating complex challenges and active participation in collaborative brainstorming sessions ensured my readiness for the subsequent stages of my thesis. Beyond your professional expertise, your kindness created a pleasant atmosphere for both serious discussions and casual conversations. I would also like to thank Paul Roling for his valuable contributions to milestone meetings, engaging in discussions about my work, and providing intelligent ideas for the way forward. Furthermore, my gratitude extends to Marilena Pavel for serving as the chair of my thesis committee.

From Schiphol Airport, I would like to express my gratitude to Mark van Gaalen. Despite the challenging circumstances of our initial call, with me atop a hill in Malaysia battling a shaky internet connection, we embarked on this project together undeterred. Throughout the project, your creativity in approaching complex tasks and positive outlook never wavered. I am also grateful to Han Mackor for sharing expert knowledge in baggage handling, which proved indispensable to the project's success. Although it was not officially designated, I have always regarded you as one of my thesis supervisors. Additionally, I would like to thank Timo van Frankenhuyzen, who joined Schiphol a few months after I began my thesis, coincidentally in a role that required the tool I was developing. We already knew each other from the time we were both students and immediately had many insightful discussions that greatly propelled my progress, alongside our friendly conversations. Finally, my sincere appreciation goes to the entire FACT department for the wonderful experience and genuine interest shown in my project.

Beyond the individuals at Schiphol, I would also like to express my gratitude to all my friends with whom I studied, worked, and enjoyed memorable moments throughout my time as a student in Delft. In special, I would like to thank Imke, who always stood by my side and provided a listening ear during moments when I was stuck with my project, allowing me to vent and find solace. Last but certainly not least, my deepest appreciation goes to my family, who have consistently offered a listening ear, unwavering support, and endless motivation throughout my entire student journey.

As we wrap up this chapter, let's come together, rejoice, and toast to the incredible memories we've created together!. Cheers to each and every one of you!

Mitchell de Keijzer
Rotterdam, June 2023

Contents

List of Figures

List of Tables

List of Abbreviations

AR	Autoregression
ARIMA	Autoregression Integrated Moving Average
BF	Baggage Factor
FACT	Forecasting, Analysis, and Capacity Management
LU	Loading Unit
MA	Moving Average
MILP	Mixed Integer Linear Programming
MLR	Multiple Linear Regression
MUA	Make-Up Area
RF	Random Forest
SARIMAX	Seasonal Autoregression Integrated Moving Average with eXogenous factors

Introduction

The demand for air transportation at Schiphol Airport has experienced a substantial increase over the past three decades. From 1992 to 2019, the number of passengers grew from approximately 19 million to over 70 million [1]. However, the COVID-19 pandemic in 2019 caused a significant decline in passenger numbers to 20 million in 2020 and 25 million in 2021. Since 2022, there has been a rapid recovery in air travel demand, with passenger numbers approaching pre-pandemic levels in 2023. Nevertheless, this surge in passenger volume presents a capacity challenge for Schiphol Airport, particularly during the months of March to September, which coincide with the summer and holiday seasons. The challenges primarily arise for outbound flights due to the complexity of baggage handling processes. The increase in passengers with checked baggage can lead to longer queues at check-in desks and customs checkpoints. Additionally, the sorting system in the baggage handling system, the capacity in the baggage halls, workforce capacity, and available resources may be insufficient if the quantity of baggage items during specific time periods is unknown. Moreover, the COVID-19 pandemic and the introduction or increase of baggage fees have influenced passenger behaviour regarding the decision to bring checked baggage.

Accurate forecasting of checked baggage quantities is crucial for optimising the entire baggage handling process. To achieve this, scientifically supported forecasting and optimisation models will be utilised. The objective of this study is to develop a forecast model that predicts baggage factors for individual flights over time spans of 7, 30, and 60 days. Subsequently, a baseline model will be constructed using the forecasting outputs to optimise baggage handling processes and allocate resources effectively, with a focus on the Make-Up Areas (MUAs) in the baggage halls. MUAs are the areas where laterals and carousels are located, along with corresponding Loading Units (LUs) where checked baggage items are loaded and transferred to the aircraft. The data used for the models includes outbound flight information from January 2022 to March 2023.

This research is conducted in collaboration with Schiphol Airport, specifically the Forecasting, Analysis, and Capacity Management (FACT) department. The need for a baggage factor forecast tool has increased in recent years due to rising demand and changing patterns. Previously, baggage demand was lower, patterns were relatively easy to identify, and a high-level forecast could be created using Excel and expert knowledge. Existing literature on this subject is limited, with a focus on passenger demand forecasting rather than checked baggage forecasting. From a societal standpoint, a positive outcome of this research could also contribute to improving the customer experience. An improved baggage handling process resulting from accurate forecasting and optimisation contributes to customer satisfaction by minimising wait times, reducing disruptions, ensuring reliability, and enhancing convenience throughout the journey.

This report is divided into three distinct parts. The first part consists of a scientific paper that details the methodology employed in the models, presents the obtained results, and draws conclusions based on the findings. The second part encompasses an extensive literature review conducted at the beginning of the research, which includes a description of the problem, the primary research question along with its sub-questions, and an exploration of the current state-of-the-art research based on qualitative sources readily available. Additionally, this part outlines various solution approaches for the identified problems. Lastly, the third part comprises supplementary work conducted as part of this study.

I

Scientific Paper

Enhancing Baggage Handling Operations at Hub Airports through Sequential Forecasting and Resource Allocation

Keijzer, M.T. de
Delft University of Technology, Delft, The Netherlands

Abstract

Baggage handling operations in airports have become substantially more challenging due to the surge in air transportation demand witnessed in the last three decades combined with an even higher pressure on the operations following the COVID-19 pandemic. The intricate nature of checked baggage requirements, impacting resource allocation and personnel scheduling, necessitates an integrated approach for problem-solving. This research paper aims to enhance baggage handling operations by predicting the baggage factor (BF) for individual outbound flights. The BF represents the ratio of checked baggage items to the number of passengers aboard an aircraft. The objective of this study is to create a forecast model that predicts baggage factors for individual outbound flights over a time span of 7, 30, and 60 days, and subsequently construct a baseline model that leverages the forecasting outputs to optimise baggage handling processes and allocate resources effectively. A novel approach is proposed that employs historical flight data within gradient boosting models to forecast the baggage factor for future flights. Additionally, a case study is conducted by developing a Mixed Integer Linear Programming (MILP) model to minimise space utilisation within a baggage handling facility during the busiest period of the day, employing as few Make-Up Areas (MUAs) as possible. The results indicate that LightGBM, a gradient boosting technique, outperforms other gradient boosting techniques in terms of performance and computation time, achieving an accuracy score for the BF prediction ranging between 78-83% for the three forecast periods. Leveraging these predictions, the MILP model demonstrates that only 3 to 5 MUAs are required in an ideal situation in the baggage handling facility during the busiest period on various days.

Keywords: Gradient Boosting, MILP, Baggage Factor, Baggage Handling, Airport Operations, Forecasting, Optimisation, Schiphol Airport

I. Introduction

The demand for air transportation has soared in the last decades. Using Schiphol Airport (AMS) as an example, AMS has experienced a significant growth over the past three decades. In 1992, the airport catered to approximately 19 million passengers, a figure that escalated to over 40 million passengers in 2002. The trend continued, reaching a record high of over 70 million passengers in 2019 [Royal Schiphol Group, 2019]. However, the outbreak of the COVID-19 pandemic in 2019 disrupted this trajectory, resulting in a sharp decline in passenger numbers to 20 million in 2020 and 25 million in 2021. Nonetheless, there has been a rapid resurgence in air travel demand since 2022, and this upward trajectory continues into 2023, with passenger numbers approaching pre-pandemic levels. However, this surge in passenger volume poses a capacity challenge for AMS, exerting heightened pressure on logistics and operations, particularly during the months of March to September, coinciding with the summer and holiday seasons. These capacity challenges during this period result in longer queues at check-in desks and customs checkpoints.

Furthermore, the airport has faced challenges related to staffing. With the resumption of international travel, AMS has encountered issues with staff shortages [le Clerq, 2022], which were exacerbated by a strike involving KLM baggage handlers in April 2022 [NOS, 2022]. Dissatisfaction among employees has been fuelled by the heavy workload and inadequate staffing levels. While resolving the staffing shortage is a crucial aspect of the solution, optimising services, operations, and logistics could also contribute to mitigating such situations to some extent. Accurate demand forecasting plays a pivotal role in addressing operational challenges and improving efficiency across various domains. Numerous studies have been conducted on this topic for airports around the world, including research on demand forecasting for air traffic passenger demand [Xu et al., 2019]. Within AMS, demand forecasting models have been developed and successfully utilised to anticipate passenger demand for many years. These models have also enabled an informed estimation of baggage demand based on historical data. However, the introduction of baggage fees by airlines in 2008 led to changes in consumer behaviour [Johnston, 2013, Seaney, 2017]. A subsequent increase in baggage fees by major European airlines in 2017 further altered behaviour [Baldanza, 2020, BBT, 2016]. Due to the emergence of the COVID-19 pandemic, forecasting baggage load factors based on historical data has become more challenging and uncertain. This is due to the non-representative nature of data from the years 2020 and 2021, as well as the unknown changes in consumer behaviour following a two-year hiatus in air travel. These changes have introduced complexities that render the current forecasting approach inadequate for accurately predicting baggage demand. Consequently, a novel forecasting model is required to effectively anticipate this specific type of demand. It is worth noting that while previous research on airport operations has predominantly focused on passengers, facilities, and aircraft, there has been relatively limited emphasis on baggage [Ma et al., 2021]. Further investigation and analysis in this area are warranted.

Baggage encompasses three distinct categories: checked baggage, hand luggage, and personal items. To ensure passenger safety, all baggage items undergo screening procedures. While hand luggage and personal items are carried by passengers into the aircraft without requiring additional logistical processing by AMS, checked baggage undergoes a complex operation behind the scenes. In 2019, the total number of checked baggage items at AMS reached 53 million, with a daily range of 120,000 to 180,000 pieces [Royal Schiphol Group, 2018]. This process becomes intricate due to the high volume. Alongside the checked baggage processed through the departure halls, a significant proportion is contributed by transfer passengers, accounting for nearly 40% of all checked baggage. AMS serves as a hub for its home carrier, KLM, and the SkyTeam partners, which is a reason for the contribution to the substantial amount of transfer baggage [Royal Schiphol Group, 2018].

Accurate forecasting of the quantity of checked baggage is vital for optimising the entire baggage handling process. This optimisation can be achieved through the utilisation of scientifically supported forecasting and optimisation models. Therefore, the objective of this study is to create a forecast model that predicts baggage factors for individual outbound flights over a time span of 7, 30, and 60 days, and subsequently construct a baseline model that leverages the forecasting outputs to optimise baggage handling processes and allocate resources effectively. This paper is believed to be the first study to sequentially employ a Gradient Boosting model for forecasting the baggage factors of outbound flights of a hub airport and utilise the prediction to allocate resources accordingly. The resource allocation strategy primarily aims to optimise the utilisation of space within baggage facilities, specifically by minimising the number of active carousels and laterals during peak time periods on a given day. It is important to note that, in this context, carousels refer to the conveyor belts within the baggage facilities where baggage items are placed after they have exited the baggage handling system, and not the belts found in the reclaim hall where passengers retrieve their bags. Laterals serve a similar purpose as carousels but are distinct in that they are straight conveyors without a belt. These laterals and carousels are utilised by baggage handlers to efficiently process and transfer baggage items onto trailers for transportation to the aircraft. The dataset utilised for the models comprise information from all departing flights between January 2022 and March 2023. Moreover, it is important to note that this research is conducted in collaboration with AMS, in particular, in partnership with its department Forecasting, Analysis, and Capacity Management (FACT).

The paper's structure is organised as follows: in section II, a concise overview of the current academic literature pertaining to both forecasting and resource allocation is provided. Subsequently, a comprehensive explanation of the problem is presented in section III. Following that, section IV describes the forecast model's methodology and the precise formulation of the Mixed Integer Linear Programming (MILP) model. These elements culminate in the presentation and analysis of the results in section V. Lastly, section VI offers the research's conclusive remarks and identifies potential areas for future research.

II. Literature Review

The task of achieving an optimal approach for forecasting checked baggage items and efficiently allocating resources based on the prediction results is a multifaceted undertaking. Despite extensive research conducted on each of these individual topics, no comprehensive studies have been conducted to investigate their combined application. Nevertheless, numerous studies exist that focus on each of these subjects independently. This section provides a detailed examination of the most significant works available in the literature concerning these respective areas of interest.

II.A. Forecast Methods

There are may different forecast methods available. However, not all of them are able to predict accurately due to the complexity of the data. Looking from a high level perspective, there are three basic types of forecasting, namely qualitative techniques, time-series analysis & projection, and causal models. Qualitative forecasting involves predicting future outcomes based on subjective assessments and expert judgement using qualitative data. Time-series & projection forecasting predicts future values based on past values and current trends using regression analysis with existing data for estimation. It's useful for long-term trend analysis and variable forecasting. Time-series models don't focus on explaining relationships. Causal models use statistical methods to identify relationships between variables and predict outcomes. They rely on past data and variables to forecast event outcomes. [Chambers et al., 1971]

As mentioned before, most airport operations research in the literature primarily focuses on passengers, facilities, or aircraft, with limited studies specifically addressing baggage [Xu et al., 2019]. One notable study by Cheng et al. [2014] examines forecasting methods for departure flight baggage demand, highlighting the importance of establishing a scientific foundation for efficient resource allocation in the checked baggage stage. The study compares a Multiple Linear Regression (MLR) model and a Back-Propagation (BP) neural network, finding that the MLR model yields a lower average relative error. Moreover, modifying the input data from all flights to solely single airline flights or flights with the same destination reduces the average relative error. Another study conducted by Ma et al. [2021] proposes a Seasonal Auto-regressive Integrated Moving Average (SARIMA) model to predict checked baggage demand for departure flights, aiming to optimise efficiency during the check-in process. The model demonstrates accurate long-term demand forecasting, enabling proactive resource allocation.

Next to these studies on predicting checked baggage, numerous studies were conducted on forecasting passenger demand. The forecast methods are varying from simpler methods to more complex ones. An example is the study of Chen et al. [2012], where a modified moving average (MA) method was employed to forecast airline passenger numbers. The MA method yielded a notable error in its output, necessitating the adoption of a neuro-fuzzy model to mitigate the error. The neuro-fuzzy model demonstrated a substantial reduction in error, indicating that the MA method can be utilised for non-linear forecasting when appropriate data is employed. Another widely used method that can be seen as a simple forecast method is Exponential Smoothing (ES). ES can be extended by adding trend or seasonality, which are called Holt's method and Winter's method, respectively. Rusyana et al. [2016] conducted a comparative study between Holt's and Winters' method for forecasting the number of domestic passengers arriving and departing from Sultan Iskandar Muda International Airport in Indonesia. The findings revealed that the optimal model for this forecast was Winters' exponential smoothing method. However, upon examining the results of both methods and evaluating their accuracy using appropriate criteria, it was observed that both methods performed exceedingly well to excellent when the appropriate smoothing parameters were employed.

A more advanced and widely used method for time-series analysis and forecasting is Box-Jenkins [Box and Jenkins, 1976], which applies Auto-Regressive Integrated Moving Average (ARIMA). ARIMA can also be extended with seasonality and is also able to add exogenous factors, leading to a SARIMAX model. Tsui et al. [2014] employed a SARIMA model to predict airport passenger traffic for Hong Kong and projected its growth trend up to 2015, utilising monthly time-series data from January 1993 to November 2010. The empirical analysis demonstrated that the SARIMA model provided precise and dependable forecasting outcomes, as evidenced by its lower values of Mean Absolute Percentage Error (MAPE) and Root Mean Squared Error (RMSE). Furthermore, a comparison between the actual and predicted values indicated that the model yielded acceptable forecast errors. In another study conducted by some of the same authors, Tsui and Balli [2017] assert that external variables such as destination marketing and tourism marketing play a crucial role in influencing the demand for international arrivals from foreign countries, thereby impacting air passenger demand. They utilised a SARIMAX model incorporating variables such as GDP per capita, tourism marketing expenditure, flight seats, fuel prices, and exchange rates. The selected SARIMAX model outperformed the SARIMA model, demonstrating strong forecasting performance with low Mean Absolute Error (MAE), MAPE, and RMSE values.

Entering the more complex methods, supervised machine learning such as MLR is also often used for forecasting. The study of Srisaeng et al. [2015] aimed to develop prediction models for estimating domestic passenger demand for Australia's low-cost carriers, utilising enplaned passengers and revenue passenger kilometers as indicators of airline traffic demand. Two approaches were compared: classical MLR modelling and Artificial Neural Network (ANN) modelling. The study involved developing econometric models based on linear regression to analyse the statistical relationship between key factors influencing demand and the corresponding level of passenger traffic for low-cost carriers in Australia. Both models exhibited favourable performance in terms of model quality metrics. However, the comparison of modelling results indicated that the ANN approach outperformed classical MLR models, offering superior estimation capabilities.

In addition to MLR, decision tree-based methods like Random Forest (RF) and Gradient Boosting (GB) have been utilized in various tasks, including forecasting. While Gradient Boosting is commonly employed in literature for flight delay prediction, Random Forest has been applied in some instances for demand forecasting purposes. Given the numerous uncertainties and limited data available for predicting passenger volume in civil aviation, the research of Yang and Liu [2018] utilised daily passenger data from the Beijing to Sanya airline between 2010 and 2017. The study employed a RF prediction model, support vector regression (SVR) model, and neural network model to fit the airline data. The random forest algorithm demonstrated high precision, stability, and interpretability, making it widely utilised in supervised learning. It effectively addresses non-linearity issues and is robust to multicollinearity, as well as handling missing values and unbalanced data. In comparison, SVR excels in handling small samples, non-linearity, and high-dimensional recognition. In practical applications, random forest regression generally outperforms SVR. However, the neural network model in this study exhibited inferior predictive performance, primarily due to its suitability for larger datasets. Since this paper employed a smaller dataset, the neural network model displayed limited predictive capability. The study of Manna et al. [2017] explores the effectiveness of the GB paradigm for predicting air traffic delays. By employing a regression model based on this paradigm, an accurate and robust prediction model has been developed, allowing for detailed analysis of patterns in air traffic delays. The Gradient Boosted Decision Tree method demonstrated high accuracy in modelling sequential data, making it suitable for predicting day-to-day sequences of departure and arrival flight delays at specific airports. In their research, the model has been implemented using the Passenger Flight On-time Performance data obtained from the U.S. Department of Transportation to forecast flight arrival and departure delays. The results indicate superior accuracy compared to alternative methods.

Lastly, the use of neural networks is also more and more upcoming for forecast models. Several different types of neural networks can be used for forecasting like a Multilayer Perceptron (MLP), Recurrent Neural Networks (RNN), and

Convolutional Neural Networks (CNN). Also Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU) are a type of RNN and often used for forecasting.

Choi and Kim [2021] introduced MLP, RNN, LSTM models for predicting airport capacity. The experimental results indicated favourable performance for all three models, with RNN and LSTM surpassing the MLP model. However, it is noteworthy that the models were trained and validated using data from Hartsfield-Jackson Atlanta International Airport. When considering the generalisation to another airport, the MLP model exhibited robust transferability without the need for additional techniques. On the other hand, the RNN and LSTM models demonstrated accurate capacity predictions for another airport after undergoing fine-tuning procedures. While neural networks can offer powerful forecasting capabilities, they have limitations such as overfitting, computational demands, interpretability challenges, and potential difficulties in handling non-stationary data patterns [Tu, 1996].

In summary, based on the existing literature, it is evident that incorporating exogenous factors is crucial for enhancing the accuracy of forecasting the quantity of checked-baggage items for outbound flights from AMS. Additionally, considering the requirement for interpretability by AMS employees, it is imperative to select appropriate methodologies. Hence, this study has opted to employ machine learning techniques, specifically gradient boosting models, primarily due to their ability to address these concerns effectively.

II.B. Resource Allocation

Resource allocation or scheduling personnel commonly involves the utilisation of various optimisation methods and algorithms. In this context, two distinct categories of optimisation methods are typically considered: exact solution methods and meta-heuristic methods. Exact solution methods are optimisation techniques that, given sufficient computational resources and time, guarantee the identification of the globally optimal solution for a given problem. These methods are generally characterised by their high accuracy and reliability. However, they can be computationally demanding and may struggle to find solutions within reasonable time frames for large or complex problems. On the other hand, meta-heuristic methods are optimisation techniques that do not provide a guarantee of finding the globally optimal solution but are capable of generating good solutions within relatively short time periods. These methods draw inspiration from natural processes, imitating phenomena such as evolution, physics, and chemistry. While meta-heuristic methods are generally less accurate than exact solution methods, they excel in swiftly finding solutions for large or complex problems. Additionally, they are applicable to scenarios where mathematical models may be poorly defined or unknown.

The research of Emde et al. [2020] focuses on the optimisation of unit load devices (ULDs) preparation at an air cargo terminal. Airlines often encounter challenges in planning this process, as they need to allocate a limited number of workers to a restricted number of workspaces, while adhering to the requirements of an existing flight schedule. The objectives during ULD preparation include meeting the flight schedule, utilising terminal space efficiently, and minimising the maximum workforce employed over time. To enhance the efficiency of ULD preparation processes, the study proposes a MILP model and presents a generalised set partitioning reformulation for this complex problem. Utilising the latter formulation, various heuristic strategies are developed, some of which demonstrate near-optimal solutions to this NP-hard problem within a short time span of approximately 10 seconds. These strategies significantly outperform a commonly used rule of thumb in practice.

Kiermaier [2015] conducted a study focusing on the optimisation of baggage handling and ground handling processes at airports. The researcher provided a systematic overview of the baggage handling process, specifically examining the four main baggage streams: check-in, outbound, transfer, and inbound baggage. Through rigorous analysis, Kiermaier demonstrated the inherent complexity of these processes, establishing their NP-complete nature. To build upon existing knowledge in the field, the researchers conducted a comprehensive survey, organising previous research efforts and categorising relevant solution methods employed in baggage handling. Additionally, a novel and generic model formulation called the Generic Assignment and Scheduling Problem (GASP) was introduced, which captured the fundamental mathematical structure of each main baggage process. The main objective of the study was to establish a unified foundation for future research in baggage handling, allowing for a holistic perspective on the subject and facilitating the development of integrated solution methods. The presented model formulations serve as valuable tools for researchers and practitioners, providing a starting point for obtaining initial results and insights when examining baggage handling operations at airports.

In summary, drawing upon the existing body of literature, it is evident that employing a MILP model serves as a favourable initial step for establishing a baseline model to assess the correct implementation of a minimisation problem. Acknowledging the NP-hard nature of this problem, it is imperative to subject the model to targeted dataset subsets during testing, while also considering the evaluation of heuristic strategies in subsequent iterations.

III. Problem Description

This section presents an overview of the airport baggage prediction and handling problem, outlining the factors that must be considered when forecasting the number of checked baggage items for each individual flight. Furthermore, it highlights the crucial considerations for utilising the prediction results in resource allocation. The research objective is clearly stated, along with the problem's specific setting, encompassing the input and output requirements, as well as the underlying assumptions and constraints.

III.A. Airport Baggage Prediction & Handling

The "load factor" is a crucial concept in predicting the volumes of passengers and baggage on board of an aircraft. At AMS, there is an existing successful forecasting model for passenger load factor, which is the relative amount of passengers given the maximum number of passenger seats. However, accurately forecasting the number of checked baggage items per flight remains challenging. In the past, this was achieved through data analysis and expert judgement, but changes in consumer behaviour resulting from baggage fees introduced by specific airlines and the impact of the COVID-19 pandemic have increased the uncertainty of this method. Because passenger numbers might vary, directly predicting the number of baggage items might be off significantly when comparing it to the actual data. Therefore it is interesting to find the baggage load factor, which can later be multiplied with the forecast of the number of passengers to find the number of checked baggage items on a flight. The baggage load factor, hereafter described as "baggage factor" (BF) is defined as the number of checked baggage items divided by the number of passengers on board. The unknown shifts in consumer behaviour following the pandemic make it difficult to only use the expert judgement approach with historical data for accurate BF forecasts. Therefore a model needs to be built that is able to capture complex correlations between exogenous factors and can more accurately predict the BF.

The BF plays a crucial role in multiple aspects of airport operations and management as it provides valuable insights into the expected volume of checked baggage items per passenger. It is of significant importance when considering capacity management, resource allocation, employee scheduling, airline planning, security screening, and enhancing the overall passenger experience. Capacity management in airports relies on effectively utilising resources such as check-in lines, baggage handling systems, and baggage halls. This information allows airports to allocate appropriate resources, optimise capacity, and design efficient processes for check-in and baggage handling, ensuring smooth operations and minimising bottlenecks. Resource allocation and employee scheduling go hand in hand with capacity management and are vital for maintaining efficient airport operations. By understanding the BF, airport authorities can accurately determine the number of baggage handlers and other staff required to handle the expected volume of checked baggage items. The ability to make effective staffing decisions to manage baggage-related tasks will ultimately result in the prevention of delays or inefficiencies. Security screening is a critical aspect of airport operations, and the BF plays a role in assessing security measures. By considering the ratio of checked baggage items to passengers, authorities can evaluate the effectiveness and capacity requirements of baggage screening systems. This helps in ensuring appropriate security measures are in place to maintain safety standards. Passenger experience is greatly influenced by efficient baggage handling processes. By understanding the BF, airports can optimise the layout and capacity of baggage claim areas, minimising wait times and congestion. This contributes to a smooth and pleasant travel experience for passengers, enhancing their overall satisfaction. In order to maximise the benefits of utilising the BF, obtaining accurate predictions of the BF for each outbound flight becomes crucial. This detailed prediction enables a comprehensive examination of various aspects related to baggage handling on both macro and micro levels.

For outbound baggage handling, the airport operators need to make assignment and scheduling decisions. Each departing flight is assigned to at least one handling facility, and the start time for baggage handling is determined. The baggage handling process begins with the introduction of baggage items into the Baggage Handling System (BHS). This can be done through checked-in baggage, where passengers bring their bags to the airport and have them labelled at the check-in desk, or through incoming transfer baggage, which is offloaded from another flight by a handling company. The BHS scans the baggage label to obtain relevant information such as the flight number, performs security screening, and determines the terminal to which the baggage needs to be transported. The BHS also determines whether robots or manual methods using laterals and carousels should be used for loading the baggage onto the aircraft. Bags that arrive earlier than the moment a lateral or carousel opens for a specific flight, are moved into the storage system, i.e. the buffer. Bags stored in the buffer can only be removed once the baggage handling for the respective flight has commenced. In some cases, the airport operator also decides on the storage location, such as the buffer lane, for early baggage. The number of stored bags at any given time is limited by the storage capacity. Work groups are assigned to handling facilities in order to load the bags onto the trailers (referred to as LUs) of the car that transports them to the aircraft. The assignment of flights to handling facilities and the scheduling of baggage handling aim to avoid peak workloads at the facilities. An objective of outbound baggage handling is to optimise the layout of baggage carousels and laterals in a handling facility, taking into account the workload of the workers. This can be achieved by minimising the number of carousels and laterals required during peak hours while considering the corresponding allocation of workers. In subsequent sections of this paper, the designated region where a lateral or carousel is situated in conjunction with the associated area where the LUs are positioned is referred to as a Make-Up Area (MUA).

III.B. Research Objective

The research objective of this study is to enhance the efficiency of baggage handling operations at a hub airport, in this case study AMS, through the utilisation of predicted BFs for resource allocation. To achieve this objective, two distinct models are developed.

The first model focuses on forecasting and utilises a historical dataset comprising various features. The data is prepared by performing necessary preprocessing steps, such as feature selection based on correlation analysis. These selected features are then incorporated into the model. The model is trained using suitable forecasting methods, and hyperparameters are tuned to optimise its performance. Result validation is conducted by comparing the forecasted values with actual values, and the importance of each feature is assessed.

Forecasts are generated for individual outbound flights, encompassing forecasting intervals of 7, 30, and 60 days. These particular forecast periods originate from the forecasting process conducted by FACT. The 60 day forecast provides an overarching depiction of the baggage factor for the subsequent two months, which is critical information for effective communication with security personnel, aiding them in their planning activities. On the other hand, the remaining two forecasts serve to offer a more intricate understanding of the baggage demand within a shorter time frame. This nuanced insight facilitates the fine-tuning of security planning protocols and also informs various aspects of resource allocation and personnel scheduling in operational contexts.

The second model aims to optimise the allocation of batches of bags and workers to MUAs at different time periods. This model considers the output of the first model, which predicts the number of incoming baggage items to the BHS. A certain time interval is made regarding the first possible arrival time of bags at the MUAs and the departure time of the flight. Based on this interval, batches of bags can be assigned to LUs within specific MUAs, with a specified number of workers at a specific time within the interval. The model identifies the busiest time period and seeks to minimise the number of used MUAs during that period. This optimisation allows for optimal utilisation of the available space within the handling facility.

III.C. Problem Setting

This subsection provides a comprehensive overview of the problem setting, the expected output, the underlying assumptions, and the constraints utilised throughout the models.

Input

The input data for this both models is derived from four primary sources:

- Baggage analysis data: this dataset encompasses all baggage items entering the BHS and includes comprehensive labels associated with each baggage item. These labels contain various information, such as the baggage item's unique identifier (ID), the date and time of processing, the scheduled departure date and time, punctuality status (on-time or delayed), baggage type (check-in or transfer), outbound flight airline code, outbound flight designator, destination airport, and flight classification (European or intercontinental). This data enables the determination of the number of checked baggage items for each flight on a daily basis.
- Flight schedule data: this dataset comprises information regarding the direction (arrival or departure) of all flights, the service type, flight designator, scheduled departure time, destination airport, the number of passengers on board (both transfer and Origin & Destination (O&D) passengers), aircraft type, the total number of seats available, and the country of destination. This dataset represents a fixed schedule containing historical data of flights that have actually departed, providing accurate and comprehensive information.
- Passengers on board (pax) prediction data: at AMS, an existing forecast model is implemented to predict the number of passengers for both transfer and O&D flights. This dataset shares similarities with the flight schedule data, with the only difference being that the flight schedule represents future schedules, not all of which will necessarily depart at the specified time, and the passenger numbers are predictions rather than actual figures.
- Additional data of baggage handling facilities: some data on the general layout of the AMS baggage handling facilities, such as the number of laterals and carousels.

Output

Each model has its own outputs. The forecast model output is twofold:

- BF per flight: the main objective of the forecast model is to produce the BFs for each flight for 7, 30, and 60 days. This output contains the BF for all predicted passengers on board, meaning both transfer and O&D passengers.
- Checked baggage items on board (bax) per flight: the reason why the BF needs to be determined is to finally make an estimation of the amount of checked bags a flight will take with it. With the passenger forecast and the BF forecast, the amount of baggage items is calculated per flight.

The resource allocation model output is fourfold:

- Minimum number of MUAs at peak period: the main solution of the resource allocation problem is to find the minimum number of MUAs in use at the peak period during a day. The problem is defined in one baggage handling facility with a number of MUAs, where all the batches of baggage items of specified flights need to be processed during a day.
- Number of MUAs in use: simultaneously with the previous output, also for the non-peak periods the number of MUAs is determined for a day.
- Number of assigned workers: the resource allocation problem can only be solved optimally by assigning a number of workers to the MUAs. Whilst it is not done for this research, it is possible to also minimise the amount of workers assigned to the MUAs to also find an optimal schedule for a smaller workforce and to try to reduce the workload by finding an optimum between the number of used MUAs and the number of bags to process at each MUA per worker.
- Flights assigned to MUA: the allocation problem also allocates the flights to specific MUAs at specific times. Each flight needs to be processed before it departures. The solution shows what flights are assigned to which MUA and at what time the processing starts.

Assumptions

In this study, several assumptions and simplifications have been made to facilitate the analysis, forecast, and resource allocation for baggage handling. Firstly, it is assumed that all predicted baggage items are on-time and this assumption is reflected in the preprocessed data. Even if a baggage item arrives late, it is still considered part of the flight's baggage count, even if it is flown with the next flight. The reason for this is to identify patterns related to the amount of checked baggage passengers tend to carry on different days, times, and destinations for each unique flight. Secondly, the analysis focuses on passenger baggage and excludes baggage items carried by the crew. The intention is to understand the behaviour of passengers when it comes to checked baggage, excluding the influence of the crew. Similar to the first assumption, this analysis aims to identify patterns related to the amount of checked baggage passengers tend to carry on different days, times, and destinations. Thirdly, the baggage handling facilities are assumed to have an infinite buffer capacity. This means that there are no constraints on the maximum number of baggage items that can be in the storage system at any given time. To further simplify the resource allocation model, it is assumed that all baggage items for a flight arrive at the MUA location simultaneously. Although, in reality, baggage items may arrive over a certain time period, the model considers that once the baggage handling for a flight begins, all items can be processed at once. Moreover, there are no limitations on the number of LUs or the availability of vehicles for transferring baggage items to the aircraft. Sufficient resources are assumed to be available to handle the transfer process. Concerning the time windows for a flight to have its baggage items processed, it is assumed that the MUA for European flights opens two hours before the scheduled departure, while the MUA for intercontinental flights opens three hours before the scheduled departure. Baggage items must be processed 30 minutes prior to the scheduled departure to account for the transfer time from the baggage handling facility to the aircraft. Once an MUA opens for a flight, it is assumed that the aircraft is ready to receive the baggage items for loading. Flights are already assigned to specific baggage facilities that are typically located near the stand or gate of the aircraft. Therefore, the assignment is not based on distance considerations but rather on the standard handling practices for these flights. In addition, it is assumed that workers assigned to the MUAs have a constant productivity. Lastly, the analysis does not take into account the sub-sorting of baggage types, such as economy O&D, transfer baggage, and priority/business class baggage. The prediction model used in the study does not distinguish between these types, making it impossible to sub-sort them in the resource allocation problem. In reality, different types of baggage items are loaded into separate LUs to ensure correct sorting, prioritisation, and accurate loading onto the aircraft.

Constraints

The resource allocation model is governed by various constraints to ensure its feasibility and practicality. These constraints can be classified into three primary categories: flight batch constraints, worker constraints, and MUA constraints. Flight batch constraints are in place to ensure that each batch of baggage items for a specific flight is processed exactly once. These constraints also guarantee that the flight batches are processed within the designated time window, aligning with the opening time of the corresponding MUA and the deadline for transferring the baggage items to the aircraft. Additionally, if a flight is assigned to a particular MUA, it is mandatory that the MUA is operational during that time. Worker constraints restrict the number of workers available for processing baggage items within a given time period to the maximum number of workers present. Furthermore, the allocation of workers to an MUA, along with their productivity, must be sufficient to process at least the required number of baggage items within the specified time period. MUA constraints ensure that the number of assigned MUAs in each time period does not exceed the total number of available MUAs. Moreover, each MUA can only be utilised once during a given time period. These constraints also ensure that the space capacity of an MUA is never exceeded, meaning that the number of LUs to be filled within a time period does not exceed the available space. By incorporating these constraints, the resource allocation model is designed to effectively manage the allocation of workers, MUAs, and time periods, while adhering to the limitations imposed by flight batches, worker availability, and MUA capacity.

IV. Methodology

This section describes the methodology employed in the forecast model and the MILP model. Initially, the procedures for data analysis and preprocessing are expounded upon. Subsequently, the forecast techniques are introduced and explained, followed by the introduction of supplementary simplistic forecast approaches and a description of feature selection methods. Furthermore, the metrics employed to assess the quality of the model are explained. Finally, the formulation of the MILP model is presented.

IV.A. Data Analysis & Preprocessing

As outlined in section III, the data for training the forecast model is obtained from multiple sources. Specifically, two sources are utilised: baggage analysis data and flight schedule data. The time range for this data spans from January 1, 2022 to March 31, 2023. These two datasets are combined to create a unified flight schedule dataset that includes historical information on the number of checked baggage items for each individual flight. This allows for the inclusion of the BF in the data by calculating the ratio of baggage items to passengers. To facilitate model training and evaluation, the data is divided into a training set and a test set. The training set comprises the initial 80% of the data (January 1, 2022 to December 31, 2022). This training set is further partitioned into K equally sized subsets, referred to as validation sets, without shuffling. These validation sets are used for K-fold cross-validation, a technique employed to optimise model parameters during the selection process, which will be elaborated upon in subsection IV.B. The test set consists of the remaining 20% of the data (January 1, 2023 to March 31, 2023). This test set is exclusively employed for evaluating and comparing the final models.

The combined dataset encompasses various features that hold significance for the model. These features can be categorised into destination, date & time, airline, aircraft, and passengers. Within the destination category, essential features include the destination airport and country. Additionally, a distinction is made between destinations within Europe and those categorised as intercontinental, which will be referred to as "outbound range" hereafter. The date & time features represent the scheduled departure dates and times for the flights. Airline features consist of the airline's IATA code and the flight designator. Aircraft features pertain to the aircraft's configuration, including its type and seating capacity. Lastly, passenger features encompass the number of passengers on board, along with the breakdown between O&D passengers and transfer passengers, as indicated in the data. In addition to the initial set of features, it is possible to engage in feature engineering to enhance the forecast model's performance. Feature engineering involves creating additional predictors. In the case of the present dataset, feature engineering primarily focuses on the date & time features. By leveraging the scheduled departure date-time, various supplementary features can be generated. These additional features encompass the year, month, week, and weekday on which the flight departed. Furthermore, Dutch holidays are incorporated as supplementary features, encompassing information about the holiday type and its duration.

Various types of features are present in the dataset, ranging from integers to date-time values to categorical types. In the context of the Gradient Boosting model, which utilises a regression approach (as will be described in subsection IV.B), the target variable being a numerical value necessitates encoding categorical features that cannot be directly used in the regression model. To accomplish this, the Target Encoder from the scikit-learn library is employed. Target Encoder is a feature encoding technique that transforms categorical target variables into numerical representations. It assigns a numerical value to each category by considering statistical measures such as the mean of the corresponding target variable. This encoding method captures the relationship between categories and the target variable, resulting in a more informative representation for machine learning models to facilitate predictions.

Feature Correlation

Prior to incorporating the features into the model, it is crucial to assess the interrelationships between these features and their connection to the target variable, the BF. This examination is accomplished by constructing a correlation matrix employing the Pearson's correlation coefficient formula. The resulting correlation matrix reveals the associations among the variables, providing insights into their magnitude and direction of correlation. The Pearson's correlation coefficient ranges between -1 and 1. The closer the coefficient is to these extremes, the stronger the correlation. Conversely, a coefficient close to 0 indicates a weak correlation between the variables. These findings aid in the identification of interdependent features and the determination of influential factors for prediction.

IV.B. Gradient Boosting

Gradient boosting is a machine learning technique that utilises a combination of multiple small Decision Tree models to make predictions. These decision trees are made distinct from each other through a process known as *boosting*, which is an iterative procedure. Boosting involves intelligently adding more weak learners to the ensemble model. At each step of the process, the individual data points are assigned weights, giving less importance to the ones that have already been well predicted. The new weak learners then focus on learning the aspects of the data that have not yet been understood, thereby enhancing the ensemble. The iterative nature of this process is termed "gradient boosting" due to the incorporation of gradients. In mathematics, a gradient represents a vector field of partial derivatives that indicates the direction of the steepest slope. When adding additional trees to the model, the objective is to introduce a tree that effectively explains the remaining variation not accounted for by the previous trees. Therefore, the target for the new tree is defined as the difference between the true values y and the predicted values $\hat{y}$. This can be expressed as the negative partial derivative of the loss function with respect to the predicted values: $y - \hat{y} = -\frac{\delta L}{\delta \hat{y}}$. By setting this difference as the target for the new tree, it is ensured that the tree explains a maximum amount of additional variation in the overall gradient boosting model. This rationale behind the name "gradient boosting" arises from the utilisation of gradients to guide the addition of trees to the model. For a more detailed description of gradient boosting, see Bentéjac et al. [2021], Hastie et al. [2001]. A visual representation of gradient boosting is presented in figure 1. Gradient boosting can both be used for classification and regression problems, where categorical forecasting deals with discrete categories, while regression focuses on continuous numerical values. There are several gradient boosting algorithms that perform slightly different, of which three are considered for this forecast problem: XGBoost, LightGBM, and CatBoost.

XGBoost

XGBoost [Chen and Guestrin, 2016] emerged as an early and widely embraced gradient boosting framework, sustaining its popularity owing to its commendable performance and scalability. It employs a level-wise approach to construct decision trees, wherein each tree is incrementally developed in layers. Nonetheless, this methodology may present inefficiencies when confronted with imbalanced data or datasets with numerous missing values. XGBoost encompasses regularisation techniques, namely L1 and L2 regularisation and tree pruning, to address the concern of overfitting. Additionally, it offers provisions for parallel processing and distributed computing, facilitating expedited training and prediction on extensive datasets. XGBoost also incorporates diverse hyperparameter tuning options, allowing for fine-tuning of the model. The hyperparameters that will enter the search space used for XGBoost are presented in the Appendix with explanation and reasoning.

LightGBM

LightGBM [Ke et al., 2017], a gradient boosting framework developed by Microsoft, has been designed to emphasise efficiency and speed. In comparison to the level-wise approach of XGBoost, it embraces a leaf-wise growth strategy for constructing decision trees, which prioritises the nodes that yield substantial reduction in loss. This strategy exhibits greater efficiency in terms of memory and computation; however, it necessitates careful control to prevent overfitting. LightGBM incorporates a technique known as Gradient-based One-Side Sampling (GOSS), which diminishes the number of data instances used for calculating gradients during the training process, thereby further enhancing efficiency. Additionally, it inherently supports categorical features, enabling users to input categorical data directly without the requirement for one-hot encoding. LightGBM also encompasses integrated mechanisms for handling missing values and facilitates parallel training. The hyperparameters that will enter the search space for this algorithm are similar to the ones of XGBoost and can be found in the Appendix.

CatBoost

CatBoost [Prokhorenkova et al., 2017], a gradient boosting framework developed by Yandex, has been specifically

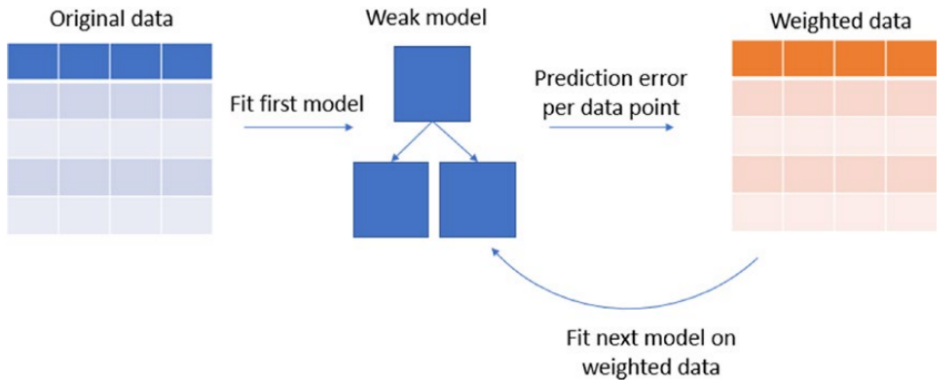


Fig. 1 Gradient boosting process [Korstanje, 2021]

designed to address the effective handling of categorical features. After the introductions of XGBoost and LightGBM, it introduced a novel approach termed Ordered Boosting, which integrates the inherent ordering of categorical features into the boosting procedure. This approach exhibits the potential to enhance model performance when confronted with categorical data. Similar to LightGBM, CatBoost automatically manages categorical features by internally performing one-hot encoding during the training phase, thereby eliminating the necessity for manual preprocessing. It incorporates advanced techniques, including gradient-based pre-sorting, to expedite the training process and minimise memory usage. CatBoost encompasses integrated methodologies for handling missing values and offers robust tools for hyperparameter tuning. CatBoostRegressor, a specific implementation of the CatBoost algorithm, has been tailored for regression tasks and capitalises on the diverse capabilities provided by the CatBoost framework. The hyperparameters that will enter the search space for this algorithm are also similar to the ones of XGBoost and are presented in the Appendix.

Bayesian Optimisation Search

For each gradient boosting algorithm a search space is mentioned. This refers to a set of possible values that can be explored during the process of hyperparameter tuning or optimisation. It represents the entire space of potential values that can be assigned to each hyperparameter in a machine learning model. The process is in fact an optimisation problem: to minimise the validation errors. There are several approaches to do this, the most common is a grid search. A grid search exhaustively explores a predefined set of hyperparameter values by creating a grid or mesh of all possible combinations. It systematically evaluates each combination, covering the entire search space. This is a relatively simple and straightforward approach, however, can be computationally expensive and time-consuming, especially when dealing with a large number of hyperparameters or a large search space. Therefore, a Bayesian search is opted. Bayesian search, specifically Bayesian optimisation, intelligently explores the search space by leveraging probabilistic models and informed decision-making. It iteratively selects hyperparameter settings based on the current knowledge of the objective function, using an acquisition function to balance exploration and exploitation. Bayesian search dynamically adjusts the search based on previous evaluations, focusing on promising areas of the search space. This approach is more efficient in terms of the number of objective function evaluations required to find good hyperparameter settings. It uses an informed search strategy that adapts to the observed results, concentrating on areas likely to yield better performance. Bayesian search tends to be more efficient and effective in finding good hyperparameter settings, especially when the search space is large or complex, or when the objective function evaluations are expensive. For a more exhaustive description, see [Pelikan et al., 1999] and [Dewancker et al., 2017]. The search will utilise a 3-fold cross-validation technique to train the training dataset.

IV.C. Hybrid Forecast Approaches

Next to the features from the dataset that enter the gradient boosting models, another way of improving the final results of the forecast is by making predictions by simpler forecast methods. The reasoning behind this is that if that outcome of such a model can steer the prediction of the gradient boosting model in the right direction by utilising the outcome as another feature. In this paper, three methods are introduced: Previous Flight model, Hierarchical model, and a Time Series model. In figure 2 the structure of the total forecast model is presented if all of these models were to be utilised.

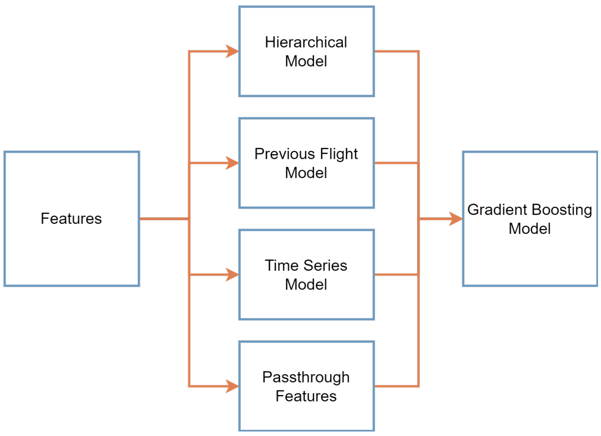


Fig. 2 Forecast model structure

Previous Flight Forecast Model

The previous flight forecast model employs a relatively straightforward approach that requires two inputs: the flight designator and the scheduled date-time. Using historical data, the model identifies the first occurrence of a similar flight designator and utilises this information to predict the BF, see figure 3. The rationale behind this prediction model is based on the observation that flights with the same flight designator often exhibit similarities in terms of destination, departure day and time, and passenger volume. However, by selecting the most recent similar flight, the model disregards potential long-term trends in the data. Moreover, a limitation of this model arises from the possibility of flight designators changing over time or new flight designators being introduced. These situations may result in a comparison with a completely different type of flight or lead to the inability to find a suitable value for prediction.

Fig. 3 Previous Flight model approach example

```
['Airline', 'Country', 'Weekday', 'Scheduled_Time', 'Airport', 'Seats']
['Airline', 'Country', 'Weekday', 'Scheduled_Time', 'Airport'],
['Airline', 'Country', 'Weekday', 'Airport'],
['Airline', 'Country', 'Weekday', 'Scheduled_Time'],
['Airline', 'Country', 'Airport'],
['Airline', 'Country'],
['Airline'],
['Country']
```

Fig. 4 Structure of the hierarchy model

Hierarchical Model Forecast Model

The hierarchical model, as an advancement over the previous flight model, incorporates a more extensive historical perspective. It employs a hierarchical structure, depicted in figure 4, to identify the most comparable flights within the dataset and compute the average BF based on these findings. When fitting and predicting, the model initially seeks flights in the training dataset that possess identical features as those in the topmost row of the hierarchy. If an exact match is not found, it proceeds to search for flights with matching features in the subsequent rows, and so on. The features defined within this structure serve as the sole inputs for this model. A key advantage of this approach, in contrast to the previous flight model, lies in its immunity to variations or alterations in flight numbers. Instead, it seeks flights with comparable characteristics and calculates the average BF based on those flights. One drawback of the hierarchical model is the potential for a relatively large error in the predicted value due to the averaging of multiple flights. This discrepancy can be attributed to several factors. Firstly, the comparable flight selected from the past may belong to a season or time period that does not adequately represent the current date, leading to a mismatch in relevant conditions. Secondly, the model does not consider other features that may have an impact on the prediction, thereby limiting its accuracy. Lastly, as the prediction descends further down the hierarchy, its precision diminishes, resulting in less accurate estimations.

Time Series Forecast Model

Predictions are commonly based on time series data, where the data points are equally spaced in time. However, the flight schedule data used to forecast individual flights does not adhere to this pattern. In this study, a time series model was developed to fit the training data by calculating the average BF per day for each airline and destination airport. A classical MLR model was then employed, using this average BF, to predict the forecast period in days for each airline and destination. The input variables required for this prediction are the airline code, airport code, and scheduled departure date-time. The MLR model incorporates additional exogenous factors, and feature engineering was performed by including week, weekday, and month variables. The test set is combined with the forecasted values from this model to generate predictions, as can be seen in figure 5. One advantage of this model is its relatively high accuracy in forecasting time series data, resulting in reasonably good predictions. Furthermore, by considering the combination of airlines and destinations, which exhibit similar BFs over time, taking the average daily value and projecting it onto future days yields accurate results. However, a limitation of this model is that not all airlines operate daily flights from AMS, and not all destinations are served by every airline. To address this issue, any gaps in the data are filled using the average BF from all days on which flights departed. Only when there are very few data points available is the prediction likely to align closely with the overall average, which may lead to reduced accuracy.

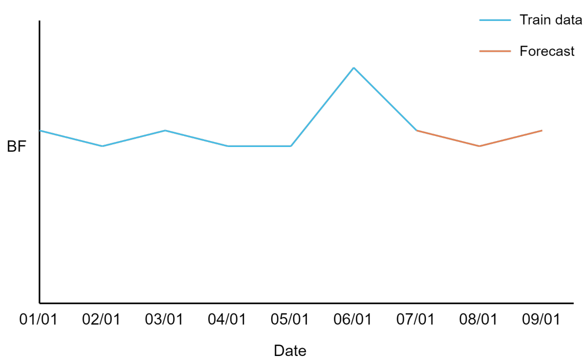


Fig. 5 Time series approach example

IV.D. Feature Selection

Finally, the resulting accuracy might not exhibit optimal performance. Especially if all features were used in the model, where one might have more influence than the other. To tackle this, feature selection approaches can be applied. Feature selection is a process that chooses a subset of features from the original features so that the feature space is optimally reduced according to a certain criterion. Several approaches are used to select features of this problem: the filter method, the wrapper method, and the embedded method. Note that these methods do not always instantly provide the best feature selection; it is an iterative process where more methods can be used to find the best subset(s).

Filter Method

Filter methods are preprocessing techniques used to select features from a dataset without considering specific machine learning algorithms. They efficiently remove duplicated, correlated, and redundant features. However, they do not address multicollinearity issues and can make it challenging to determine the individual effects of predictor variables on the response variable. Feature selection is evaluated individually, which is advantageous for independent features but may lag for feature combinations that improve model performance. Pearson's correlation coefficient can identify subsets of independent features or those strongly correlated with the target variable, helping assess their impact on model accuracy.

Wrapper & Embedded Method

Wrapper methods involve iteratively training an algorithm using a subset of features. Features are added or removed based on insights gained during prior training. The best subset is determined by predefined stopping criteria, such as decreased model performance or reaching a specific number of features. Wrapper methods provide an optimal feature set for higher model accuracy compared to filter methods, but they are computationally expensive. Selection can be done exhaustively, forward (adding features incrementally), or backward (removing features iteratively) until no further improvement is observed. Embedded methods incorporate feature selection algorithms into the learning algorithm itself, making them faster and more accurate. Gradient boosting, for example, provides feature importance to select impactful features. This study combines backward elimination and embedded feature importance for feature selection, aiming to achieve high model accuracy by iteratively evaluating the impact of features on the target variable.

IV.E. Model Quality Metrics

To assess the quality of the model's predictions, it is essential to examine the results in detail. Various error metrics are available to evaluate accuracy, although their interpretation may not always provide deep insights. In addition to understanding the error metrics, it is important to gain a comprehensive understanding of the model's predictions and delve further into their implications. To achieve this, certain model statistics will be explored and analysed.

Error Metrics

In order to compare the predictive performance of the different selected features and model algorithms, the forecast performance is measured by five different metrics: the R^2 score, Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), Median Absolute Error (MdAE), and the Root Mean Squared Error (RMSE). The R^2 score, equation 1, is used to assess the goodness of fit of a regression model in forecasting. It indicates how well the model's predictions explain the variability observed in the actual data. It ranges from 0 to 1, with a higher value indicating a better fit of the model to the data.

$$R^2 = 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y}_i)^2} \tag{1}$$

The MAE, equation 2, provides a straightforward indication of the magnitude of the forecasting errors, which makes it easy to interpret. It is less sensitive to outliers compared to other error metrics.

$$MAE = \frac{1}{n} \sum |y_i - \hat{y}_i| \tag{2}$$

The MAPE, equation 3, is useful to assess the accuracy of the forecast relative to the magnitude of the actual values. The R^2 score and the complement of the MAPE are expected to yield similar values. Thus it is a good check to calculate both errors.

$$MAPE = \frac{1}{n} \sum \left| \frac{y_i - \hat{y}_i}{y_i} \right| \tag{3}$$

The MdAE, equation 4, is similar to MAE but uses the median instead of the mean. It is less influenced by extreme outliers compared to MAE and provides a more robust measure of error.

$$MdAE = \text{median}\,(|y_i - \hat{y}_i|) \tag{4}$$

The RMSE, equation 5, is a widely used error metric that penalises larger errors more than the MAE, as it squares the differences. It is a popular choice for assessing the overall accuracy of a forecasting model.

$$RMSE = \sqrt{\frac{1}{n} \sum (y_i - \hat{y}_i)^2} \tag{5}$$

Model Statistics

To gain deeper insights into the forecast results, more comprehensive statistical measures can be employed. While error metrics such as MAE and MdAE provide valuable information, understanding the nature of large errors and the overall distribution of errors is crucial. Visualisations like boxplots and histograms can be utilised to assess the disparities between predicted and actual values, which can vary in both positive and negative directions. Boxplots offer a graphical summary of the error distribution, with the box representing the middle 50% of the errors. A smaller distance of the box from zero signifies better results. Additionally, boxplots facilitate the identification of distribution skewness and the presence of outliers. However, it can be challenging to discern whether the errors are concentrated around the box or evenly spread within the boxplot's whiskers. To address this, boxplots can be complemented with violin plots or histograms that incorporate the absolute errors. The combination of these visualisations aids in gaining a more comprehensive understanding of the error patterns. Moreover, it is informative to examine a histogram where the bins represent ranges of the actual BFs, and the bars indicate the average forecast error within each range. This histogram is expected to exhibit lower error margins for lower actual values and ideally demonstrate consistently low error margins across all actual values.

In addition to the statistical review using boxplots, violin plots, and histograms, conducting error analysis for different flight features is valuable. Examining the average error per airline, per weekday, per destination, and the conversion of BF to baggage items provides insights into the prediction performance in relation to these specific features. These statistics enable the assessment of prediction results in different contexts and facilitate the identification of areas where the model excels or exhibits relatively larger errors.

IV.F. Mixed Integer Linear Programming Model

The resource allocation or scheduling model is formulated as a MILP model. The input data is one day of the prediction of the forecast model. Furthermore, there are some assumptions that have been made based on interviews with AMS employees. Inspired by the problem description and mathematical formulation of Emde et al. [2020], this paper introduces the problem of scheduling the build-up of Make-Up Areas at a baggage handling facility under Space and Personnel constraints (MUASP). Given a set of MUAs and outbound flights to be handled within a certain time window, where each MUA must be built-up for a given amount of flight batches that contain different numbers of baggage items. The processing time depends on the number of workers that are assigned to the flight and MUAs. The goal is to keep the demand for workers just about level at all times and simultaneously reduce the number of MUAs that need to be used at the peak period throughout a day. The mathematical formulation is set up in such a way that it assumes an ideal situation, where building the MUAs up with the number of required workers is always possible. The sets, parameters, and decision variables can be found in table 1, table 2, and table 3. The mathematical formulation of the MILP model is shown next.

The objective function, equation 6, minimises the number of MUAs in use in the busiest period throughout the day. This objective function is subject to fourteen constraints in total. Constraint (7) in conjunction with constraint (8) and (9) limits the number of MUAs in use to the maximum amount of MUAs present each time period and sets the decision variable α to the maximum value over time. The model will then try to find a solution where this maximum number during a specific time (the peak period) is the lowest. Within this model there is no real changes between the number of MUAs that are available per period. However, considering that an MUA might not be available due to malfunctioning, maintenance or other reasons or intentions, this value might change per time period when making the model more realistic. Constraint (10) enforces that each flight is handled exactly once. Constraints (11) and (12) makes any time window violation impossible, taking into account the process time needed depending on the amount of workers are assigned to the flight. Constraint (13) prevents that the total workspace of all MUAs in use per time period is exceeded. If that is the case an it is inevitable, another MUA must be opened. Within this constraint for each time period the decision variable x will be checked for all periods that are needed to handle the flight batch by setting t' such that $t \geq t' \geq t - P_{fk} + 5$. This ensures that for each time periods all assigned flights, both done within one time period and over multiple time periods are taken into account in the summation. From here onward, the constraints have this t' for the same reasoning. Constraint (14) limits the total number of workers assigned to all in use MUAs to the total worker capacity in a certain time period. Constraint (15) ensures that an MUA can only be in use once per time period. If the MUA in use is a lateral instead of a carousel, next to the fact that only a maximum of five LUs can be assigned per period, that means that there is no possibility to stack up the flights. Thus constraint (16) ensures that per time period only one flight can be assigned to a lateral. Constraint (17) prevents the build-up of more LUs in a time period than the MUA capacity can provide. In a similar way, constraint (18) prevents that the number of baggage items on the MUA in a certain time period exceeds the capacity of the MUA, taken into account that the workers remove baggage items during that time period. It is assumed that for each handled bag, instantly a new one can appear on the MUA. Constraint (19) checks on the left-hand side how many baggage items the assigned number of workers can handle at an MUA. This cannot be exceeded by the number of baggage items that are assigned to the same MUA of multiple flights divided by the number of periods it takes to process the whole batch, which is formulated on the right-hand side of

Table 1 Overview of the sets in the MILP model.

Sets	
F	Set of flights, index $f \in F$
T	Number of periods, index $t \in 0, 5, \ldots, T$. Time interval is 5 as flight departure times are separated by 5 minutes
M	Set of MUAs, index $m \in M$
K	Set of workers, index $k \in K$

Table 2 Overview of parameters in the MILP model.

Parameters	
S_m	The number of workspaces (number of LUs) at MUA m
M_t	Maximum number of MUAs available in period t
S_f	Workspace (number of LUs) required by flight f
B_f	The batch size (number of baggage items) of flight f
A_f	Release datetime of flight f, when the batch is ready to be processed
D_f	Deadline of flight f, which is 30 minutes before departure
P_{fk}	Processing time of the batch of flight f if k workers are assigned
MUA_m	Workspace available at MUA m (number of LUs)
CAP_m	Space available at MUA m (number of baggage items)
P_k	Productivity of k workers; the number of baggage items k workers can process per time period

the constraint equation. Constraint (20) prevents the total number of workers assigned to the flights that are assigned to an MUA in a certain time period to exceed the number of workers present at that MUA. Constraint (21) makes sure that if there are no flights assigned to a specific MUA in a certain time period, the MUA cannot be in use. Lastly, similar to the previous constraint, constraint (22) makes sure that is a flight in a certain time period can only be assigned to a specific MUA, if that MUA is in use.

$$\min \alpha \tag{6}$$

s.t.

$$\sum_{m \in M} \sum_{k \in K} k \cdot y_{mtk} \leq \alpha_t \quad \forall t = 0, \ldots, T \tag{7}$$

$$\alpha_t \leq M_t \quad \forall t = 0, \ldots, T \tag{8}$$

$$\alpha \geq \alpha_t \quad \forall t = 0, \ldots, T \tag{9}$$

$$\sum_{m \in M} \sum_{t=0}^{T} \sum_{k \in K} x_{fmtk} = 1 \quad \forall f \in F \tag{10}$$

$$\sum_{m \in M} \sum_{t=0}^{T} \sum_{k \in K} t \cdot x_{fmtk} \geq A_f \quad \forall f \in F \tag{11}$$

$$\sum_{m \in M} \sum_{t=0}^{T} \sum_{k \in K} \left(t + P_{fk}\right) \cdot x_{fmtk} \leq D_f \quad \forall f \in F \tag{12}$$

$$\sum_{m \in M} \sum_{k \in K} MUA_m \cdot y_{mtk} \geq \sum_{f \in F} \sum_{m \in M} \sum_{k \in K} \sum_{t'=\max\{0; t-P_{fk}+5\}}^{t} S_f \cdot x_{fmt'k} \quad \forall t = 0, \ldots, T \tag{13}$$

$$\sum_{m \in M} \sum_{k \in K} k \cdot y_{mtk} \leq K_t \quad \forall t = 0, \ldots, T \tag{14}$$

Table 3 Overview of the decision variables in the MILP model.

Decision Variables	
x_{fmtk}	Binary variable: 1, if flight f is assigned to MUA m, processed in period t by k workers; 0, otherwise
y_{mtk}	Binary variable: 1, if MUA m is used in period t with k workers to process the flight batches; 0, otherwise
α_t	Continuous variable: number of MUAs used in period t
α	Integer variable: the maximum α_t value in period $t \in 0, 5, \ldots T$

$$\sum_{k \in K} y_{mtk} \leq 1 \quad \forall m \in M \wedge t = 0, \ldots, T \tag{15}$$

$$\text{if} \quad MUA_m = 5 :$$

$$\sum_{f \in F} \sum_{k \in K} \sum_{t'=\max\{0;t-P_{fk}+5\}}^{t} x_{fmt'k} \leq 1 \quad \forall t = 0, \ldots, T \wedge m \in M \tag{16}$$

$$\sum_{f \in F} \sum_{k \in K} \sum_{t'=\max\{0;t-P_{fk}+5\}}^{t} S_f \cdot x_{fmt'k} \leq MUA_m \quad \forall t = 0, \ldots, T \wedge m \in M \tag{17}$$

$$\sum_{f \in F} \sum_{k \in K} \sum_{t'=\max\{0;t-P_{fk}+5\}}^{t} B_f \cdot x_{fmt'k} \leq \sum_{k \in K} CAP_m + P_k \cdot y_{mtk} \quad \forall t = 0, \ldots, T \wedge m \in M \tag{18}$$

$$\sum_{k \in K} P_k \cdot y_{mtk} \geq \sum_{f \in F} \sum_{k \in K} \sum_{t'=\max\{0;t-P_{fk}+5\}}^{t} \lceil B_f / P_{fk} \rceil \cdot x_{fmt'k} \quad \forall t = 0, \ldots, T \wedge m \in M \tag{19}$$

$$\sum_{f \in F} \sum_{k \in K} \sum_{t'=\max\{0;t-P_{fk}+5\}}^{t} k \cdot x_{fmt'k} \leq \sum_{k \in K} k \cdot y_{mtk} \quad \forall t = 0, \ldots, T \wedge m \in M \tag{20}$$

$$\sum_{f \in F} \sum_{k \in K} \sum_{t'=\max\{0;t-P_{fk}+5\}}^{t} x_{fmt'k} \geq \sum_{k \in K} y_{mtk} \quad \forall t = 0, \ldots, T \wedge m \in M \tag{21}$$

$$\sum_{k \in K} \sum_{t'=\max\{0;t-P_{fk}+5\}}^{t} x_{fmt'k} \leq \sum_{k \in K} y_{mtk} \quad \forall t = 0, \ldots, T \wedge m \in M \wedge f \in F \tag{22}$$

Computational Experiments

The computational experiments were conducted using the Python programming language and solved utilising the open-source solver PuLP Coin-or Branch and Cut, version 2.7.0, in conjunction with Python 3.9. The experiments were executed on a Dell Latitude 5430 laptop equipped with an Intel(R) Core(TM) i5 processor and 16 GB of RAM. However, it was observed that running the experiments on this laptop occasionally failed to generate feasible solutions within the designated computational time for larger instances. To overcome this limitation, a Databricks server with enhanced computational power was employed. The server offered 16 cores and 112 GB of RAM. The maximum computational time allocated for all experiments was set at 18,000 seconds, equivalent to four hours.

The computational experiments were focused on a single baggage facility responsible for handling flights associated with a specific airline, particularly European flights. The objective was to determine the minimum number of MUAs required during the busiest period of the day. Several test instances were utilised to evaluate the model's performance. These instances were selected to ensure diversity in three aspects: the number of flights during a day (achieved by choosing different days), the capacity of LU, and the time windows allocated for processing individual flights. The variation in the number of flights is expected to impact both the computation time and the model's objective value, as a larger number of flights generally leads to higher objective values and longer computation times. While LU capacity may not significantly influence computation time, it does affect the model's outcomes, with lower LU capacities resulting in higher objective values. The time window variability primarily affects the model's outcomes, with narrower time windows typically leading to higher objective values. Table 4 provides details of the test instances, including their specific characteristics. The first instance serves as the baseline, while the remaining three instances differ in terms of the aforementioned factors.

Table 4 Names, definitions, and characteristics of the different test instances.

Instance	Day	LU capacity	Time window
MUASP_t_35_n	Tue	35	1H30M
MUASP_t_25_n	Tue	25	1H30M
MUASP_s_35_n	Sat	35	1H30M
MUASP_t_35_l	Tue	35	1H

V. Results

This section shows the results that were obtained by the two models. First the feature selection of the forecast model will be addressed in detail. Secondly, the results of the forecast model will be presented with a deepdive into what these results entail. Finally, the computational set-up of the MILP model will be explained and its results are presented.

V.A. Feature Selection

In table 11, in the Appendix, the correlation matrix can be found which provides information about the relationships among all possible features. It reveals the extent of correlation between each feature and both other features and the target feature. Notably, the flight number, airport, airline, aircraft, number of transfer passengers, and outbound range exhibit the strongest correlations with the target feature. This constitutes the first subset of features as it is worth exploring the impact on the model when including the strongest correlated features.

First and foremost, the flight number stands out in terms of correlation strength compared to other features. This is not surprising, as airlines commonly assign the same flight number to flights with similar characteristics (e.g., destination, time, aircraft). Consequently, the flight number exhibits a strong correlation with numerous other features. Although incorporating this feature into the model seems logical, it carries certain risks. The high correlation with other features may lead to overfitting, resulting in reduced accuracy scores for actual predictions. By solely considering the flight number in conjunction with features that are weakly correlated to it, important predictive aspects could be overlooked. Additionally, airlines may employ distinct approaches when assigning flight numbers, leading to potential changes over time or the introduction of new numbers. Consequently, it is anticipated that accuracy scores would be lower for such a subset. To investigate this, two subsets are constructed, accounting for intercorrelations: 1) an isolation subset containing the flight number while excluding strongly correlated features, and 2) an isolation subset excluding the flight number.

To further delineate the isolated subset that excludes the flight number, an examination of the remaining features was conducted. Within the time category, the year, month, week, weekday, and holiday features displayed negligible correlations with both the other feature types and the target feature. However, they exhibited strong correlations with one another. This suggests that these features contribute minimally to the final prediction and could be omitted. Conversely, the scheduled departure time and date demonstrated relatively strong correlations with the target feature and certain other features. Hence, retaining them in the model during this phase is of interest. The airline feature exhibited strong correlations with a subset of features, primarily the flight number. Nonetheless, it displayed a highly robust correlation with the target feature, as anticipated, given the airline's substantial influence on the number of checked baggage items carried by passengers. Shifting focus to the destination category, the airport, outbound range, and aircraft features exhibited strong correlations with each other, rendering their simultaneous inclusion in the model unnecessary. Specifically, the airport and aircraft features displayed strong correlations with nearly all features, except for the time features. Lastly, the features within the passenger category demonstrated moderate correlations, although not excessively strong. The correlation of these features with the target feature aligns with expectations, as transfer passengers typically transport a checked baggage item, while O&D passengers often do not. Consequently, these features prove relevant for inclusion in the model. Naturally, predicting the exact number of passengers in advance is unfeasible; nevertheless, incorporating the passenger forecast in the model is expected to positively impact the final BF prediction. To summarise the composition of the second isolated subset, based on the aforementioned rationale, certain destination features and most time features were excluded.

Determining the precise impact of each feature on the model's accuracy and discerning whether strongly correlated features are redundant or collectively contribute to a stronger positive impact pose challenges. Consequently, a fourth subset akin to the first subset was generated, omitting the flight number feature. Lastly, the embedded backward elimination method was employed, wherein the model selects features by iteratively eliminating the least significant feature until no further improvement is observed. To summarise, the subsets are as follows:

- Subset 1: all strong correlated features: `[flight number, airport, airline, aircraft, transfer pax, outbound range, country, scheduled time, seats, scheduled date, O&D pax]`
- Subset 2: isolated subset incl. flight number: `[flight number, outbound range, transfer pax, O&D pax, scheduled time, scheduled date, weekday]`
- Subset 3: isolated subset excl. flight number: `[airport, airline, country, seats, scheduled time, scheduled date, transfer pax, O&D pax]`
- Subset 4: subset with many strong correlated features excl. flight number: `[airport, airline, aircraft, transfer pax, country, O&D pax, scheduled time, seats, weekday]`
- Subset 5: Embedded backwards elimination, contains all features and narrows it down for each algorithm.

As mentioned previously, subset 5 comprised all the features except for a few eliminated ones. Through the application of embedded backward elimination tests for XGBoost, only the features "holiday region" and "holiday type" were eliminated. For LightGBM, only the feature "outbound range" was removed, while for CatBoost, both "outbound range" and "holiday region" were eliminated.

All five subsets were subjected to testing, and the results are presented in table 12 in the Appendix. Initial results revealed that optimal forecast accuracy across all three models was achieved when the train set encompassed as much data as possible for each forecast period. Specifically, this entailed restricting the test set size solely to the flights on the days targeted for forecasting, while the train set incorporated the remaining data. Looking at these results for the subsets, it is evident that subset 5 yielded the best results across all three algorithms. This suggests that the correlated features collectively contribute positively to the accuracy. To verify this, additional test rounds were conducted where some features were removed from the subset. However, the results did not improve. Consequently, subsets 5 will be used for the next step, which involves incorporating additional model features.

To accomplish this, an exhaustive search was performed to explore all possible combinations of additional model features for each of the three algorithms. The results of this exhaustive search revealed that incorporating the time series model led to the highest increase in accuracy, ranging from 1-2% for all three gradient boosting algorithms. Although the predictions for 7 days remained similar, the predictions for 30 and particularly 60 days became more accurate with the inclusion of the time series model. The differences in accuracy between using the time series model, the previous flight model, or both were very small. However, in general, incorporating only the time series model generated slightly superior results. Therefore, the time series model is incorporated in the final forecast model. It is worth noting that adding the hierarchical model to the feature combinations resulted in decreased accuracy. Although incorporating the hierarchical model output as a feature yielded a higher train score, indicating slight overfitting, it was decided to exclude this model.

All three algorithms exhibited sufficiently high scores to enable relatively accurate predictions. Among them, Light-

GBM had slightly better accuracy compared to the other two, and it boasted an average computation time of 80 seconds, whereas CatBoost took an average of 108 seconds and XGBoost required 150 seconds. Consequently, LightGBM emerged as the most favourable algorithm and will be further examined in the subsequent phase. Table 5 presents the results of the LightGBM model, including the time series model output as a feature. It can be observed that the train and test scores for all forecast periods were marginally better than the results without the time series model output, as shown in table 12 in the Appendix with subset 5. Furthermore, the train and test scores were highly similar, indicating the absence of overfitting or underfitting. The MAE indicates that, on average, the model's estimation of the BF deviates by 0.09-0.1. The MAPE suggests that the average percentage error between the predicted and actual values is approximately 18-21%, confirming the test score, as this value should be similar to the complement of the MAPE. The MdAE has a lower value than the MAE, which is a positive indication, as it implies that most values are closer to zero error rather than exceeding the MAE values. Similarly to the MAE, the RMSE provides an interpretable measure of error. However, the RMSE is significantly higher than the MAE, indicating that there is greater variability and dispersion in the errors.

V.B. LightGBM Model Forecast Results

To elucidate the underlying basis of the results obtained from the LightGBM model, the feature importance of each variable is depicted in figure 8. Evidently, the week, flight number, airport, and scheduled departure time emerge as the most influential features that underpin the predictions. Conversely, the year feature exhibits significantly lower importance relative to the other variables. While it may be argued that eliminating this feature could be a viable option, such an exclusion leads to slightly diminished prediction accuracy. One plausible explanation for this outcome is the subtle discrepancy in average BF between 2023 and 2022, considering that the 2023 dataset encompasses substantially fewer flights than the 2022 dataset. Consequently, in future forecasts with augmented information, it is plausible that the year feature might be eliminated from the feature subset.

Upon delving deeper into the results of this model, a boxviolin plot (figure 6) and a histogram (figure 7) were utilised to gain further insights. The boxplot illustrates that the interquartile range, representing the middle 50% of errors, falls within the range of -0.06 to 0.08. The violin plot and histogram reveal that a significant portion of the data, beyond the middle 50%, concentrates near the edges of the box, while relatively fewer data points are observed around the whiskers. Outliers are present on both the upper and lower ends of the distribution. Specifically, for the 7-day forecast, there are 130 outliers out of a total 3877 flights, accounting for approximately 3% of the forecasted data. Furthermore, the majority of predictions exhibit absolute errors in the range of 0.0 to 0.04, indicating a high level of accuracy and providing an accurate depiction of the expected BFs within the forecasted time period.

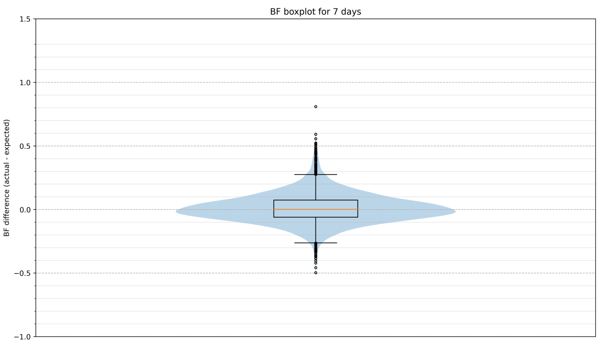


Fig. 6 Box-Violinplot of the model errors

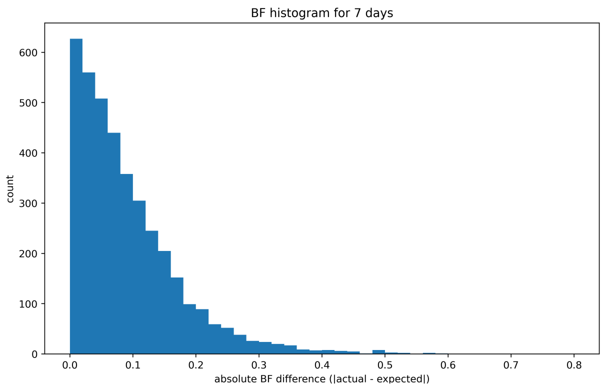


Fig. 7 Histogram of absolute model errors

To identify the regions where the majority of errors occur, figure 9 presents an overview of the average error relative to the actual BF, with the number of data points falling within each specific range displayed atop the bars. It is evident that

Table 5 LightGBM final results for the different forecast periods including time series model output as a feature.

	Train score	Test score	MAE	MAPE	MdAE	RMSE
7 days	0.833	0.833	0.089	0.183	0.068	0.120
30 days	0.846	0.819	0.092	0.189	0.071	0.123
60 days	0.808	0.792	0.101	0.214	0.079	0.134

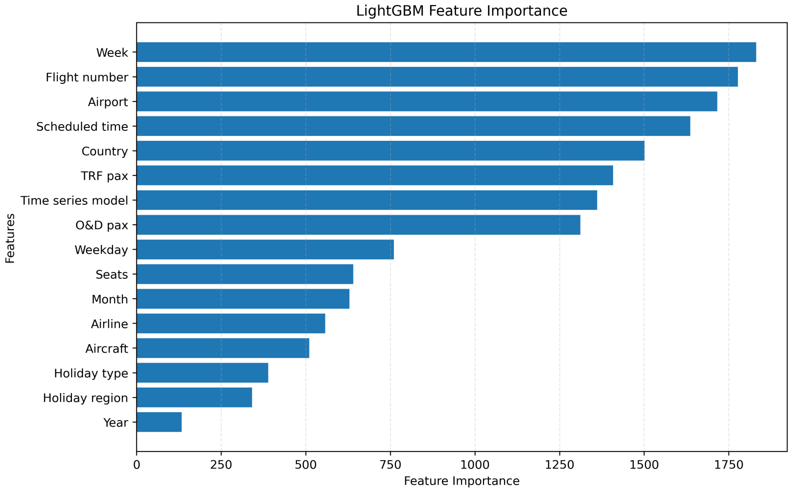


Fig. 8 Feature importance LightGBM

the bins with the highest concentration of data points generally exhibit the lowest average error, approximately 0.1. As the actual BF increases, the error also tends to increase. However, this trend is partly attributed to the decreasing number of data points available as the BF rises, leading to a reduced basis for prediction. In addition to examining BF errors, it is also valuable to consider the distribution of these factors in relation to the number of passengers and the actual count of checked baggage items. Figure 10 in the Appendix presents a similar relative representation as for the BF, but focusing on the quantity of baggage items. The figure illustrates that, for lower baggage item counts, the average error hovers around 10-20 baggage items per flight. As the actual baggage item count increases, the average error also rises. This phenomenon, akin to the BF, can be attributed to the reduced number of data points available for higher values, resulting in less accurate predictions. Moreover, since the quantities are higher, a BF error of 0.1 leads to a greater error in baggage item count compared to lower values.

Table 6 Average BF error per weekday for LightGBM prediction.

	Average Absolute BF Error		
Weekday	7-day	30-day	60-day
Monday	0.082	0.095	0.103
Tuesday	0.103	0.098	0.105
Wednesday	0.106	0.096	0.105
Thursday	0.086	0.091	0.100
Friday	0.086	0.093	0.100
Saturday	0.100	0.101	0.105
Sunday	0.091	0.097	0.108

Moreover, to dive deeper into the results, in table 6 the average BF error can be found for each weekday. It can be seen that the error is consistent for each day and thus the results respond well to different days in the week. This is of significant importance as the BF typically demonstrates elevated values on Saturdays in contrast to the remaining weekdays. Consequently, this outcome serves as evidence that the forecast model effectively captures the temporal

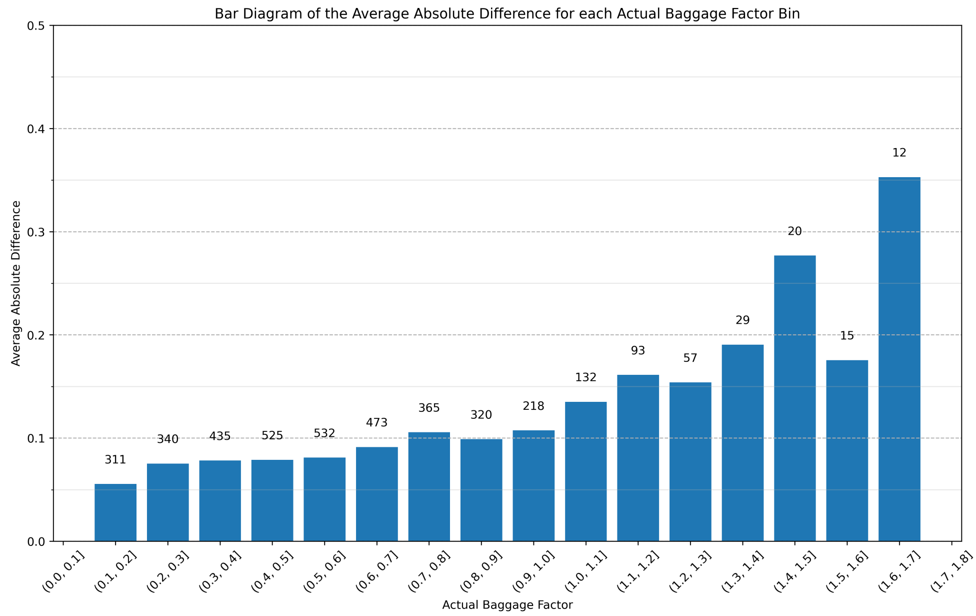


Fig. 9 Average BF error relative to the actual BF

pattern of BF throughout the course of the week.

Table 7 Average BF error per weekday for LightGBM prediction.

	Average Absolute BF Error		
Time Period	7-day	30-day	60-day
00:00-04:00	-	-	-
04:00-08:00	0.094	0.107	0.114
08:00-12:00	0.096	0.102	0.107
12:00-16:00	0.097	0.094	0.102
16:00-20:00	0.078	0.083	0.092
20:00-24:00	0.079	0.085	0.105

In addition, this is also checked for the scheduled departure time and outbound range, which can be found in table 7 and table 8, respectively. In the analysis of scheduled departure times and outbound ranges, no significant outliers were observed. However, it is evident that as the day progresses, the accuracy of the results improves for the scheduled time. This can be attributed to the presence of numerous similar flights departing to the same destination during later hours. Consequently, the model finds it easier to identify patterns within these time periods, leading to enhanced predictive outcomes. Additionally, a similar approach was employed for airlines and destination airports where certain outliers were identified. Among the 70 airlines analysed, four airlines exhibited an average error exceeding 0.15. Similarly, out of the 220 airports examined, 24 airports displayed an average error surpassing 0.15, and among them, eight airports had an error exceeding 0.20. Notably, all these outliers were characterised by their infrequent occurrence, accounting for only a few instances among all the flights examined.

Table 8 Average BF error per outbound range for LightGBM prediction.

	Average Absolute BF Error		
Outbound range	7-day	30-day	60-day
Europe	0.090	0.094	0.102
Intercontinental	0.096	0.101	0.112

Finally, it is of interest to assess the outcomes at a broader level by examining the total volume of forecasted departing baggage items from AMS. These findings are presented in 9. Notably, the error margins for all forecast periods exhibit exceptionally low values, albeit with a slightly higher error observed in the 60-day forecast, which aligns with expectations. Intriguingly, the 7-day and 30-day forecasts demonstrate a slight underestimation, whereas the 60-day forecast displays a slight overestimation. This divergence in predictions may arise from the nature of the currently utilised data, implying that alternative data sources could yield different outcomes. Nevertheless, it is evident that the collective overestimations and underestimations pertaining to individual flights offset each other, culminating in an exceptionally accurate prediction at a higher level.

Table 9 Average BF error per outbound range for LightGBM prediction.

Total bax	7-day	30-day	60-day
Actual	359,967	1,375,243	2,565,020
Predicted	354,669	1,354,961	2,610,156
Difference	-5,298	-20,282	45,136
Error %	-1.47%	-1.47%	1.76%

V.C. Resource Allocation & Scheduling Results

This subsection presents the findings pertaining to the MUASP alongside operational perspectives to provide valuable insights.

Computational Results

Table 10 presents the computational results obtained from the conducted experiments, showcasing the outcomes for different instances. Notably, the computation time for all four instances was relatively high, as anticipated since the MILP model aimed to find an exact solution. The computational time required for each instance to complete a full day's workload was excessively long, necessitating the division of days into multiple shifts. The shifts were designated as follows: 00:00-09:00, 09:00-13:00, 13:00-16:00, and 16:00-24:00. The initial and final shifts were chosen to encompass the majority of the daytime hours, as they typically experience lower flight volumes and consequently handle fewer baggage items. However, an inherent limitation of this approach is the failure to account for potential overlap between shifts. To address this concern, a compensatory adjustment was made by adding a value of 1 to the output result of each shift. Among the instances, the baseline instance demonstrated the shortest computation time compared to the others, while the third instance exhibited significantly longer computation time. The second and fourth instances displayed relatively similar computation times, although slightly higher than the third instance, as anticipated. Taking into account the compensation for the overlap, the baseline instance exhibited an objective value of 3, indicating that during the busiest period(s) of the day a maximum of three MUAs were required. In the second and third instances, the objective values were both 5. This outcome aligns with expectations since in the second instance, the LU capacity experienced a significant reduction. Consequently, the capacity per time period at an MUA decreased, rendering the baseline three MUAs insufficient for the busiest period of the day. Lastly, the fourth instance yielded a higher objective value compared to the baseline instance due to flight batches having a shorter processing period, thereby limiting the model's ability to find a solution with the lowest possible objective value.

Table 10 Results computational experiments for the MILP model with test instances.

Instance	# MUAs	Computation time [s]
MUASP_t_35_n	3	10,483
MUASP_t_25_n	5	12,296
MUASP_s_35_n	5	16,129
MUASP_t_35_l	4	11,914

Operational Insights

From an operational standpoint, ensuring sufficient capacity in baggage handling facilities is crucial to avoid bottlenecks and maintain smooth operations, particularly during peak seasons. Insufficient capacity to handle the incoming demand can severely impede the overall operation. Therefore, the insights gained from this study are highly valuable, as they reveal that in an ideal scenario, only a limited number of MUAs are required to handle the workload effectively. However, it is important to acknowledge that the model has certain limitations due to the assumptions made. For instance, it is not always feasible to direct all baggage items of a flight to a single MUA. Various sub-sorting requirements exist, such as segregating transfer and O&D baggage into different LUs, allocating economy and priority/business class baggage to separate LUs, and accommodating odd-sized bags in different locations. Furthermore, baggage items do not arrive simultaneously in the system, leading to potential challenges in processing an entire flight within a single time period or requiring division across multiple time periods or MUAs. These examples illustrate why the model is not yet ready for implementation. Nevertheless, the predictive model's outcomes provide intriguing insights that can be leveraged to optimise baggage handling facilities further.

VI. Conclusion & Recommendations

This paper presents an approach to enhance baggage handling operations at hub airports through sequential forecasting and resource allocation. The proposed method concerns an exact MILP model whilst utilising the data from a forecast made by a gradient boosting method. The objective of the study was to create a forecast model that predicts BFs for individual flights over a time span of 7, 30, and 60 days, and subsequently construct a baseline model that leverages the forecasting outputs to optimise baggage handling processes and allocate resources effectively.

When comparing the different gradient boosting forecast models, XGBoost, LightGBM, and CatBoost, it is clear that the LightGBM model produces the best results with the selected subset of features where the prediction has an accuracy of up to 83%. Nevertheless, the alternative two models also yield satisfactory outcomes, with discrepancies of merely 1-2% from the desired or expected results. When including features in the model, implementation of intercorrelated features contribute to a positive effect on the performance compared to only implementing independent features. Furthermore, the incorporation of additional feature models yields intriguing insights into enhancing the forecast model's performance. The utilisation of the time series model output as a feature demonstrates the most substantial improvement in prediction accuracy. While the observed performance enhancement may not be significant, the inclusion of the aforementioned feature ultimately yields a superior prediction outcome compared to its exclusion.

Upon evaluating the alignment between the forecasted and predicted quantities of checked baggage items, the proposed forecast model exhibits remarkable accuracy. A substantial proportion of individual flights demonstrate an error margin of approximately 10 checked baggage items in relation to the actual count of checked baggage items present on board. It is noteworthy that this error can be both overestimated or underestimated, contributing to the cancellation of such discrepancies at a higher level, ultimately presenting a representative perspective on the overall anticipated total number of checked baggage items. The resultant errors amount to a mere 1.5% deviation for the 7 and 30 day predictions, and a 1.8% deviation for the 60 day projections when compared against the actual figures.

The main limitations of the forecast model are the complexity and randomness of the problem, and the fact that the data that was used still has some errors. As the BF has to do with how passengers behave and external factors, it is impossible to reach a prediction accuracy of 100%. However, with data that is correct and clean of errors where for each flight each feature has the correct values, it is believed that even better predictions can be made.

The resource allocation and personnel scheduling model provides evidence of the advantages of integrating the BF prediction in examining the MUASP problem from an ideal standpoint. This approach yields a significant reduction in the spatial requirements for baggage handling. The model illustrates that on low-demand days such as Tuesday, or during high-demand periods like Saturday, a minimal number of MUAs were required, specifically 3 and 5 MUAs respectively. This output is contingent on the actual workforce's ability to manage the projected demands effectively. However, real-life scenarios deviate from this ideal situation. Practical cases entail various additional factors that must be taken into account, including the segregation of economy, priority, and business class baggage, the handling of odd-sized baggage items, the occurrence of last-minute baggage arrivals, the variability in employee productivity, potential system malfunctions, and other related considerations. To implement a model like this successfully, it is essential to address these real-life complexities and integrate the assumptions made in this study into the model. Moreover, the current model is computationally demanding due to its NP-hard nature. To make the model applicable for real-world operations, it becomes imperative to incorporate either a heuristic-based approach for assignment selection or utilise an alternative solver.

In future work, there are a number of things that can be looked at and improved. First of all, as is often the case with forecasting models, it is crucial to emphasise that utilising a dataset of superior quality would yield more precise predictions. Additionally, employing such a dataset has the potential to enable separate prediction of the BF for transfer and O&D flights. This segregation of predictions could enhance the final forecast further, presenting an intriguing prospect for integration into the resource allocation model. Notably, this implementation facilitates the feasibility of subsorting, a valuable feature in optimising baggage handling operations. Secondly, it is interesting to consider more subjective features from passenger factors like purpose of the trip, general income level of passengers that fly with certain airlines, character traits, and average trip duration. These features are harder to interpret, but might make a positive impact on the forecast model. Moreover, incorporating data regarding the checked baggage fees specific to each airline and flight can provide valuable insights for predicting passengers' likelihood of availing checked baggage services.

When it comes to the additional feature models, there is potential room for improvement for the hierarchical model and the time series model. The current hierarchical model is constructed to generate predictions by considering the average BF of all comparable flights, which encompasses flights from previous time periods, including flights that departed longer ago. To enhance the accuracy of BF forecasts, implementing an exponential moving average technique that assigns greater weightage to more recent data would likely be beneficial. Regarding the time series model, a potential enhancement could involve developing a methodology to address missing data gaps by either imputing the missing values or exploring techniques to mitigate the impact of these gaps, subsequently evaluating if such measures improve the model's performance. Lastly, for the forecast model, which is primarily driven by managerial and operational considerations, it is imperative to recognise that the forecast itself should not be perceived as an infallible source of information. As previously indicated, forecasts are inherently subject to a degree of uncertainty and are not entirely precise. Therefore, it is crucial to incorporate a secondary verification step, such as involving a market analyst, to provide an additional review and ensure the accuracy and reliability of the forecasted information.

Finally, in addition to the current method employed for leveraging BF predictions, there exist alternative avenues for exploration. For instance, initiating the utilisation of BF at the initial stage of baggage handling, namely at the check-in phase, allows for more precise forecasting of passenger show-up profiles, thereby enhancing the accuracy of predicting baggage item arrival times within the BHS. Furthermore, optimising resource allocation and personnel scheduling within various airport operations involved in baggage handling, such as check-in, security, and other relevant areas, can be achieved by leveraging the insights derived from BF predictions. This optimisation potential encompasses streamlining operations, improving efficiency, and enhancing overall operational performance.

References

Royal Schiphol Group, "Schiphol | Traffic and transport figures per month," https://www.schiphol.nl/en/schiphol-group/page/transport-and-traffic-statistics/, 2019. (Accessed on 25/10/2022).

le Clerq, P., "Jacht op nieuw personeel voor Schiphol verloopt nog moeizaam | RTL Nieuws," https://www.rtlnieuws.nl/economie/bedrijven/artikel/5319477/schiphol-banenmarkt-personeelstekort-beveiligers-bagage, 7 2022. (Accessed on 01/20/2023).

NOS, "Chaos op Schiphol door KLM-staking: reizigers opgeroepen niet meer te komen," https://nos.nl/artikel/2426169-chaos-op-schiphol-door-klm-staking-reizigers-opgeroepen-niet-meer-te-komen, 4 2022. (Accessed on 01/27/2023).

Xu, S., Chan, H. K., and Zhang, T., "Forecasting the demand of the aviation industry using hybrid time series SARIMA-SVR approach," *Transportation Research Part E: Logistics and Transportation Review*, Vol. 122, 2019, pp. 169–180. https://doi.org/10.1016/j.tre.2018.12.005.

Johnston, K., "Hate airline baggage fees? Needham man is to blame - The Boston Globe," https://www.bostonglobe.com/business/2013/04/27/the-man-behind-bag-fees/JHLvuMZhXLOgBZgGzWKCbJ/story.html, 2013. (Accessed on 01/25/2023).

Seaney, R., "History of Airline Fees: Bags, Food & More | FareCompare," https://www.farecompare.com/travel-advice/airline-fees-bags-history/, 2017. (Accessed on 01/25/2023).

Baldanza, B., "Once Derided, Airline Fees For Large Carry-On Bags Now Becoming Mainstream," https://www.forbes.com/sites/benbaldanza/2020/12/07/once-derided-airline-fees-for-large-carry-on-bags-now-mainstream/?sh=343d400476a6, 2020. (Accessed on 01/25/2023).

BBT, "Air France-KLM baggage fee rises on European flights | Business Travel News Europe," https://www.businesstravelnewseurope.com/Air-Travel/Air-France-KLM-baggage-fee-rises-on-European-flights, 2016. (Accessed on 01/25/2023).

Ma, Q., Bi, J., Sai, Q., and Li, Z., "Research on Prediction of Checked-baggage Departed from Airport Terminal Based on Time Series Analysis," Institute of Electrical and Electronics Engineers Inc., 2021, pp. 264–269. https://doi.org/10.1109/ICNISC54316.2021.00055.

Royal Schiphol Group, "Baggage at Schiphol," https://assets.ctfassets.net/biom0eqyyi6b/1LFrmaEaaI2Kgsw0G8IYgC/48c4c91d2b45712d1bb32b4501ad34a8/Baggage_at_Schiphol.pdf, 2018. (Accessed on 28/10/2022).

Chambers, J. C., Mullick, S. K., and Smith, D. D., "How to Choose the Right Forecasting Technique," *Harvard Business Review*, 1971. URL https://hbr.org/1971/07/how-to-choose-the-right-forecasting-technique.

Cheng, S., Gao, Q., and Zhang, Y., "Comparative Study on Forecasting Method of Departure Flight Baggage Demand," 2014.

Chen, C.-J., Yang, S.-M., and Wang, Z.-C., "Development of a Neuro-Fuzzy Model for Airline Passenger Forecasting *," *Astronautics and Aviation. Series A*, Vol. 44, 2012, pp. 169–176. https://doi.org/10.6125/JoAAA.201209_44(3).04.

Rusyana, A., Hasan, A., Oktaviana, M., Statistika, P. S., and Matematika, J., *Forecasting Passenger by Using Holt's Exponential Smoothing and Winter's Exponential Smoothing*, 2016. URL https://www.researchgate.net/publication/316349640.

Box, G., and Jenkins, G., *Time series analysis, forecasting and control*, San Francisco, Holden Day, 1976.

Tsui, W. H. K., Balli, H. O., Gilbey, A., and Gow, H., "Forecasting of Hong kong airport's passenger throughput," *Tourism Management*, 2014. https://doi.org/10.1016/j.tourman.2013.10.008.

Tsui, W. H. K., and Balli, F., "International arrivals forecasting for Australian airports and the impact of tourism marketing expenditure," *Tourism Economics*, Vol. 23, 2017, pp. 403–428. https://doi.org/10.5367/te.2015.0507.

Srisaeng, P., Baxter, G. S., and Wild, G., "Forecasting demand for low cost carriers in Australia using an artificial neural network approach," *Aviation*, Vol. 19, 2015, pp. 90–103. https://doi.org/10.3846/16487788.2015.1054157.

Yang, H. T., and Liu, X., "Predictive simulation of airline passenger volume based on three models," Springer Verlag, 2018, pp. 350–358. https://doi.org/10.1007/978-981-13-2206-8_29.

Manna, S., Biswas, S., Kundu, R., Rakshit, S., Gupta, P., and Barman, S., "A Statistical Approach to Predict Flight Delay Using Gradient Boosted Decision Tree," IEEE, 2017.

Choi, S., and Kim, Y. J., "Artificial neural network models for airport capacity prediction," *Journal of Air Transport Management*, Vol. 97, 2021. https://doi.org/10.1016/j.jairtraman.2021.102146.

Tu, J. V., "Advantages and Disadvantages of Using Artificial Neural Networks versus Logistic Regression for Predicting Medical Outcomes," 1996, pp. 1225–1231.

Emde, S., Abedinnia, H., Lange, A., and Glock, C. H., "Scheduling personnel for the build-up of unit load devices at an air cargo terminal with limited space," *OR Spectrum*, Vol. 42, 2020, pp. 397–426. https://doi.org/10.1007/s00291-020-00580-2.

Kiermaier, F., "Models and Methods for Optimizing Baggage and Ground Handling at Airports," Tech. rep., Technische Universität München, 2015.

Bentéjac, C., Csörgö, A., and noz, G. M.-M., "A comparative analysis of gradient boosting algorithms," *Artificial Intelligence Review*, Vol. 54, 2021, pp. 1937–1967. https://doi.org/10.1007/s10462-020-09896-5.

Hastie, T., Friedman, J., and Tibshirani, R., "The Elements of Statistical Learning," Springer, 2001, pp. 299–345.

Korstanje, J., *Advanced Forecasting with Python*, Apress, 2021. https://doi.org/10.1007/978-1-4842-7150-6.

Chen, T., and Guestrin, C., "XGBoost: A Scalable Tree Boosting System," 2016. https://doi.org/10.1145/2939672.2939785.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T.-Y., "LightGBM: A Highly Efficient Gradient Boosting Decision Tree," 2017.

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., and Gulin, A., "CatBoost: unbiased boosting with categorical features," 2017. URL http://arxiv.org/abs/1706.09516.

Pelikan, M., Goldberg, D. E., and Cantú-Paz, E., “BOA: The Bayesian Optimization Algorithm,” Tech. rep., University of Illinois at Urbana-Champaign, 1999.

Dewancker, I., McCourt, M., and Clark, S., “Bayesian Optimization Primer,” , 2017.

VII. Appendix

Table 11 Correlation matrix of all features.

	Flight number	Airline	Scheduled date	Scheduled time	Holiday type	Holiday region	Year	Month	Week	Weekday	Airport	Outbound range	Country	Aircraft	Seats	TRF pax	O&D pax	**BF**
Flight number	1.00	0.77	0.05	0.43	0.01	0.00	-0.02	0.03	0.03	0.04	0.85	0.63	0.75	0.75	0.43	0.64	-0.16	**0.84**
Airline	0.77	1.00	0.05	0.33	0.00	0.00	-0.01	0.03	0.03	0.03	0.57	0.40	0.45	0.65	0.17	0.47	-0.28	**0.65**
Scheduled date	0.05	0.05	1.00	0.06	0.37	0.14	0.08	0.59	0.66	0.48	0.03	0.01	0.02	0.03	0.01	0.02	-0.12	**0.22**
Scheduled time	0.43	0.33	0.06	1.00	0.03	-0.01	-0.04	0.05	0.05	0.03	0.34	0.29	0.31	0.33	0.22	0.33	-0.07	**0.38**
Holiday type	0.01	0.00	0.37	0.03	1.00	0.31	-0.20	0.48	0.54	0.04	0.01	0.00	0.00	0.00	0.00	0.00	-0.05	**0.08**
Holiday region	0.00	0.00	0.14	-0.01	0.31	1.00	0.08	0.15	0.19	0.04	0.00	0.00	0.00	0.00	0.00	-0.01	0.01	**0.03**
Year	-0.02	-0.01	0.08	-0.04	-0.20	0.08	1.00	-0.45	-0.39	0.01	-0.02	-0.01	-0.02	-0.01	-0.01	-0.02	0.02	**0.02**
Month	0.03	0.03	0.59	0.05	0.48	0.15	-0.45	1.00	0.90	0.00	0.02	0.00	0.01	0.02	0.00	0.00	-0.11	**0.13**
Week	0.03	0.03	0.66	0.05	0.54	0.19	-0.39	0.90	1.00	0.00	0.02	0.00	0.01	0.02	0.00	0.01	-0.10	**0.14**
Weekday	0.04	0.03	0.48	0.03	0.04	0.04	0.01	0.00	0.00	1.00	0.02	0.02	0.02	0.03	0.02	0.04	-0.01	**0.11**
Airport	0.85	0.57	0.03	0.34	0.01	0.00	-0.02	0.02	0.02	0.02	1.00	0.74	0.88	0.71	0.53	0.56	0.00	**0.72**
Outbound range	0.63	0.40	0.01	0.29	0.00	0.00	-0.01	0.00	0.00	0.02	0.74	1.00	0.84	0.67	0.74	0.45	0.33	**0.53**
Country	0.75	0.45	0.02	0.31	0.00	0.00	-0.02	0.01	0.01	0.02	0.88	0.84	1.00	0.70	0.64	0.55	0.13	**0.63**
Aircraft	0.75	0.65	0.03	0.33	0.00	0.00	-0.01	0.02	0.02	0.03	0.71	0.67	0.70	1.00	0.57	0.64	-0.01	**0.63**
Seats	0.43	0.17	0.01	0.22	0.00	0.00	-0.01	0.00	0.00	0.02	0.53	0.74	0.64	0.57	1.00	0.48	0.54	**0.35**
TRF pax	0.64	0.47	0.02	0.33	0.00	-0.01	-0.02	0.00	0.01	0.04	0.56	0.45	0.55	0.64	0.48	1.00	-0.33	**0.56**
O&D pax	-0.16	-0.28	-0.12	-0.07	-0.05	0.01	0.02	-0.11	-0.10	-0.01	0.00	0.33	0.13	-0.01	0.54	-0.33	1.00	**-0.19**
BF	**0.84**	**0.65**	**0.22**	**0.38**	**0.08**	**0.03**	**0.02**	**0.13**	**0.14**	**0.11**	**0.72**	**0.53**	**0.63**	**0.63**	**0.35**	**0.56**	**-0.19**	**1.00**

Hyperparameters XGBoost

The hyperparameters that enter the search space for XGBoost are:

- `max_depth`: The maximum depth of a tree, increasing this value will make the model more complex and prone to overfitting.
- `learning_rate`: Determines the step size at each boosting iteration.
- `subsample`: The subsample ratio of the training instances, which randomly samples the training data prior to growing trees. It is used to minimise overfitting.
- `colsample_bytree`: Determines the fraction of columns (features) to be randomly sampled or subsampled when constructing each individual tree. It ranges between 0 and 1, where 1 represents the use of all features and 0 represents none. A value less than 1 enables random column subsampling, introducing additional randomness to prevent overfitting and improve generalisation.
- `colsample_bylevel`: Specifies the fraction of columns to be subsampled at each level of a tree. It controls the feature subsampling within a level (depth) of the tree, which can provide regularisation by reducing the correlation between trees. This parameter also ranges between 0 and 1.
- `colsample_bynode`: determines the fraction of columns to be randomly sampled when splitting each node of a tree. It applies only to the current node being split and not to the entire level or tree. Similar to the previous two parameters, it ranges between 0 and 1.
- `reg_alpha`: This parameter controls L1 regularisation, also known as Lasso regularisation. L1 regularisation adds a penalty term to the loss function that encourages the model to reduce the absolute magnitude of the weights. It helps to drive less important features' weights towards zero, effectively performing feature selection by shrinking the less informative features. A higher value increases the strength of L1 regularisation, resulting in more aggressive feature selection. This parameter helps preventing overfitting, improve the model's generalisation performance, and enhance its ability to handle noisy or correlated features.
- `reg_lambda`: Similar to the previous parameter, this parameter controls L2 regularisation, also referred to as Ridge regularisation. The difference is that the penalty given to the loss function reduces the squared magnitude of the

weights. Also this parameter helps control the overall complexity of the model and thus prevention of overfitting.

- gamma: Minimum loss reduction required to make a further partition on a leaf node of the tree. The larger the value, the more conservative the algorithm will be.

Hyperparameters LightGBM

The hyperparameters that enter the search space of the LightGBM model are similar to the ones in XGBoost, only colsample_bylevel is not part of the LightGBM parameters. New parameters are num_leaves and max_bin.

- max_depth
- learning_rate
- subsample
- colsample_bytree
- reg_alpha
- reg_lambda
- num_leaves: Specifies the maximum number of leaves (terminal nodes) that can be present in a tree. It controls the complexity and depth of each tree in the ensemble. A higher value allows the tree to capture more fine-grained patterns but may also lead to overfitting. Conversely, a lower value constrains the tree to have fewer leaves, which can help prevent overfitting but may result in underfitting or insufficient model capacity to capture complex relationships in the data.
- max_bin: Determines the maximum number of bins used for discretizing continuous features in the data. Binning is a process of dividing continuous values into discrete intervals or bins, which helps in handling numerical data and reducing the memory footprint.

Hyperparameters CatBoost

In CatBoost, regularisation is primarily achieved through L2 and does not have a dedicated L1 regularisation parameter. Also max_depth is different from the other two techniques. Furthermore, the parameters colsample_bytree, colsample_bynode, and gamma are not present in the search space of CatBoost. An addition to the search space are iterations and bagging_temperature.

- max_depth: the same definition as XGBoost and LightGBM, however, instead of an infinite maximum the maximum depth is only 10.
- learning_rate
- subsample
- colsample_bytree
- iterations: each boosting iteration adds a new decision tree to the ensemble, whereas in other gradient boosting algorithms, these decision trees are often referred to as "trees" or "estimators". Therefore, the iterations hyperparameter determines the number of decision trees to be trained in the CatBoostRegressor model.
- bagging_temperature: Determines the temperature value used in the Softmax function applied during the bagging process. It ranges from 0 to positive infinity. A higher value increases the probability of selecting the most confident predictions during bagging, effectively reducing the randomness and making the algorithm more deterministic. Conversely, a lower value introduces more randomness, leading to a broader range of predictions being selected during bagging.

Table 12 Feature subset results for all three algorithms and forecast time windows.

	XGBoost						LightGBM						CatBoost					
	Train score	**Test score**	**MAE**	**MAPE**	**MdAE**	**RMSE**	**Train score**	**Test score**	**MAE**	**MAPE**	**MdAE**	**RMSE**	**Train score**	**Test score**	**MAE**	**MAPE**	**MdAE**	**RMSE**
7 days																		
S1	0.776	0.770	0.110	0.256	0.091	0.141	0.814	0.794	0.104	0.233	0.084	0.134	0.795	0.788	0.106	0.241	0.087	0.136
S2	0.751	0.754	0.115	0.270	0.095	0.146	0.778	0.773	0.109	0.256	0.090	0.140	0.782	0.774	0.109	0.254	0.089	0.140
S3	0.758	0.744	0.116	0.276	0.097	0.149	0.806	0.791	0.105	0.239	0.086	0.135	0.780	0.781	0.107	0.248	0.087	0.138
S4	0.747	0.772	0.109	0.252	0.090	0.140	0.793	0.814	0.097	0.213	0.076	0.127	0.768	0.797	0.102	0.230	0.083	0.133
S5	0.786	0.811	0.097	0.209	0.077	0.128	0.820	0.823	0.091	0.189	0.071	0.122	0.803	0.817	0.094	0.196	0.073	0.126
30 days																		
S1	0.774	0.750	0.113	0.260	0.093	0.145	0.793	0.766	0.109	0.249	0.089	0.140	0.789	0.763	0.110	0.249	0.091	0.141
S2	0.749	0.728	0.118	0.275	0.097	0.151	0.777	0.749	0.113	0.263	0.093	0.145	0.764	0.736	0.115	0.268	0.094	0.149
S3	0.759	0.729	0.118	0.274	0.097	0.151	0.776	0.751	0.113	0.262	0.093	0.144	0.780	0.765	0.109	0.248	0.089	0.140
S4	0.745	0.754	0.111	0.257	0.090	0.144	0.765	0.778	0.105	0.241	0.086	0.136	0.766	0.780	0.105	0.236	0.084	0.136
S5	0.785	0.795	0.100	0.215	0.079	0.131	0.820	0.807	0.095	0.200	0.074	0.127	0.803	0.804	0.097	0.205	0.076	0.128
60 days																		
S1	0.779	0.743	0.116	0.268	0.094	0.149	0.792	0.757	0.112	0.258	0.093	0.145	0.792	0.754	0.113	0.258	0.094	0.146
S2	0.769	0.731	0.118	0.276	0.097	0.152	0.776	0.740	0.116	0.272	0.095	0.150	0.780	0.737	0.117	0.273	0.096	0.151
S3	0.759	0.720	0.121	0.285	0.100	0.155	0.772	0.743	0.116	0.271	0.095	0.149	0.778	0.748	0.114	0.265	0.094	0.147
S4	0.744	0.746	0.114	0.266	0.093	0.148	0.792	0.786	0.104	0.234	0.084	0.136	0.764	0.768	0.109	0.248	0.088	0.142
S5	0.779	0.771	0.107	0.234	0.085	0.140	0.789	0.776	0.106	0.237	0.085	0.139	0.784	0.776	0.106	0.234	0.085	0.139

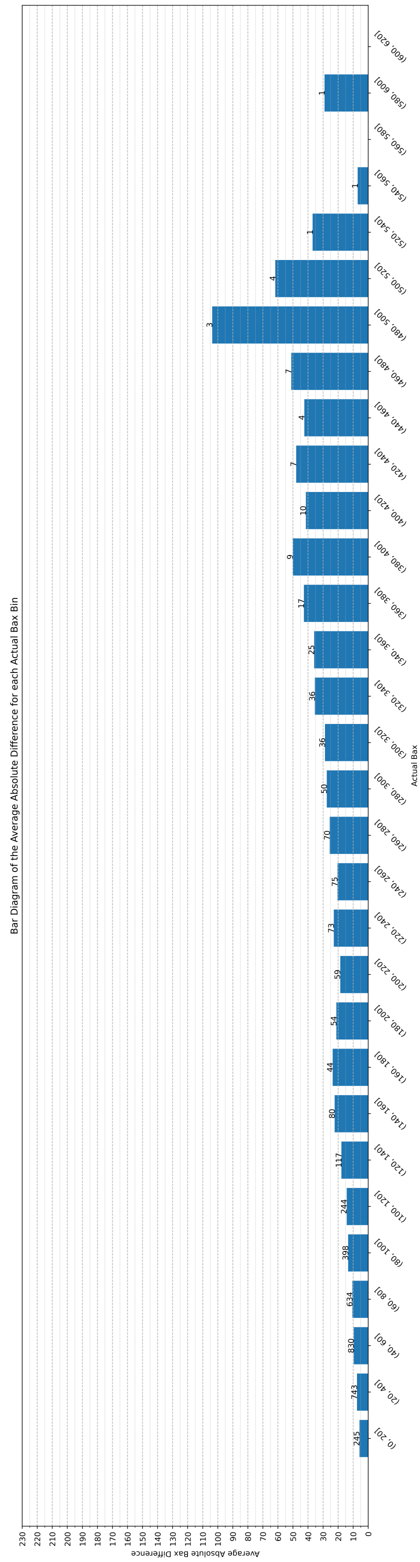


Fig. 10 Average bax error relative to the actual bax

II

Literature Study
previously graded under AE4020

Literature Research

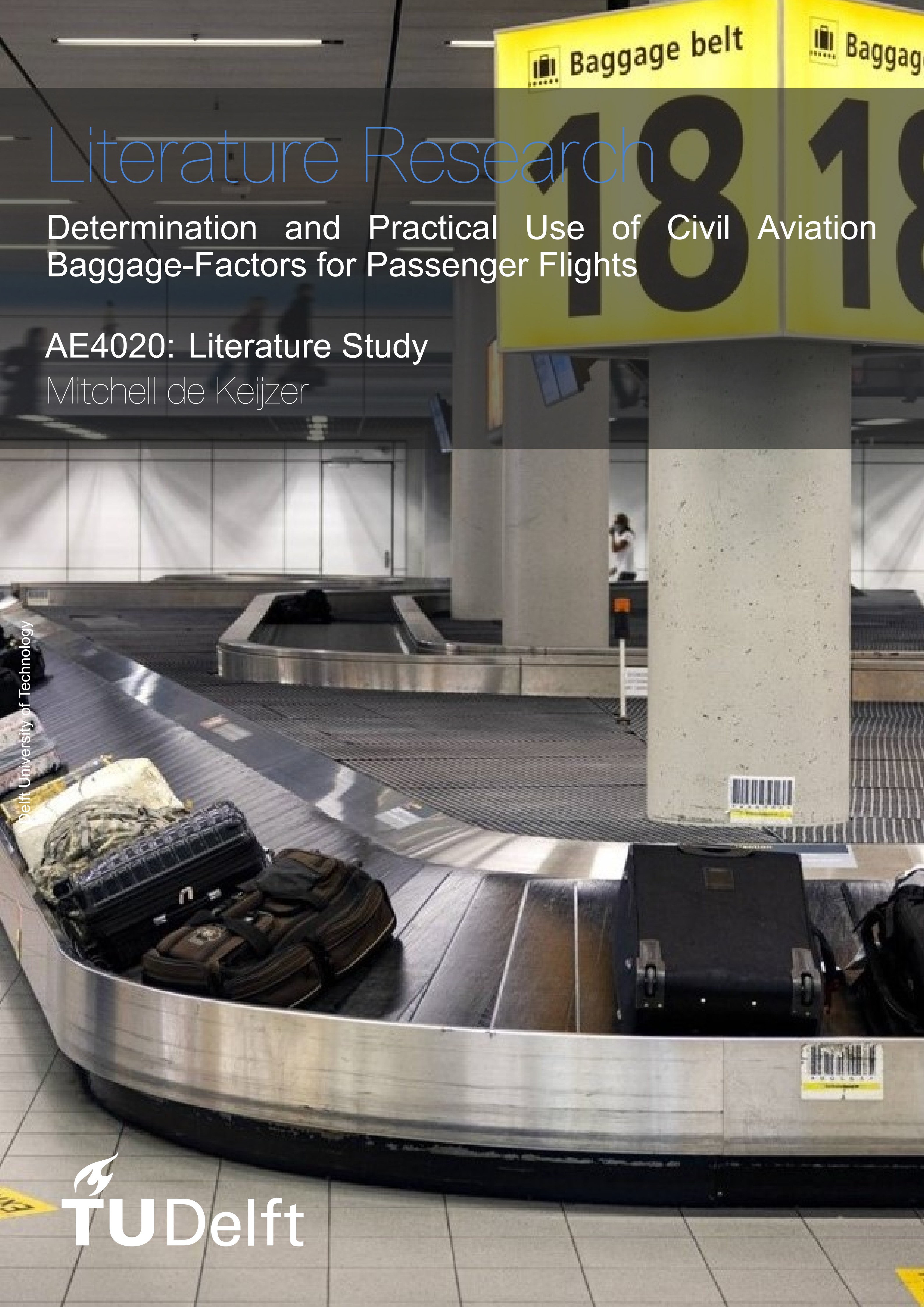
Determination and Practical Use of Civil Aviation Baggage-Factors for Passenger Flights

AE4020: Literature Study

Mitchell de Keijzer

Delft University of Technology

TU Delft



Literature Research

Determination and Practical Use of Civil Aviation Baggage-Factors for Passenger Flights

by

Mitchell de Keijzer

This literature study is submitted in fulfilment of the course AE4020

TU Delft Supervisor:	A. (Alessandro) Bombelli
Institution:	Delft University of Technology
Place:	Faculty of Aerospace Engineering
Schiphol Airport Supervisor:	M. (Mark) van Gaalen
Project Duration:	November, 2022 - January, 2023
Date:	13-03-2023

Cover Image: *NOS News - article 2426718*

TUDelft

Contents

Nomenclature

Abbreviations

ACO	Ant Colony Optimisation
AI	Artificial Intelligence
AR	Auto-Regressive
ARIMA	Auto-Regressive Integrated Moving Average
ARMA	Auto-Regressive Moving Average
BHS	Baggage Handling System
BP	Back Propagation
CNN	Convolutional Neural Network
DP	Dynamic Programming
EMA	Exponential Moving Average
ES	Exponential Smoothing
EWMA	Exponential Weighted Moving Average
FACT	Forecasting, Analysis and Capacity Management department
GA	Genetic Algorithm
GAM	Generalised Additive Models
GRNN	General Regression Neural Network
IP	Integer Programming
KPI	Key Performance Indicator
KPI	Key Performance Indicators
LP	Linear Programming
LSTM	Long Short-term Memory
MA	Moving Average
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MILP	Mixed Integer Linear Programming
MLP	Multilayer Perceptron
NN	Neural Network
PMS	Parallel Machine Scheduling
RF	Random Forest

RMSE	Root Mean Squared Error
RNN	Recurrent Neural Network
SA	Simulated Annealing
SARIMA	Seasonal Auto-Regressive Integrated Moving Average
SES	Simple Exponential Smoothing
SMA	Simple Moving Average
SVR	Support Vector Regression
TrES	Simple Exponential Smoothing + Trend
TrSeES	Simple Exponential Smoothing + Trend + Seasonality
WMA	Weighted Moving Average

List of Figures

1 Introduction

Over the last three decades, the demand at Schiphol increased significantly. From a total demand of 19 million passengers in 1992 to a total of over 40 million passengers in 2002 to a record number of passengers in 2019 of over 70 million [95]. However, in the years from 2019 onward this changed drastically due to the COVID-19 pandemic that hit the world. In the years 2020 and 2021 the total number of passengers decreased to 20 million and 25 million, respectively. After these two years, in 2022 the demand for flights is increasing rapidly again towards the old capacity.

The general increase of passengers results in a challenge for Schiphol Airport in terms of capacity. The higher the amount of passengers present at the airport at the same moment, the higher the pressure on the logistics and operations. The capacity in the departure halls causes longer queues at the check-in desks and at the customs checkpoint. Especially in the period between March and September. Schiphol had issues with staff shortage [23]. Moreover, in April 2022 baggage handlers of KLM were on a strike [81]. The high workload and staff shortage causes dissatisfaction for these employees. Next to the fact that solving the staff shortage problem would make things a lot better, the Forecasting, Analysis and Capacity managemenT (FACT) department at Schiphol believes that more optimised services, operations, and logistics could also prevent these kind of situations in some way.

In order to solve these problems and make operations more efficient, knowing the demand is of utmost importance. Many studies have been conducted over the years for demand forecasting in many areas [111], including air traffic passenger demand. Schiphol Airport has also been forecasting passenger demand for years and with great success, for the models are working well and are improved over the years. However, the FACT department is noticing a change in human behaviour when it comes to the amount of checked baggage passengers take with them. Therefore, a novel forecasting model is needed to be able to forecast this. At present, the majority of research on airport operations primarily focuses on passengers, facilities, or aircraft. However, relatively few studies have been conducted specifically on baggage [65]. Thus a research project on this topic came to light.

There are three types of baggage: checked baggage, hand luggage, and a personal item. All of these baggage items need to be checked for everyone's safety. The hand luggage and personal item will be taken with the passenger into the aircraft, are checked by security when entering the airside, and does not require any further operations from Schiphol. However, the checked baggage is not taken by the passenger, but goes through a whole operation behind the scenes. One can imagine that with 53 million baggage items in the year 2019 - varying between 120.000 and 180.000 pieces per day - this is can become quite a complex operation [39].

Moreover, next to the checked baggage coming in via the departure halls, the checked baggage items from transfer passengers is also coming in. Almost 40% of all checked baggage is transfer baggage. This is mainly due to the fact that Schiphol functions as a hub in the network of the home carrier KLM and the SkyTeam partners [39]. Being able to forecast the amount of checked baggage items can pave new studies for optimising the entire baggage (handling) process. In this study methods are investigated for forecasting the amount of checked baggage items that can be scientifically supported. Additionally, research will be conducted for an optimisation model that uses the output of the forecasting model as its own input and makes decisions based on optimising baggage (handling) processes and the associated resource allocation. Therefore the objective of this research is:

Find a scientifically supported means of determining 'Baggage Factors' in civil aviation, and advise on the practical implementation of a method for periodic forecasting of baggage items.

The goal of this literature study is to give an overview of the currently available literature on existing forecasting methods and resource allocation optimisation methods. This is done in order to provide more in-depth analysis on the research topic before answering the final research question. The outline of the report is as follows. First, the research outline is stated in chapter 2, where the research problem, objective, questions, and planning is described. Next, a general overview of the baggage handling process is given in chapter 3. Then the currently available literature on forecasting methods is studied in chapter 4. Also a general definition of baggage load factors is described and forecasting approached and variables are argued. Lastly, chapter 5 provides an overview of the literature on exact solution methods and meta-heuristic methods for resource allocation scheduling optimisation.

2

Research Outline

This chapter describes the outline of the research on the determination and practical use of civil aviation baggage-factors for passenger flight. In section 2.1 the research problem is first defined. The objective that arises from the problem statement is then described in section 2.2. Following from these first two sections, in section 2.3 the research question and sub-questions are formulated. Finally, section 2.4 gives a general planning of the thesis project.

2.1. Research Problem

The Forecasting, Analysis and Capacity Management department of Amsterdam Airport Schiphol (FACT in short) is responsible for creating a forecast of Aircraft movements and Passenger and Baggage counts per movement, which is used in the planning of operational processes for amongst others baggage handling.

Schiphol collects huge amounts of data from sensors, air movements, camera footage, and information systems. All this data is being processed by data analysts. One of the purposes of collecting and analysing this data is to make forecasts. Having good working forecasting models is essential to optimise the logistics and operations at Schiphol. For example, being able to forecast the amount of passengers at a certain date within a certain time frame, enables the possibility to prepare all logistical and operational systems at Schiphol.

Central to the forecasts is the concept of the ”load factor”: the relative amount of passengers or baggage items given the potential maximum. Being able to predict how many passengers will be on board of an aircraft gives many advantages as explained before. As of this moment, a forecasting model is already present for the passenger load factor at Schiphol and is proven to work well. An important item in forecasting Baggage items per flight is the ”Baggage load factor” (of ”Baggage factor” in short): the relative amount of baggage items given the number of passengers on board of an aircraft. However, predicting how many baggage items these passengers take with them, focused on checked baggage, is still a challenge. Currently, the best way of predicting this amount of baggage items, is by analysing available historical data and based on that make an expert guess. This method was effective due to relatively stable numerical factors and minimal time investment. However, the introduction of baggage fees by airlines in 2008 led to changes in consumer behaviour [56, 96]. A subsequent increase in baggage fees by major European airlines in 2017 further altered behaviour [6, 8]. These variations were apparent within the operations of the organisation, yet still manageable. The emergence of the COVID-19 pandemic resulted in a substantial decline in commercial air travel, leading to a cessation of individual flights. Despite the apparent control of the virus in 2022, forecasting baggage load factors based on historical data has become more challenging and uncertain. This is due to the non-representative nature of data from the years 2020 and 2021, as well as the unknown changes in consumer behaviour following a two-year hiatus in air travel. Moreover, employees of FACT argue that it is important to provide a scientific basis for the determination of baggage load factors for resource allocation.

The baggage factor is thought to be influenced by factors as flight destination type (vacation of business), season, operating airline, passenger demographic, and more. An opportunity with this baggage factor is to look at possible decision-making tools to tackle the problems and opportunities that were stated above. Not being able to predict the amount of baggage items sufficient enough can lead to some problems, or otherwise stated: being able to predict these amounts of baggage items can lead to interesting opportunities. Many studies have been performed to optimise baggage handling. It would be very interesting to see what the impact of knowing the amount of baggage items coming in will have on the logistics, systems, and operations of baggage handling.

2.2. Research Objective

In section 2.1 the problem of the research was stated and why this is a problem and arises opportunities. Having this in mind, the objective of the research is to find a scientifically supported means of determining 'Baggage Factors' in civil aviation and advise on the practical implementation of a method for periodic forecasting of baggage items. Moreover, to use this forecasting of baggage items for a decision-making tool to optimise the logistics within Schiphol by looking at the planning for the resource allocation for Baggage Handling Systems (BHS).

To reach this objective, first all variables that are affecting the amount of baggage items need to be identified in order to find the inputs for a forecast model. Secondly, a forecasting method will be used which formulates the baggage factor as output. This baggage factor will benchmark for a decision-making optimisation tool for the resource allocation within Schiphol.

The contribution of this research to Schiphol is a scientifically supported means to predict the amount of baggage items given the number of passengers on board of an aircraft which can be used to optimise the BHS planning. With this baggage factor, Schiphol will be able to further optimise its planning and certain operations coming with those logistics.

2.3. Research Question

The main research question has been defined as follows:

How can a decision-making tool that sequentially forecasts baggage load factors and utilises the prediction to allocate resources accordingly improve the baggage handling operations at a hub airport?

The main focus thus lies on the determination of baggage factors by the means of a forecasting model for the reason that this is currently not present at Schiphol. The second focus point is to discover ways to implement these baggage factors by developing a decision-making tool to optimise certain logistics or operations within Schiphol, mainly focused on resource allocation for the BHS. In order to answer this, the main research question is divided into and supported by several sub-questions. By answering these sub-questions, the answer to the main research question will be provided. The sub-questions are:

1. **What variables, constraints, and assumptions need to be taken into account when determining baggage factors?**
   The baggage factor is influenced by different factors. The corresponding variables and constraints must be identified and for some of these assumptions need to be made. If possible, the assumptions must be aligned with policy requirements such that the output of the model is based on valid and pragmatic choices to ensure customer satisfaction.

2. **What type of forecasting method is needed to determine baggage factors and what are the bias factors?**
   There are numerous methods for making forecasts, each with its distinctive approach. The task

can be accomplished either by conducting a basic data analysis or by constructing a neural network, depending on the approach preferred and complexity of the forecasting problem.

3. **Is it necessary to look at the number of passengers or is it possible to predict the amount of baggage items on itself considering other features and what would the effect be?**
   The prediction of the amount of passengers is prone to errors and bias and therefore not always 100% correct. Using this prediction to predict the baggage factors, is a prediction on a prediction to find the number of baggage items.

4. **What are the risks of using a forecasting model for predicting baggage factors?**
   A forecasting model is almost never 100% right. There is always some bias and a certain error in the output. It is therefore important to look at the risks that come with making decisions based on the outcome of a forecasting model.

5. **How can forecasting Baggage Factors improve customer satisfaction?**
   In the core, Schiphol is a service provider for both airlines and passengers. Customer satisfaction is a very important part of the service criteria. Therefore, examining in what ways the forecast of baggage factors can improve this is an interesting study.

6. **What are the opportunities when it comes to optimising the BHS knowing the amount of baggage items beforehand?**
   The operation of baggage handling is quite complex. Many optimisation studies have been performed on this topic. It is therefore interesting how knowing the baggage factors might influence this system and what optimisation opportunities this presents.

7. **What is the impact of incorporating periodic forecasting and baggage factor analysis on the logistical and operational implementation of BHS**
   Assuming the availability of an efficient forecasting model that is user-friendly, what are the potential impacts on the department of FACT, as well as on the micro and macro operational and logistical levels of Schiphol Airport, for Schiphol employees from different other departments, external handlers, and airlines?

8. **What other operational and/or logistics business at Schiphol can be positively influenced by knowing the baggage factors?**
   Other logistical and operational business might also be influenced by forecasting the baggage factors. Schiphol is a large organisation with many logistical challenges. Identifying which parts can be influenced positively is therefore interesting.

2.4. Research Planning

The research project consists of four phases. Each phase represents a moment in time where a certain part of the project will be performed. In Figure 2.1, this planning can be seen. A more detailed explanation of each phase is given below.

Phase 1: Literature Study Phase

In this phase, a literature study will be performed and a research plan will be created. The goal of this phase is to come up with a research plan with supporting literature to tackle the research question and afterwards being able to dive deeper into different methods. Deliverables that come with this phase are a literature study report and a research plan report.

Phase 2: Initial Thesis Project Phase

Once phase one is finished, the second phase starts with a kick-off meeting. Here, the findings of the literature study will be discussed and the project will officially start. This part will focus on sub-questions 1, 2, 3, and 6. Additionally, data obtained from Schiphol Airport will be analysed, forecasting models will be established, and initial evaluations will be conducted. The optimal solution for a decision-making tool will be determined through this process. At the end of this phase, the mid-term meeting will take place. This meeting is meant as a go / no go meeting where the progress will be reviewed. A no-go will require a revision of the work done up till then. The performance will be evaluated and feedback will be given by experts.

Phase 3: Final Thesis Project Phase
In this phase, the thesis final parts of the thesis project will be performed. This part will focus on sub-questions 4, 5, 7, and 8. At the beginning, a selection of the most promising forecasting methods will be made, and the most suitable ones will be finalised. Concurrently, methods for decision-making tools will be developed and thoroughly evaluated. The process will include model verification, validation, examination of results, and drawing of conclusions to ultimately arrive at a final product. This phase ends with the Green Light meeting, which is the final meeting before the graduation phase. A draft of the final thesis report will be handed in beforehand and a presentation will be given during the meeting. This will be evaluated and final feedback will be given to incorporate in the final version.

Phase 4: Graduation Phase
The final phase is the graduation phase, where the last feedback will be processed and the last required improvements will be made. No new developments of the model will be made during this phase. This phase ends with handing in the final report and paper and the graduation presentation. After which a defence will be hold which will give the final go or no go for graduating.

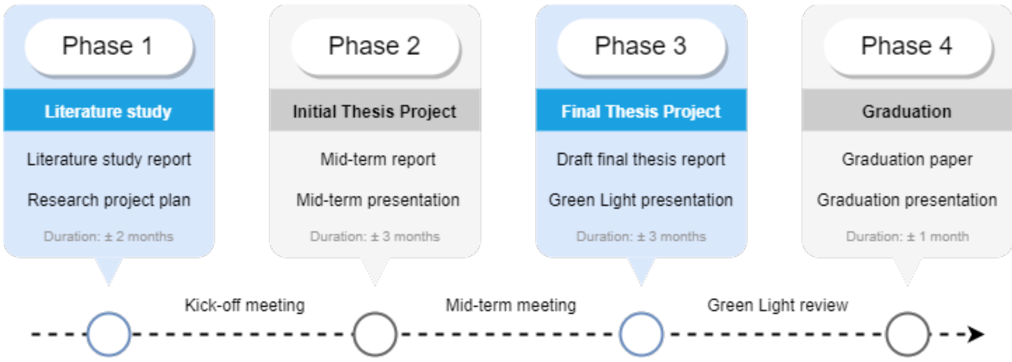


Figure 2.1: General thesis project planning

3 Baggage Handling Process at Schiphol

The handling of passengers' baggage items is one of the major aspects of the operational business of an airport. Throughout time, due to some (major) historical events, the security at an airport has become more and more urgent. Therefore, airport and especially baggage security have become a major part of this operational business. For these security reasons, all baggage at Schiphol that is checked in, must undergo extensive security checks behind the scenes. Moreover, due to the growing number of passengers and thus increase of baggage items, handling baggage is becoming a bigger challenge than before logistical and operational. This chapter is divided into two sections, section 3.1 provides an overview of the step by step process of baggage handling in general and at Schiphol and section 3.2 lays out some key performance indicators for forecasting checked baggage items and optimising the BHS.

3.1. Step by Step Overview of Baggage Handling at Schiphol

There are three types of baggage: checked baggage, hand luggage, and a personal item. The latter two are to be taken with the passenger into the aircraft. The first, checked baggage, is to be left behind at one of the check-in desks and will eventually be loaded into the assigned aircraft. The passenger has to collect this checked baggage at arrival at one of the baggage belts just before the arrival hall. There are two ways for baggage to enter into the system: checked-in baggage and incoming transfer baggage.

In Figure 3.1 an overview of the baggage handling process can be found. The baggage handling process begins with the input of baggage items into the system, which can be done via checked-in baggage or incoming transfer baggage. Checked-in baggage is brought to the airport by the passenger and labelled at the check-in desk before being entered into the BHS, which can be done manually or through automated check-in desks. Incoming transfer baggage is offloaded from an incoming flight by another handling company and then entered into the BHS. The BHS scans the baggage label to read all its information, including the flight number, screens the baggage for security purposes, and determines to which terminal the baggage needs to be transported and whether the flight is to be loaded via robots or manually. If the baggage is to be loaded via robots, it is transported to a buffer, where it is stored and batched by an automated buffer robot until a batch is complete. The weight and volume of each baggage item is measured and stored, and used by intelligent algorithms to calculate when a batch is ready to be loaded. Once a batch is complete, the baggage is retrieved and delivered back to the BHS, where it is loaded into a baggage cart via robots or manually by employees. The system also checks whether the baggage is on-time or too late, and handles it accordingly. Late-arriving baggage is dumped on a carousel and may be delivered to an airplane through a process called ”milkrun” if the flight is still parked at the aircraft stand. If a baggage item is dumped at the carousel after the flight has already departed, it is labelled as ”mishandled bag” and typically sent on the next flight with the same destination.

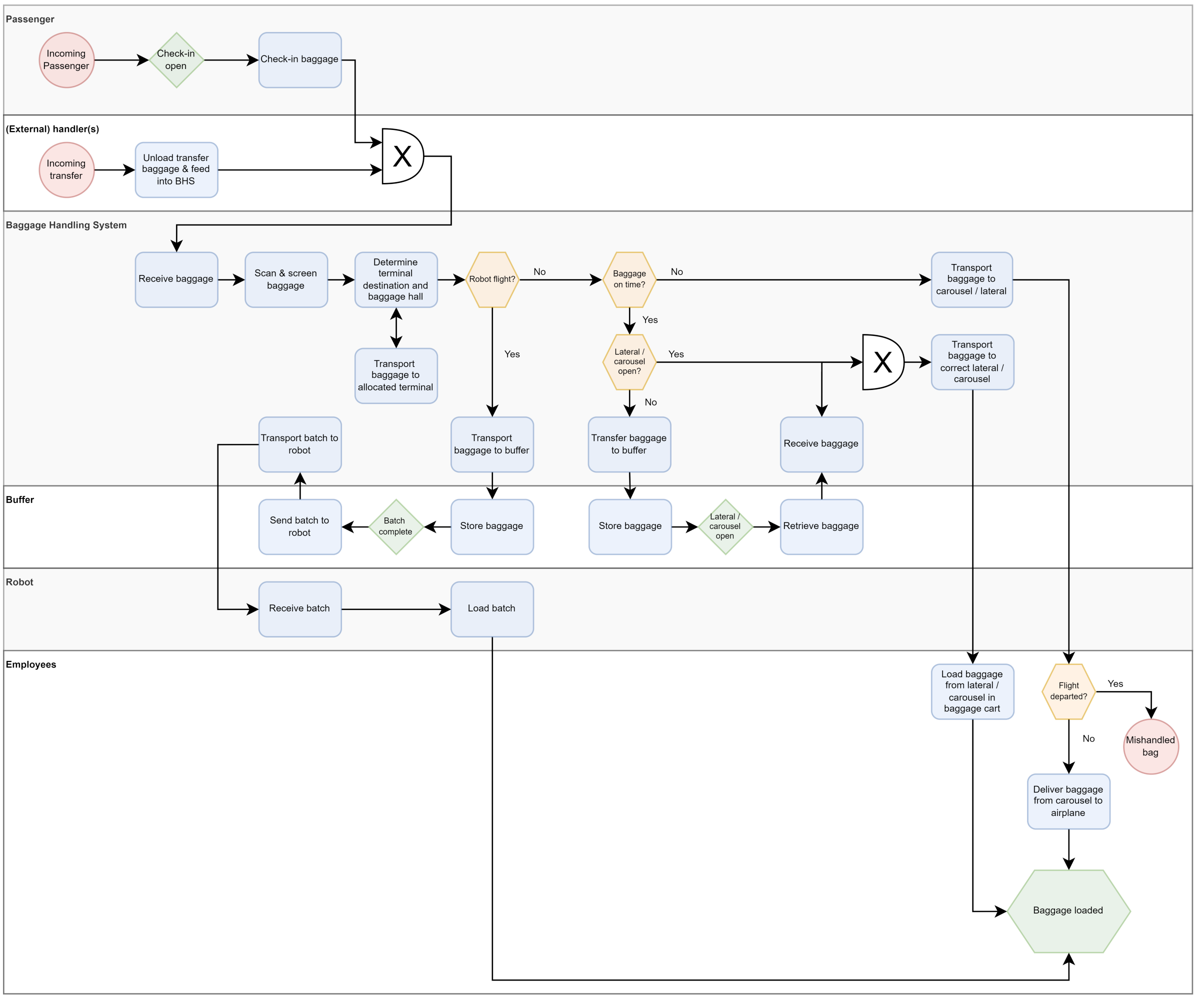


Figure 3.1: General baggage handling process

In the baggage handling process, there is potential for operational improvement. By accurately forecasting the number of baggage items, it is possible to identify which airlines and terminals have the highest volume of checked baggage. This information can be used to optimise the utilisation of check-in counters and reduce queues. Additionally, forecasts of transfer flight baggage can be used to optimise the scheduling of handlers, which is both cost and operationally efficient. Once the baggage items enter the BHS, the forecasted number can be taken into account to optimise the system's performance. Research has been conducted on optimising BHS, and further studies could examine the impact of these optimisations on baggage volume. Additionally, by analysing the maximum capacity of buffers and the overall layout of the airport, an interesting research is to improve the scheduling and layout of the BHS. For robots used in baggage handling, knowing the volume of checked baggage can be used to improve the time-efficiency of the process. Lastly, by optimising human resource allocation for the loading of baggage onto airplanes, the overall efficiency of the baggage handling process can be improved.

In addition to the operational improvements described above, scheduling of service teams, such as those provided by companies like Viggo or Vanderlande, can also be optimised. These service teams are essential during peak periods when the BHS experiences malfunctions or failures. Identifying the peak times when the BHS experiences high demand can aid in the scheduling of these teams. Furthermore, maintenance can be more efficiently planned by identifying periods of low baggage demand, during which certain parts of the system may not be needed.

3.2. Key Performance Indicators

The previous section outlined the general baggage handling process at Schiphol airport. It is crucial to ensure that the specific baggage handling process at the airport aligns with and optimises this general process. In order to perform any optimisation, it is necessary to be able to evaluate the performance of the baggage handling process. The performance of the process is assessed by evaluating key performance indicators (KPIs). The KPIs are divided into two categories: one for the forecasting part and one for the BHS optimisation.

KPI's for forecasting the checked baggage items are:

1. **Amount of passengers on board:** this KPI gives an overview of the amount of passengers there are on a flight, which is always connected to the number of checked baggage items there are in total per day.
2. **Passenger Factors:** this KPI focuses on the behavioural patterns of passengers, i.e. what are the trip purposes, character traits, etc.
3. **Flight factors:** this KPI focuses on the objective variables of a flight, i.e. when the flight departures, where the destination is, the duration of the flight, what type of airline and aircraft is being flown.

KPI's for BHS operations optimisation are:

1. **Amount of Baggage Items:** to optimise a part of the operations around the BHS, knowing the forecasted amount of baggage items is an important KPI.
2. **Airport On-Time Performance:** this metric is a key indicator of the efficiency of the baggage handling process. It is expressed as the percentage of flights that are handled within the specified time frame. A high OTP is crucial to minimise the number of delayed flights. A target timestamp is established, before which all baggage for a flight must be loaded and prepared for transport to the aircraft. If all baggage is loaded prior to this timestamp, the flight is considered ”on-time”; otherwise, it is considered ”too late.”
3. **Employees:** the number of employees required to execute the baggage handling process is a crucial metric to consider. The cost of employees can have a significant impact on expenses, and thus efforts should be made to minimise the number of employees required to perform the process.
4. **Employee Productivity:** the productivity of employees can be measured by determining the percentage of time that they are engaged in productive work. This metric is calculated by recording the amount of time that employees are occupied, and comparing it to the total time they are available. This will help to understand how much time they are spending on working and how much is wasted on non-productive activities. With that knowledge, a more optimised schedule can be made
5. **Time on Lateral:** a lateral is a transfer station for baggage handling where baggage is transferred between different conveyor systems or areas of an airport. The dwell time of a piece of baggage on the lateral or conveyor belt before it is loaded into a baggage cart is an important metric to measure. This can be calculated by determining the difference between the timestamp when the baggage is deposited on the lateral by the BHS and the timestamp when it is loaded into the baggage cart. This metric will help to understand how much time is spent by the baggage on the lateral and how efficient is the system in processing the baggage.

More on these KPIs will be described in the next chapters.

4

Baggage Forecasting and Load Factors

This chapter focuses on the literature on forecasting baggage load factors. section 4.1 describes the baggage load factor definition and some forecasting approached. Problems within the literature on baggage forecasting and some solutions are presented in section 4.2. Lastly, section 4.3 describes different forecasting methods with examples of usage within different industries.

4.1. Baggage Load Factor in General

As described in section 2.1, the Baggage Load Factor is defined as the relative amount of baggage items given the number of passengers on board of an aircraft. The ultimate purpose of the baggage factors is to predict the number of baggage items per flight and thus per day or time period. However, reaching this goal can be done in multiple ways. The first approach is to look at historical data of passengers and the number of checked baggage items, basically the historical baggage factors. From these two data streams, a pattern might be found and a prediction can be done for in the future. A second approach is to look at the baggage items as a separate and independent items of passengers. Here the prediction of number of baggage items might be predicted by looking at different variables that determine the amount of baggage items. A third approach could be a combination of the first two approaches. By looking at variables that determine both the behaviour of passengers and baggage items. Using the current forecasting model of Schiphol to predict the number of passengers and using that to predict the amount of baggage items. In the subsections below, each approach will be further explained.

- **Forecast based on historical data of baggage factors**
  This methodology represents a straightforward approach and is relatively uncomplicated. By analysing historical data on a per-airline, per-time slot, and per-season basis, a forecast of baggage load factors can be generated for future periods. This approach is similar to the current methodology, but utilises a forecasting model instead of relying on expert estimates. If feasible for the chosen forecasting technique, incorporating external variables such as holidays and conferences can potentially enhance the accuracy of the forecast.
- **Forecast baggage items instead of baggage factors**
  The baggage load factor is a commonly used metric for estimating the number of baggage items. This alternative approach bypasses the calculation of the baggage load factor and instead aims to predict the number of baggage items directly through the forecasting model. This methodology focuses on variables that influence the decision of passengers to travel with checked baggage at specific times. As this approach incorporates a greater number of variables compared to the other method, it may result in a more complex forecasting model. However, by considering a larger number of variables, it is hypothesised that the forecasting model would exhibit improved accuracy and performance.
- **Combination of passenger forecast, baggage item forecast, and historical baggage load factor data**
  This methodology builds upon the previous approaches by incorporating both the prediction of

passenger load factor (which can be obtained from Schiphol Airport) and the forecast of the number of baggage items. The combination of both predictions result in a simple calculation to get the baggage load factor. The predicted baggage load factor will then be compared to historical data and adjusted as necessary. While this approach is the most complex among the proposed methods, it is expected to yield the best performance.

For all three proposed approaches, it is important to consider that as the forecast date approaches, more information becomes available regarding bookings that have been made, including the number of tickets booked and the number of checked baggage items that passengers intend to travel with. Incorporating this information into the forecasting model may aid in improving the accuracy of predictions for shorter-term forecasts.

4.2. Baggage Forecasting Literature, Problems, and Variables

As mentioned in chapter 2, at Schiphol there was no necessity in the last few decades to invest time and energy in developing a model that can forecast the number of checked baggage items. A prediction of air traffic passenger demand and the passenger load factor was sufficient. At present, the majority of research on airport operations primarily focuses on passengers, facilities, or aircraft. However, relatively few studies have been conducted specifically on baggage [65]. Cheng et al. [21] conducted a comparative study on forecasting methods for departure flight baggage demand. Similarly to the current report, Cheng et al. argue that it is crucial to establish a scientific foundation for the allocation of resources in the checked baggage stage, which enhances the efficiency of service provided at airport passenger terminals. The results of the study based on a multiple linear regression model and a back propagation (BP) neural network show that the multiple linear regression method has a lower average relative error compared to the BP neural network. Moreover, the average relative error decreased when changing the data for all flights to single flight data to the same destination flight data. In another paper, Ma et al. [71] conducted research investigating the prediction of checked baggage demand for departure flights with the aim of optimising efficiency in airport operations during the check-in process. A SARIMA model was proposed for predicting the baggage volume. The study found that the SARIMA model was capable of accurately forecasting the checked baggage demand on a long-term basis, which can aid in the proactive allocation of airport resources. The methods mentioned in both studies will be described in further detail in section 4.3.

When considering the variables that could be taken into account for forecasting the baggage load factor of number of checked baggage items as in one of the three approaches that were mentioned in section 4.1, they can be divided into two categories as can be seen in Figure 4.1: the number of passengers on board and the number of checked baggage items. For the first category, it can be assumed that the number of passengers on board is known via the forecast of FACT. However, it is still interesting to see the difference in the type of passengers. How many origin & destination passengers and how many transfer passengers are forecasted to be on board of the plane? And does the type of passengers impact the baggage load factor? These questions are interesting to combine with the second category, which one layer deeper focuses on passenger variables and flight variables. Most of the passengers variables are believed to be subjective and it is assumed that data is either not available or less reliable. Still it is interesting to analyse whether there are certain trends that can be taken into account. The flight variables on the other hand are all objective and data should be available for these variables. Important to note is that these variables have not been structured on their impact on the baggage factor yet. Defining the weights of these variables, is to be done after an extensive data analysis.

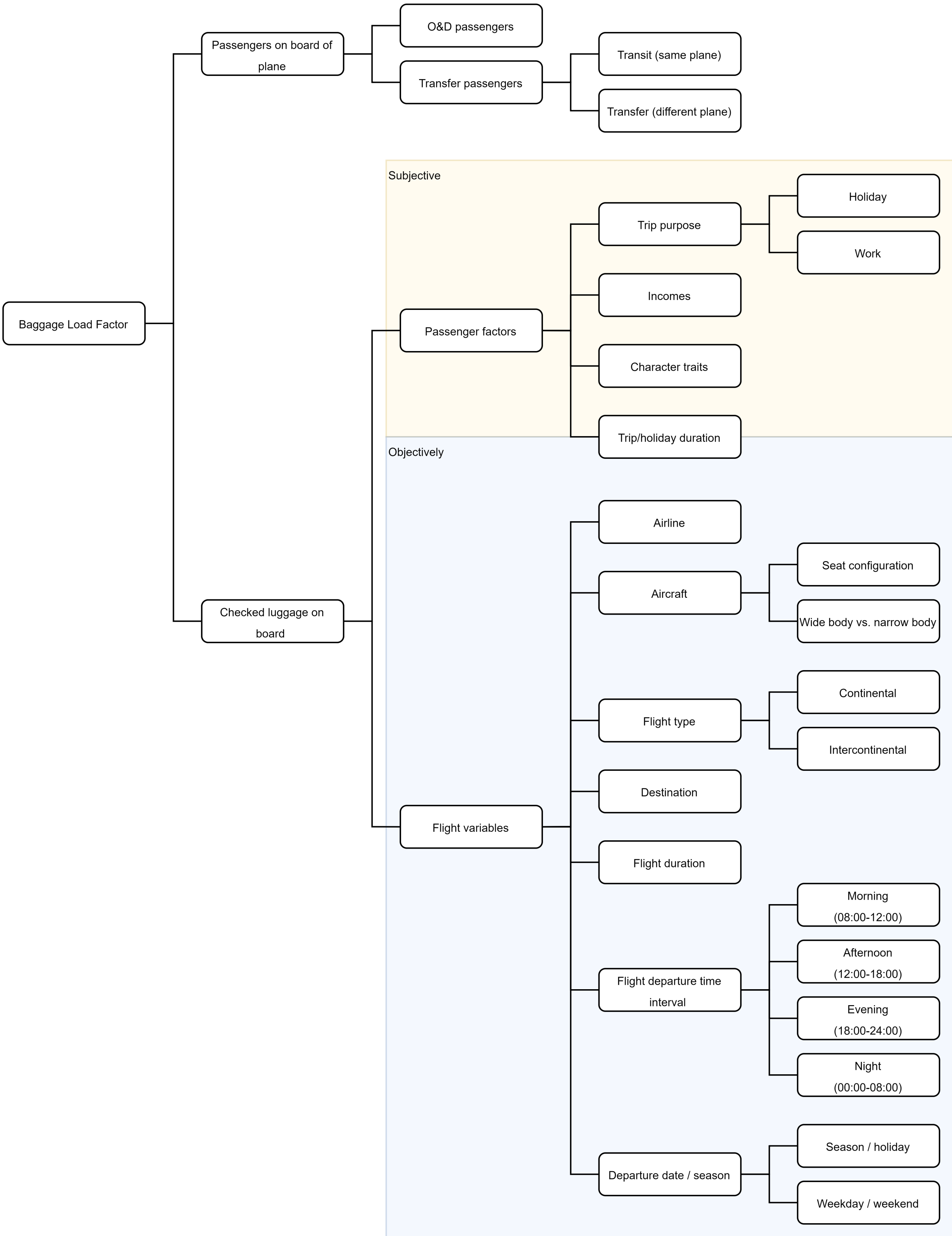


Figure 4.1: Baggage Factor variables feature tree (not weighted and structured on impact on baggage factor)

4.3. Forecasting Methods

There are multiple different forecasting techniques currently in existence. This section will focus on some of the existing forecasting techniques and explain whether they are usable for the research that has to be conducted. There are three different basic forecasting types: qualitative techniques, time-series analysis & projection, and causal models. Qualitative forecasting techniques are methods of predicting future trends and outcomes based on qualitative data. They involve the subjective assessment of a situation and the application of expert judgement to arrive at a forecast. Qualitative forecasting techniques are typically used in situations where more traditional quantitative methods are not appropriate or feasible, such as when forecasting customer demand, new product acceptance, or industry trends. Examples of qualitative forecasting techniques include scenario analysis, surveys, Delphi technique, and cause and effect diagrams [19]. time-series forecasting is a method of predicting future values of a variable based on the past values of the same variable. It is a form of regression analysis and is typically used to make short-term predictions. The most commonly used time-series models are moving average, exponential smoothing, Box-Jenkins, and neural networks. Projection is a process of predicting future outcomes based on the current trends. It is a form of extrapolation, which uses existing data to estimate future values. It is particularly useful for analysing trends in long-term data or for predicting future values of a variable. time-series models do not focus on why certain relationships between variables exist due to for example trends or seasonality. These models use a mathematical formula on the past and utilise it to make a forecast of the future. Causal models on the other hand are a type of forecasting that use statistical methods to identify the relationships between variables and how they impact the future. These models can be used to predict the outcome of a given event based on past data and variables. Just as for time-series analysis and projection techniques, the past is important for these kind of models. Commonly used causal models are regression models and econometric models. [19]

Zooming in on time-series models, two categories can be defined: uni-variate and multivariate. Univariate time-series models use one variable of which the past data is used, the target variable, to make a prediction. Multivariate time-series models have the ability to predict multiple related variables at the same time. These models utilise the correlation between target variables to improve the performance of the model, given that there is a strong correlation [59]. However, as described above, time-series models do not look at the causal relationship between the variables. Some forecast problems require the model to take other independent information into account to find possible relationships. This independent information can be called explanatory variables. In these cases, supervised and unsupervised machine learning can be applied to the model to find the relationships between the target variable(s) and explanatory variable(s). The main difference between supervised and unsupervised learning is whether the data has already been classified or labelled, meaning that the model either does have information about what the correct output should be or not [52]. Thus, unsupervised learning must find patterns and relationships in the data on its own. Whereas supervised learning can visually detect correlations between the variables. Finally, with considering supervised machine learning, classification and regression are two types of problems. The main difference is that classification has a categorical target variable and regression is numerical. Since this research focuses on the prediction of baggage factors, numerical supervised machine learning should be applied. [52, 59]

This research will primarily focus on the the time-series and causal forecasting types, since qualitative forecasting techniques are mainly not mathematical methods and more subjective. This is however a method that Schiphol should consider next to the forecasting model, as a forecast model is almost never 100% accurate and a qualitative forecasting technique might be a very good solution as a second opinion. In the subsections below, some of the most common forecasting methods are described with examples for what they have been used in the past.

4.3.1. Moving Average

Moving average forecasting is a form of time-series forecasting that uses a weighted average of historical data to make predictions about future values. It is based on the assumption that the future values of a given series are determined by the average of the past few values. This technique is useful for predicting the long-term trend of a time-series, as it smooths out short-term fluctuations and emphasises the overall direction. It is also useful for making short-term forecasts, as it can provide a quick indication of how the series is likely to develop in the near future. [32]

There are different kinds of moving average techniques. The first and most simplistic one is the simple moving average (SMA) technique. The SMA technique is a method of analysing data points by calculating the average of a predetermined number of data points. This is done by taking the sum of the data points and dividing it by the total number of data points. Equation 4.1 [32] shows the formula for the SMA. The resulting number is the average and this value can be plotted on a chart to identify trends in the data, such as whether the data is increasing or decreasing over time.

$$SMA = \frac{A_1 + A_2 + \ldots + A_n}{n} \tag{4.1}$$

where A is the average in period n and n is the number of time periods.

A second technique is called the exponential moving average (EMA) technique. The EMA technique is a variation of the SMA which gives more weight to recent data points than to those further in the past. Due to this design, the weights of data points that are getting older are falling exponentially [70]. This technique is used to better capture changes in trends over time. The EMA is calculated as can be seen in Equation 4.2 by taking the previous period's EMA and adding a percentage of the current period's value to it, resulting in a smooth curve that reflects changes in the trend more accurately. [32]

$$EMA_t = \left[V_t \times \left(\frac{s}{1+d}\right)\right] + EMA_y \times \left[1 - \left(\frac{s}{1+d}\right)\right] \tag{4.2}$$

where EMA_t is the EMA today, V_t is the value today, EMA_y is the EMA yesterday, s is the smoothing value, and d is the number of days.

According to Ghobbar et al. [35], the (Exponential) (Weighted) Moving Average (EWMA) is mainly effective as a forecasting tool for time-series data with a linear trend. A weighted MA allows for weighting to be assigned to the data being averaged. The paper evaluated different forecasting methods for intermittent parts demand in the field of aviation. In the evaluation of this paper, it became clear that the WMA is one of the superior methods for forecasting. However, the experiments of Bartezzaghi et al. [7] showed that the EWMA is applicable only with low level of lumpiness. Chen et al. [20] used a modified moving average method for airline passenger forecasting, however, the output of the MA had a significant error and therefore it was necessary to use a neuro-fuzzy model to lower the error. This model did show that the error was attenuated significantly, meaning that a MA method could be used for non-linear forecasting with the correct data.

4.3.2. Exponential Smoothing

Exponential smoothing (ES) is a method of forecasting that is based on weighted averages of past data, and another term for EMA which was explained in subsection 4.3.1. However, compared to the MA method, it is a type of time-series forecasting that can take into account the trend, seasonality, and level of a data set to predict future values. Exponential smoothing assigns more weight to recent data points, which makes it more responsive to recent changes in the data. There are three main types of exponential smoothing: Simple, double, and triple exponential smoothing. Simple exponential smoothing (SES) assigns a weight to all past data points, with more weight assigned to more recent data points. This method is used to forecast data that does not show any seasonal or trend patterns. Double exponential smoothing (also called Simple Exponential Smoothing + Trend, or TrES) adds a weight to the trend component of the data, which allows it to better reflect changes in the trend. This method is often used when the data has a clear trend. Lastly, triple exponential smoothing (also called Exponential Smoothing + Trend + Seasonality, or TrSeES) adds a weight to the seasonality of the data, which allows it to better reflect seasonal changes. This method is used when the data has a clear seasonality pattern. [54]

Ghobbar et al. [35] mentions that SES is mainly effective in circumstances of low and intermittent demand, moreover, for this kind of demand it is in practice the most frequently used method. Furthermore, it is mentioned that the TrES method is mainly effective for forecasting time-series data with a linear trend. The results of this study conclude that the WMA method is much superior to exponential smoothing. Another study conducted by Adeniran et al. [1] looking into the domestic air passenger demand in Nigeria for the year 2018 by comparing the simple exponential smoothing to the simple moving average method using yearly data from 2010 to 2017. The results of the study showed that the moving average method forecast came closer to the raw data than the exponential smoothing method, meaning that the exponential smoothing method gave a less good forecast.

An extension of the SES method is made by Holt [49]. Holt's method, also known as Holt-Winters, is an extension of exponential smoothing that incorporates both trend and seasonality in the forecast. The method uses three equations to model the data: one for the level of the series, one for the trend and one for the seasonality. The level equation is the same as in simple exponential smoothing, where the forecast for the next period is a weighted average of the previous forecast and the last observation, with the weight determined by the smoothing parameter α. The trend equation adds a second smoothing parameter β, which controls the weight given to the deviation of the last forecast from the last observation. The seasonality equation adds a third parameter γ, which controls the weight given to the deviation of the last forecast from the last seasonal index. The method starts with an initial estimate of the level, trend and seasonality, and then iteratively updates the forecast based on the new data [49, 50, 82]. The SES equation is denoted as follows [50]:

$$\begin{aligned} \hat{y}_{t+h|t} &= \ell_t \\ \ell_t &= \alpha y_t + (1-\alpha)\ell_{t-1} \end{aligned} \tag{4.3}$$

where $\hat{y}_{t+h|t}$ is the forecast and ℓ_t is the smoothing equation. Holt's first method is similar to the one used in the SES and is only extended by allowing the forecast of data with a trend and involves two smoothing equations next to the forecast equation [50]:

$$\begin{aligned} \text{Forecast equation:} \quad \hat{y}_{t+h|t} &= \ell_t + hb_t \\ \text{Level equation:} \quad \ell_t &= \alpha y_t + (1-\alpha)(\ell_{t-1} + b_{t-1}) \\ \text{Trend equation:} \quad b_t &= \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1} \end{aligned} \tag{4.4}$$

Extending it even further with seasonality, the Holt-Winters' method is similar to the equations of Holt's method adjusted for the seasonality. It can be represented by the following equations [50]:

$$\begin{aligned} \text{Forecast equation:} \quad \hat{y}_{t+h|t} &= \ell_t + hb_t + s_{t+h-m(k+1)} \\ \text{Level equation:} \quad \ell_t &= \alpha(y_t - s_{t-m}) + (1-\alpha)(\ell_{t-1} + b_{t-1}) \\ \text{Trend equation:} \quad b_t &= \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1} \\ \text{Seasonal equation:} \quad s_t &= \gamma(y_t - \ell_{t-1} - b_{t-1}) + (1-\gamma)s_{t-m}, \end{aligned} \tag{4.5}$$

where for all equations above ℓ_t is an estimate of the level of the series at time t, b_t is the estimate for the trend (slope), the smoothing parameters α, β, and γ have a value between 0 and 1, h makes the forecast function trending (h-step-ahead forecast), s_t is the seasonal component, k is the integer part of $(h-1)/m$, and m is the frequency of the seasonality. [50]

In the study of Ghobbar et al. [35], the results show a superiority not only of the WMA, but also the Holt and Winters method. Another interesting study has been performed by Rusyana et al. [89] where Holt's method (taking into account trends) was compared to Winters' method (taking into account seasonality) for forecasting the number of domestic passengers arrivinig and departing from Sultan Iskandar Muda International Airport in Indonesia. The results of this study found that the best model for this forecast is the method of Winters' exponential smoothing. However, looking at the results of both methods and the criteria of measuring the accuracy of the methods, both are working very good to excellent with the right smoothing parameters.

4.3.3. Linear Regression

Linear regression is used to predict the value of a continuous variable (a dependent variable) based on the values of one or more other independent variables. It is a type of causal model, meaning that it is used to predict the effect of one variable on another by holding other variables constant; a predictive modelling technique that establishes a linear relationship between a dependent variable and one or more independent variables. A linear regression model can be used to predict the value of the dependent variable based on the values of the independent variables [98]. There are three different types of linear regression: simple, multiple, and multivariate linear regression. In the subsections below, these different types are described in more detail.

Simple Linear Regression
Simple linear regression is a model that uses one independent variable to predict one dependent variable. It is used to model the relationship between the dependent variable and independent variable by fitting a linear equation to the observed data. The equation for a simple linear regression model is expressed as [50]:

$$y_t = \beta_0 + \beta_1 x_t + \varepsilon_t \tag{4.6}$$

where y_t is the dependent variable or forecast variable, β_0 is the coefficient for the intercept, β_1 is the coefficient for the slope of the line, x_t is the independent variable or predictor variable, and ε_t is the deviation from the underlying straight line model (a Gaussian error term with N(0,σ^2). This technique assumes that the relationship between the dependent and independent variables is linear, and it uses the least squares method to estimate the parameters of the linear equation [50]. This technique can be used to forecast time-series data by predicting future values of the dependent variable based on past values of the independent variable.

To fit a simple linear regression model, the best-fit line that describes the relationship between the predictor (dependent) variable and the response (independent) variable needs to be determined. This is done by minimising the sum of the squared errors between the predicted values and the actual values of the response variable. Once the model is fitted, it is used to make predictions about the future values of the response variable based on the values of the predictor variable. Simple linear regression is a useful tool for time-series forecasting because it is simple to implement and can capture linear relationships between variables. However, it has some limitations, such as the assumption that the relationship between the predictor variable and the response variable is linear, which may not always be the case in real-world data. [26]

Multiple Linear Regression
Multiple linear regression is an extension of simple linear regression that allows for the prediction of one dependent variable based on the values of two or more independent variables. It is used for forecasting future values of a response variable based on current or past values of multiple explanatory variables. When it comes to forecasting short-term and mid-term electric load, this is the most widely applied technique [84]. The technique involves fitting a linear regression model to a set of data points that represent a time-series. The linear regression model is then used to predict future values of the response variable based on the explanatory variables. This technique is useful for forecasting trends in time-series data such as stock prices, interest rates, or economic indicators. It is also useful for predicting future values of the response variable given current or past values of the explanatory variables. It can be represented by the following formula [50]:

$$y_t = \beta_0 + \sum_{i=1}^{p} \beta_i x_{i,t} + \varepsilon_t \tag{4.7}$$

where the variables are the same as for simple linear regression with p predictor variables.

Multiple linear regression is a useful tool for time-series forecasting because it allows one to consider the influence of multiple predictor (independent) variables on the response (dependent) variable. However, it has some limitations, such as the assumption that the relationship between the predictor variables and the response variable is linear, which may not always be the case in real-world data. In addition, multiple linear regression can be sensitive to the inclusion of irrelevant predictor variables, which can affect the accuracy of the model. [50, 73]

Nearly all real-world regression models are multiple regression models [108]. Vislocky and Fritsch [106] compared the traditional multiple linear regression technique with Generalised Additive Models (GAM) for model output statistics forecasts of aviation weather parameters. The study's results show that traditional linear regression methods did not perform as well as the Generalised Additive Models (GAM) technique for the given data set, including variables, lead times, and seasons. This is likely due to the GAM technique's ability to automatically estimate the appropriate functional relationship for each predictor term in an additive model, whereas in linear regression, these relationships must be manually identified and computed or assumptions about linearity must be made. As described in subsection 4.3.4 for the ARIMA model, Li [64] proposed an ARIMA-regression model. To emphasise on the part played by the regression part, the regression technique used was multiple linear regression. The results show that the prediction accuracy of the multiple linear regression model is higher than for the ARIMA model. However, the ARIMA-regression combination shows better results.

Multivariate Linear Regression

Multivariate linear regression is often interchanged with multiple linear regression [45]. However, there are some differences. Multivariate linear regression is a type of statistical analysis used to predict the outcome of one or more dependent variables based on the values of multiple independent variables by fitting a linear equation to the observed data. It looks at the relationships between the different variables and how they can be used to predict the outcome of the dependent variable. This is then generalised to handle the prediction of several dependent variables. The independent variables can be lagged versions of the dependent variable, or other external variables that may have an effect on the dependent variable [45]. The model has the form of [43]:

$$y_{t,k} = \beta_{0,k} + \sum_{i=1}^{p} \beta_{i,k} x_{i,t} + \varepsilon_{t,k} \tag{4.8}$$

for $t \in \{1, ..., T\}$ and $k \in \{1, ..., m\}$ where $y_{t,k}$ is the k-th real-valued response for the t-th observation, $\beta_{0,k}$ is the intercept for the k-th response, $\beta_{i,k}$ is the i-the predictor's slope for the k-th response, $x_{i,t}$ is the i-th predictor for the t-th observation, and $\varepsilon_{t,k} \infty$ N($\mathbf{0}_m, \sum$) is a multivariate Gaussian error vector.

In a multivariate linear regression model, the dependent variable is modelled as a linear combination of the independent variables, with the coefficients representing the strength of the relationship between each independent variable and the dependent variable. The model is fit by minimising the residual sum of squares, which is the difference between the observed values of the dependent variable and the values predicted by the model. [43]

4.3.4. Box-Jenkins

The Box-Jenkins forecasting method is a statistical technique used to analyse and forecast time-series data that incorporates prior observations and a priori information. It involves the application of a series of steps, including model identification, parameter estimation, diagnostic checking and forecasting. The method is based on an iterative approach, which involves constructing a model for the series and then using the model to generate forecasts. This process is repeated until the model is refined to produce the best possible forecast. The Box-Jenkins method is based on the assumption that the data is generated by an underlying process with a stationary mean, variance and auto-correlation structure. The model is fitted to the data using least squares estimation and then forecasts are extrapolated from the fitted model. [80, 4, 14]

The Box-Jenkins method applies ARMA or ARIMA (Auto-Regressive Integrated Moving Average) forecast method is a statistical technique for time-series analysis that attempts to identify and quantify the underlying patterns within a time-series data set. It is a model-based approach that uses a combination of auto-regressive (AR) and moving average (MA) components to determine the underlying structure of the data and make predictions about future values. The ARIMA model is a linear combination of the AR and MA components, but can also include other components such as a seasonal component. The ARIMA model is used to forecast future values by taking into account the past values of the time-series data. The model can also be used to identify the likely sources of any underlying structure or patterns in the data, as well as to estimate the associated parameters that are needed to accurately forecast future values. To understand how the ARIMA model works, the model is stripped down and described below part by part. [80, 4, 14]

AR

The AR model uses a linear combination of past values of a time-series to predict future values. It is based on the assumption that the current value of a time-series is dependent on its past values. The model is denoted by AR(p), where the hyperparameter p represents the number of used lagged values. The model can be represented by an equation of the form [50]:

$$y_t = \beta + \sum_{i=1}^{p} \phi_i y_{t-i} + \varepsilon_t \tag{4.9}$$

where y_t is the current value of the time-series at time t, y_{t-1},y_{t-2}, ..., y_{t-p} are the past p values of the time-series, β is a constant, ϵ_t is a white noise term, and ε_1, ε_2, ..., ε_p are the auto-regressive coefficients. The value of p, which represents the number of past values used to predict the current value, is known as the order of the model. A higher value of p means that more past values are used, which can result in a more accurate prediction but also a more complex model.

MA

The MA model uses past forecast errors instead of past values of the forecast. As a result, future predictions are not based on what happened in the past, instead it is based on the error made in the past. The current value is estimated through a constant and a moving average of the residuals. The model, denoted by MA(q) where hyperparameter q represents the number of weighted moving average values of the past few forecast errors, can be represented by the following equation [50]:

$$y_t = \mu + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j} + \varepsilon_t \tag{4.10}$$

where y_t is the target value at time t, μ is a constant, θ_1, θ_2, ..., θ_q are the coefficients and ε_t is the residual at time t. The values of ε_t are not known. Unlike AR, MA is not a regression in the usual sense.

ARMA

ARMA is a direct combination of AR and MA, as explained by Korstanje (2021) [59], which can be clearly observed in the following equation:

$$y_t = c + \sum_{i=1}^{p} \phi_i y_{t-i} + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j} + \varepsilon_t \tag{4.11}$$

where c is a constant, y_t is the target value, and ε_t is the residual at time t. p and q are the same hyperparameters as described before, resulting in the following notation: ARMA(p, q). As a combination of the AR and MA models, the ARMA model predicts future values by both looking at the past values and the past errors. [59]

Many past studies and researches use ARMA together with another forecasting model and combine them in the end for a more sophisticated forecast. For example, Yunjian Jia et al. [55] combined a regular ARMA model with a Grey Model to forecast passenger flow. The results show that ARMA has a high precision and good performance for short-term predictions. However, for medium and long term forecast, ARMA is not suitable and the Grey Model is used. The combination of the two showed better results. Another example is the study of Gong [37]. Gong wrote a paper on forecasting passenger demand by using a technique where ARMA is combined with a General Regression Neural Network (GRNN). Arguments for this combination are that the determination of travel demand time-series generated from (non)linear processes is complex and secondly that it is not possible to accurately predict future time-series using societal phenomena, making the selection of an appropriate prediction technique intractable. Therefore, combining methods that can deal with linear or nonlinear processes successfully on their own, without being competent with complex reality situations, has become a common practice for improving performance and accuracy. The results of this paper proves the effectiveness of the technique.

ARIMA

When extending the ARMA model with an integrating component, it becomes the ARIMA model. This integrating component stands for automatic differencing of non-stationary time-series. Stationary is an important concept for time-series, for it has no long-term trend. The integrating component in the

ARIMA model applies differencing to make a non-stationary time-series stationary by replacing the actual values by the difference between the actual an previous value. This model can be denoted as ARIMA(p, d, q), where p and q are the same hyperparameters are described before and hyperparameter d stands for the order of differencing involved. It can be written as [50]:

$$y'_t = c + \sum_{i=1}^{p} \phi_i y'_{t-i} + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j} + \varepsilon_t \tag{4.12}$$

where y'_t is the differenced series, c is a constant, and the terms on the right side of the equation include both lagged values of y_t and lagged errors. It must be noted that it might be necessary to difference the time-series twice in order top become stationary and in rare cases where a second order differencing is not sufficient as well, a higher-order differencing is possible. The ARIMA model is primarily a powerful tool compared to the ARMA model for its ability to make non-stationary time-series stationary. However, the model is still not taking seasonality into account. Therefore, a seasonal component can be added. [50, 59]

In order to simplify Equation 4.12 and the upcoming equations, the lag operator L is introduced. The lag operator can be referred to as backshift operator and is defined as:

$$L^n y_t = y_{t-n} \quad \text{for} \quad t \geq n \tag{4.13}$$

The denotation of ARMA in Equation 4.11 can be rewritten to:

$$\begin{gathered} \left(1 - \sum_{i=1}^{p} \phi_i L^i\right) y_t = c + \left(1 + \sum_{j=1}^{q} \theta_j L^j\right) \varepsilon_t \\ \phi_p(L) y_t = c + \theta_q(L) \varepsilon_t \end{gathered} \tag{4.14}$$

where $\phi_p(L)$ represents the p-order polynomial $\left(1 - \sum_{i=1}^{p} \phi_i L^i\right)$ and $\theta_q(L)$ represents the q-order polynomial $\left(1 + \sum_{j=1}^{q} \theta_j L^j\right)$. For the ARIMA model, the differencing hyperparameter d is added. To difference a time-series, the difference between consecutive observations is computed as follows:

$$y'_t = y_t - y_{t-1} \quad \text{for} \quad t \geq 1 \tag{4.15}$$

In second order differencing, it is mathematically shown as:

$$\begin{aligned} y''_t &= y'_t - y'_{t-1} \\ &= y_t - 2y_{t-1} + y_{t-2} \quad \text{for} \quad t \geq 2 \end{aligned} \tag{4.16}$$

Rewriting the formula for the ARIMA model as given in Equation 4.12, results in the following:

$$\phi_p(L)(1 - L)^d y_t = c + \theta_q(L) \varepsilon_t \tag{4.17}$$

Tang and Deng [99] used the ARIMA model to develop the future trend of civil aviation passenger transport and to make reasonable predictions. The data that had been used was downloaded from the Civil Aviation Administration of China and was studied for the monthly data from January 2010 to august 2015. This data was used to predict the next 6 months in time. The results of this study showed that the ARIMA model was able to accurately predict the volume of passenger transportation per month with a prediction error maintained at about 1%. Li [64] proposed a combined forecasting method based on ARIMA-regression with an IOWHA operator concept to forecast civil aviation passenger volume in China. The findings were that the ARIMA model itself had an absolute percentage error of 3% between the predicted and actual values, the regression forecasting model obtained an error of 2.3%, and the combination of the two obtained a higher accuracy with an error of 2.1%. Showing that a combined forecasting model is, just like with the ARMA model, effective and reasonable.

SARIMA
As an extension of the ARIMA model, seasonality can be added. A seasonal ARIMA (SARIMA) includes seasonal terms to the ARIMA model. This makes the model very powerful for many cases of forecasting. As Korstanje (2021) [59] describes, it is the most complete model within uni-variate time-series modelling, using AR, MA, integration for modelling trends, and seasonality. Coming from the simplification as described above, the following equation defines the SARIMA model [59]:

$$\begin{gathered} y_t = u_t + \eta_t \\ \phi_p(L)\Phi_p\left(L^s\right)(1-L)^d(1-L^s)^D u_t = A(t) + \theta_q(L)\Theta_Q\left(L^s\right)\zeta_t \end{gathered} \tag{4.18}$$

where η_t is only applicable in case of measurement error, ϕ_p are the coefficients for the regular AR part, Φ_p are the coefficients for the seasonal AR part, A(t) is the trend polynomial (including the intercept) [104], θ_q are the coefficients for the regular MA part, Θ_Q are the coefficients for the seasonal MA part, d indicates the order for the regular integration part, D indicates the order for the seasonal integration part, and s is the coefficient of seasonality. Next to the known hyperparameters p, q, and d there are three more hyperparameters added: P, Q, and D. The seasonal period parameter s is not a hyperparameter. It is based on logic, where s is the number of observations per year. [59]

The paper of Tsui et al. [102] uses the seasonal ARIMA model to forecast airport passenger traffic for Hong Kong and projects its future growth trend to 2015 using monthly time-series data between January 1993 and November 2010. Empirical analysis revealed that the SARIMA model yielded accurate and reliable forecasting results as evidence by its lower Mean Absolute Percentage Error (MAPE) and Root Mean Squared Error (RMSE) values. Additionally, comparison of the actual and forecasted values revealed that the model produced acceptable forecast errors. A master's thesis performed by Bougas [13] studied the air passenger traffic flows in Canada by comparing multiple time-series forecasting models, among which the SARIMA model. All tested forecasting models provided accurate forecasts, however, in a comparison to the other models and in particular the ARIMA model, the results showed that the SARIMA model dominated in general.

In the study of Xu et al. [111] a SARIMA-SVR model is proposed to forecast the demand of the aviation industry. It is argued that many studies have been performed on forecasting with (S)AR(I)MA(X) models, however, it should be noted that these models are all based on the assumption of a linear relationship between the level of the time-series and its preceding data points. Whereas Artificial Intelligence (AI) models have been shown to possess exceptional capacity for identifying nonlinear patterns within data without the need for data transformation and assumptions on the underlying distributions. Therefore, a common approach is to construct a hybrid model that separately captures linear and nonlinear patterns and then integrating them to generate the final forecast. Some examples of this approach were given above of these combinations: Li [64], Yunjian Jia et al. [55], and Gong [37]. In another approach, previous research has utilised linear models to extract linear patterns and non-linear models to extract non-linear patterns from residuals, which are the differences between actual values and linear forecast results. Without going into depth, some examples of these studies are: Ruiz-Aguilar et al. [88] and Farajzadeh and Alizadeh [31]. Xu et al. propose several SARIMA-SVR models, which uses the SARIMA model for the inputs of the SVR model which output is the forecast. Additionally, Xu et al. look at the impact of bringing Gaussian White Noise into the forecasting models. The results show the SARIMA-SVR models are superior to the model of Ruiz-Aguilar et al., which was taken as benchmark.

SARIMAX
A final extension to the ARIMA model are external variables. By adding the X component (eXogenous factors) to the SARIMA model [104], this model becomes the most complete version of classical time-series models [59]. This model is very similar to the SARIMA model and is as follows:

$$\begin{gathered} y_t = \beta_t x_t + u_t \\ \phi_p(L)\Phi_p\left(L^s\right)(1-L)^d(1-L^s)^D u_t = A(t) + \theta_q(L)\Theta_Q\left(L^s\right)\zeta_t \end{gathered} \tag{4.19}$$

where β represents the external variable and the other variables are the same as described before. [59]

The study of Tsui and Balli [101] argues that external variables like destination marketing and tourism marketing are important factors in affecting the demand of international arrivals from foreign countries and thus for air passenger demand. In their study, the authors presented an accurate and reliable forecast of international passenger traffic for eight key Australian hub airports and conducted a comprehensive examination of the impact of external factors on passenger traffic. This study employed a SARIMAX model that incorporated five key explanatory variables: Gross Domestic Product (GDP) per capita in Australia, Tourism marketing expenditure by Australia's State Tourism Commissions, total scheduled international flight seats to Australian airports, fuel prices, and the exchange rate (AUD vs. USD). The study found that the selected best-fit SARIMAX model had higher R^2 values compared to the SARIMA model and demonstrated strong forecasting performance as evidenced by low values of MAPE, MAE, and RMSE. Additionally, all the selected best-fit SARIMAX models' residual series had the characteristics of white noise and did not exhibit the problem of serial correlation, indicating that the selected SARIMAX model was adequate for forecasting the monthly international passenger arrivals to the eight Australian airports.

Virate et al. [105] studied the effect of external variables on the demand of passengers of TransJakarta using the SARIMAX model to forecast this number. The external variables in this study are holidays and Eid holidays, during which usually the number of passengers drop. The forecast results prove that the SARIMAX model is quite accurate with small errors. A Degree Thesis performed by Salmi [93] looked at the impact of the COVID-19 pandemic on machine learning models in a commercial aviation use case. The objective of this thesis was to investigate the feasibility of developing models for forecasting airline passenger counts, to assess the decline in performance over time and to explore potential methods for improving the models. The forecasting techniques employed in the development of these models were Prophet and SARIMAX. Although the results were found to be incapable of adapting to changing circumstances, a comparison could be made between the models until the moment that the circumstances changed dramatically. It showed that both models are accurate predictors, where the SARIMAX model performs slightly better.

4.3.5. X-11

The X-11 forecasting method is a statistical technique used to analyse time-series data. It was developed by the U.S. Census Bureau and is used to identify and adjust for seasonal patterns and trends in the data. The X-11 method uses an iterative process to decompose the data into its component parts, including seasonal, trend, and irregular components. It then applies various smoothing techniques to the data to reduce the effect of outliers and other random variations. Finally, the X-11 method uses a "final" smoothing algorithm to produce a forecast based on the components. The X-11 method is a useful tool for analysing and forecasting time-series data sets and is often used in economics and finance. [50]

4.3.6. Machine Learning Methods

Forecasting the demand for checked baggage items is a crucial task for many airports around the world, particularly for Schiphol Airport. This is because the passenger demand for these items can vary greatly depending on the season, the type of traveller, and the weather conditions. Accurate and precise demand forecasting enables airports to better manage their resources, resulting in an improved customer experience and enhanced profitability. Over the years, Schiphol has experienced that the behaviour of people has changed when it comes to choosing whether to check in baggage or take it with them on the plane as hand luggage. Some reasons for this could be the fact that from 2017 on wards more airlines started charging checked baggage for passengers. Another factor could be the COVID-19 pandemic, of which the impact on the human behaviour is unknown at this moment.

To meet these forecasting needs, several traditional forecasting methods have been proposed in the past, such as ARIMA models and ES models as described above. However, these models are limited in their ability to capture the complex patterns of passenger demand and are unable to adapt to the changing dynamics of the demand. In recent years, machine learning and deep learning techniques, such as Neural Networks (NN), have emerged as a powerful tool for forecasting. These techniques have the ability to capture long-term dependencies in data and their capacity to learn from complex patterns. [72]

Artificial Neural Network
It has been shown that Artificial Neural Networks (ANNs) possess the capability to approximate any continuous function. Furthermore, ANNs have been successfully utilised for forecasting financial data series. Traditional time-series prediction methods such as Box-Jenkins, ARMA or ARIMA rely on the assumption of a linear relationship between inputs and outputs. ANNs, on the other hand, possess the advantage of being able to approximate any nonlinear functions without the need for prior information about the properties of the data series. [83]

ANNs are a type of machine learning algorithm inspired by the structure and function of the human brain. They are composed of interconnected units called ”neurons,” which are organised into layers. These layers are composed of input, hidden, and output layers, and the connections between the neurons are called ”weights”. In the context of forecasting, ANNs can be used to make predictions about future values in a series based on historical data. Kanavos et al. [57] made a comparison between the ARIMA and SARIMA models and a Deep Learning Neural Network (DLNN), which is a specific type of ANN that is composed of multiple layers of interconnected neurons, for forecasting air passenger demand for multiple airports in the USA. The results show that, as would be expected, the SARIMA model has a higher accuracy on seasonal data compared to the ARIMA model and that the DLNN outperforms the two forecasting models for the particular aviation data. Therefore, a DLNN can be proposed as an effective method for forecasting aviation demand in comparison to traditional time-series models such as ARIMA or SARIMA.

Multilayer Perceptron
One type of ANN commonly used for the forecasting task is the multilayer perceptron (MLP). The MLP is composed of multiple layers of neurons, with the input layer receiving the input data and the output layer producing the final prediction. The hidden layers between the input and output layers process and transform the input data through the use of weights, which are adjusted during the training process to optimise the prediction accuracy of the model. The training process involves feeding the model a set of input-output pairs, called the training data, and adjusting the weights to minimise the error between the predicted output and the true output. This process is known as backpropagation and is typically done using an optimisation algorithm such as stochastic gradient descent. [115, 24]

In general, the use of ANNs, particularly the MLP, for forecasting allows for the modelling of complex nonlinear relationships in the data and can provide more accurate predictions compared to simpler methods such as MA or linear regression [115, 24]. In the study of Şahin et al. [91], different ANN forecasting methods are compared to Croston Based methods [25] for forecasting aviation spare parts demand. The ANN models used in this study were the MLP, Recurrent Neural Network, Time-Delay Network, and Radial Basis Function. Different types of demand data was used to test the forecasting performance of the models: intermittent, erratic, and lumpy. The results showed that for the ANN models, the MLP outperformed the other three models among all demand data types. Compared to the Croston based methods, the MLP showed superior results for intermittent and lumpy demand type data. For the erratic demand data type, the Leven & Segerstedt [25] method had the best performance, however, the performance of the MLP for this type is similar with the Croston method slightly better. The study shows that the MLP is an effective method for forecasting different types of demand data. In another study of Blinova [12], a time-lagged feedforward network embodied by MLP is used for forecasting passenger traffic flows in Russia. The results present a satisfactory result for the forecast up till a maximum of three years. The relative forecast error, at the stage of adaptation, for the neural network generated is less than 5%. The study thus concludes that the model can be proposed for short-term forecasts. Amal and Ammar [3] presented a paper with an overview of deep learning types for forecasting time-series. In the overview, the MLP performance is compared to other models for forecasting challenges in other industries than the aviation industry and with different types of data. It was shown that the MLP performes best in the studies of Alhassan et al. [2] and Di Persio and Hornchar [85]. Other studies that used MLP of Lago J. et al. [62] and Hernández et al. [44] show that the MLP is a good method for forecasting, however, other methods or adjusted methods have better performance.

Recurrent Neural Network

Recurrent Neural Networks (RNNs) are a type of ANN that can process sequences of data, such as time-series. RNNs are typically composed of a set of neurons or nodes that are connected in a loop or recurrent structure. Each neuron takes an input and produces an output, and the output of one neuron is fed as input to the next neuron, allowing the network to memorise, or “learn”, information over time. This recurrent structure enables the network to capture temporal patterns and make predictions about future values. [38, 86]

RNNs can be used for a variety of tasks, including forecasting. In forecasting, the RNN is trained using historical data, and then the network is used to make predictions about future values. This is done by taking the most recent input values and using them to compute the outputs of the neurons in the network. The outputs are then combined to obtain the predicted value. This process is repeated for each time step in the, allowing the network to make predictions about future values, in for example a time-series data set. RNNs have a variety of architectures, with the most popular being the long short-term memory (LSTM) and gated recurrent unit (GRU) networks. These architectures use gates to control the flow of information into and out of the memory, allowing the network to selectively retain or forget information as needed. Like with any other NN, RNNs are trained with a lot of data and good pre-processing. [86, 34]

Long Short-Term Memory

A type of RNN that are well-suited for modelling data are LSTM networks introduced by Hochreiter and Schmidhuber [47]. LSTMs are a specific type of RNN that are particularly effective at modelling long-term dependencies in the data. They do this by using a special type of neuron called a ”memory cell” that can store information over long periods of time. LSTMs also have three special types of gates (input, output, and forget gates) that control the flow of information into and out of the memory cell, allowing them to selectively remember or forget past information as needed. To use an LSTM for forecasting, the data is typically divided into training and test sets, with the model being trained on the training set and the forecast performance being evaluated on the test set. The model can then be fine-tuned by adjusting the hyper-parameters, such as the number of layers in the network or the size of the hidden state vectors. [47, 38]

One advantage of using an LSTM for forecasting is that it has the capacity to learn from complex patterns and can handle long-term dependencies and seasonality in data. However, it can be more computationally intensive to train than some other machine learning algorithms, and it can be difficult to interpret the internal workings of the model. [47, 38]

Zheng et al. [116] explored the LSTM-based RNN to forecast electric load in smart grids. To guarantee fair comparison of the performance, next to the electric load data set, also an airline passenger data set was used. The results showed for both data sets that the LSTM, together with a SARIMA model, was superior to other methods. Concluding that LSTM is capable of forecasting complex univariate time-series or other more complex data sets. Although in this case the SARIMA model did outperform the LSTM model. The cause for the observed phenomenon is attributed to the presence of a robust multiplicative seasonal component and a distinct upward trend in the airline passenger data set. The SARIMA model effectively utilised these consistent seasonal patterns by utilising logarithmic transformation and seasonal differencing techniques. The utilisation of specialised, data-dependent pre-processing techniques facilitated the forecasting capabilities of the SARIMA model, however, it should be noted that the efficacy of these techniques is highly dependent on the specific characteristics of the data set and the LSTM model, which does not require such pre-processing, may be a more general solution for forecasting. Choi and Kim [22] proposed MLP, RNN, and LSTM for the prediction of airport capacity. The results showed that all three models showed good performance where the RNN and LSTM models outperformed the MLP model. However, the models were trained with and validated against Hartsfield-Jackson Atlanta International Airport data. When examining the transferability to another airport, the MLP model demonstrated strong transferability without the need for additional techniques, while the RNN and LSTM models were able to accurately predict the capacity of another airport following fine-tuning. Liu et al. [69] compared the LSTM RNN with other traditional prediction models like ARIMA and GRNN. The results of the proposed method based on LSTM exhibits several advantages over traditional statistical prediction methods. Specifically, the LSTM RNN is capable of fitting a wider range of data patterns. Additionally, the modelling process utilising LSTM RNNs is less time-consuming and does not require manual steps, such as stability checking, auto-correlation function checking, and partial auto-correlation function checking. Furthermore, after proper training, the LSTM RNN model demonstrates a higher level of predictive accuracy. However, traditional statistical-based models still possess certain advantages such as reduced resource consumption and faster forecasting speed.

Gated Recurrent Units

Gated Recurrent Units are a type of RNN architecture based on the concept of memory gates. They are well-suited to forecasting due to their ability to capture important information from the past and use it to make predictions. They are simpler to train than LSTM networks and often require less training data. The main advantage of GRUs is that they are more efficient than LSTM networks and can be trained faster. However, they may not be able to capture long-term dependencies as well as LSTM networks. [38, 86]

Yu [113] proposed a method for airline passenger flow prediction where first a Deep Belief Network reduces the dimension of the data, secondly a GRU model extracts features, and finally an attention mechanism preserves key features for a high-precision prediction. When comparing the MSE to other models, the proposed GRU model results are significantly better than the other models. In the study of Wang et al. [109], a Bi-directional GRU model is proposed for traffic flow prediction. In comparison to GRUs, which are capable of retaining information from previous sequences, this model exhibits the ability to retain traffic flow information from both previous and subsequent sequences. The Bi-GRU model is compared to the common GRU and three other benchmark models: ARIMA, LSTM, and Bi-LSTM. The results show that the performance of both bi-directional models are better compared to the normal LSTM and GRU, where the Bi-GRU is slightly better than the Bi-LSTM and the GRU is slightly better than the LSTM. The performance of the ARIMA model is the worst.

Convolutional Neural Network

Convolutional neural networks (CNNs) are a type of NN that are particularly effective at processing data with a grid-like technology, which are commonly used for image and video processing. CNNs are able to learn features from the data automatically, which makes them particularly useful for tasks like image classification and object detection. However, they can also be used for forecasting. CNNs are able to extract local patterns and trends from data and use them to make predictions. For example, CNNs can learn to identify patterns in the data that are indicative of future trends, such as seasonal patterns or changes in the level of the data. To use a CNN for forecasting, the data is typically transformed into a 2D matrix, with each time step in the series represented as a row in the matrix. The CNN is then trained to predict the next time step in the series given the past time steps as input. [38, 17]

For example, consider a time-series of temperature data, with one record for each day. The CNN might be trained to predict the temperature for the next day given the temperature data for the past week. To do this, the temperature data for the past week would be transformed into a 2D matrix, with each row representing one day and the columns representing the different time steps within a day. The CNN would then be trained to predict the temperature for the next day based on the patterns it learned in the past week's temperature data.

An advantage of CNNs is that they are able to handle high-dimensional data and can identify complex patterns in the data. Another advantage is that CNNs can be trained relatively quickly and require less training data compared to other NNs. However, they may not be as well-suited to long-term forecasting as RNNs, because they do not have the ability to maintain internal state over long periods of time. [17] Mehtab and Sen [74] designed multiple CNN and LSTM models to forecast financial time-series. The models are evaluated and ranked based on two performance metrics: execution time and the ratio of the RMSE to the mean open value in the test data set. The results indicate that onf of the CNN models is the fastest in execution and possesses the highest level of forecasting accuracy. In general, it is observed that the CNN models exhibit faster execution times compared to their corresponding LSTM counterparts. In an earlier study of Mehtab and Sen [75], a CNN model was compared to many other classification models for stock price prediction. Based on the RMSE, the CNN models were clearly superior to the other models and proved the capability of short-term forecasting of CNNs.

Random Forest
Random Forests (RF) are a type of ensemble machine learning algorithm that can be used for a variety of tasks, including forecasting. Ensemble algorithms combine the predictions of multiple individual models to produce a more accurate and stable prediction. In the case of RF for forecasting, the algorithm works by training a large number of decision trees on different subsets of the data, and then averaging their predictions to produce a final forecast. It is called ”random” because each decision tree is trained on a random subset of the data, and the final prediction is made by taking the average of the predictions of all the trees. [18, 16, 36]

One advantage of using RF for forecasting is that it can handle high-dimensional data and nonlinear relationships between features and the target variable. It is also resistant to over-fitting, which can be a problem with some other machine learning algorithms when working with large data sets. Just as for the other machine learning methods described above, the data for a RF forecast is typically divided into training and test sets, with the model being trained on the training set and the forecast performance being evaluated on the test set. The model can then be fine-tuned by adjusting the hyper-parameters, such as the number of trees in the forest or the maximum depth of the trees. [36, 18]

Lin and Tian [66] used RF, LSTM, and RF and LSTM combined model to predict short-term metro passenger flow. The combined model utilises RF for a feature importance screening to initially evaluate the significance of all extracted features. Subsequently, features with lower impact on the prediction outcome, as determined by the feature importance ranking, are filtered out. Finally, a LSTM model is utilised for prediction. The results were analysed with the mean absolute error (MAE) and showed that the combined model has the highest accuracy, followed by the LSTM model, and the lowest accuracy for the RF model. Tyralis and Papacharalampous [103] conducted an extensive set of computational experiments in time-series forecasting using RF and other forecasting methods. Empirical evidence has demonstrated that utilising a limited number of recently lagged predictor variables in RF models yields superior performance. This phenomenon can be attributed to the reduction in training set length and subsequently the amount of information extracted from the original time-series during model fitting, when an increasing number of lagged variables are utilised.

4.3.7. Gradient Boosting

Another decision-tree method is Gradient Boosting. Just as with RF, a gradient boost model uses numerous small decision trees and makes its predictions based on them. This method involves combining multiple weak predictive models (the trees) into an ensemble that makes a more accurate prediction. Given a set of input features and a corresponding set of output values, the algorithm first trains a ”weak” decision tree, to predict the output values from the input features. This model is called the base learner. The algorithm then calculates the errors or residuals between the predicted values and the actual values. The next step is to train a second model to predict these errors, rather than the output values themselves. This second model is also called the base learner, and is trained using the residuals as the output values and the same input features as the first model. The algorithm then combines the predictions from the two base learners to produce a more accurate ensemble model. This is an iterative process that represents that is called *boosting*. The *gradient* part represents the part of the method that involves optimising a loss function by minimising its gradient or slope. In other words, the algorithm tries to find the direction in which the loss function decreases the fastest, and updates the model in that direction. The gradient boosting process is iterative with each new model trained to reduce the residual error of the previous ensemble. [59, 10]

The current passenger forecast model of Schiphol Airport is based on this method. Its ability to produce highly accurate predictions in a relatively fast way gives advantages over more complex models like neural networks. Moreover, gradient boosting is able to provide insights into which features are most important for making predictions, which allows for better interpretability. [10]

4.3.8. Prophet

The Prophet model, developed by Facebook's Core Data Science team in 2017, is a powerful and accurate forecasting tool for time-series data. It is based on a decomposable additive approach, meaning that it uses non-linear trends to capture seasonality, holidays components, and other effects in the data to produce forecasts. The model utilises a Bayesian structural time-series model to account for the seasonality in time-series data and to search for the best combination of parameters for the model that minimise the loss function and maximise the accuracy of the predictions. Additionally, the Prophet model allows for the inclusion of user-defined holidays and other user-defined regressors. This approach to forecasting is both automated and highly flexible, allowing users to tailor the forecasting process to their specific needs. [100]

The Prophet model works by taking into account a variety of factors to make its predictions. Firstly, it takes into account the trend of the data. It uses a piecewise linear or logistic model to fit non-linear trends in the data. Additionally, Prophet takes into account seasonality in the data by identifying trends that repeat over time, such as weekly or monthly patterns. It also allows for the user to incorporate additional effects such as holidays and user-defined variables. It then combines these factors to build its prediction model. Finally, Prophet uses an additive regression model to fit the data and forecast future values. The model includes a baseline adjustment, which is a linear or logistic regression model that captures the long-term trends in the data. It also includes a seasonal component which captures the short-term trends. Together, these components capture the patterns in the data and allow for accurate forecasting. The Prophet model has been found to be highly accurate in a variety of forecasting scenarios, with the capability to accurately predict up to 18 months in advance. Furthermore, the model is exceptionally fast, with the ability to generate forecasts in less than a second. This makes it an ideal choice for quick and efficient forecasting of time-series data. [100]

In summary, the Prophet model uses a combination of non-linear trends, seasonality, and user-defined effects to build an accurate model for forecasting time-series data. Moreover, the Prophet model is a powerful and accurate forecasting tool for time-series data. It is highly flexible and fast, allowing users to quickly generate accurate forecasts for a variety of scenarios. Satrio et al. [94] used the Prophet model for forecasting the coronavirus disease in Indonesia. The evaluation of forecast data generated by the Prophet model indicates a high degree of accuracy in the initial stages, with minimal deviation from the actual data. However, as the forecast horizon extends, the discrepancy between the forecasted and actual data increases, resulting in a noticeable divergence. In contrast, the ARIMA model that was used for the same goal demonstrates lower accuracy compared to Prophet throughout the entire forecast period.

Navratil and Kolkova [79] subjected Prophet to further research and concluded from the results of their study that the model is relatively simple to interpret and meets the requirements of forecasting accuracy. Although, it is proposed that incorporating elements of artificial intelligence or multivariate analysis within the Prophet model would enhance its versatility and applicability to a broader range of business entities. An example of incorporating machine learning with the Prophet model is demonstrated by Guo et al. [40] for product demand forecasting. The proposed hybrid approach combines the Prophet model for forecasting seasonal fluctuations and identifying input variables for the Support Vector Regression (SVR) model, with the SVR model utilised to capture non-linear patterns in the data. An evaluation of eight models revealed that the hybrid Prophet-SVR approach demonstrates superior performance. Based on various statistical indicators, the Prophet-SVR model consistently produces the most accurate forecasting results in comparison to the other models. Additionally, the Prophet-SVR model is found to accurately forecast both the trends and fluctuations in the data. When comparing the Prophet model to the SVR model, it showed better seasonal forecasting of the Prophet model. However, the MAPE of the SVR model is generally lower than that of the Prophet model, which suggests that SVR outperforms Prophet in trend prediction. The paper argues that this result may be attributed to the fact that the Prophet model utilises a traditional time-series analysis modelling strategy, which may result in underfitting during the model training process and impede the ability to learn complex patterns. The paper proves the potential of combining machine learning with the Prophet model.

5

Resource Allocation Optimisation Baggage Handling Systems

There are several optimisation methods and algorithms that are commonly used for resource allocation optimisation. In this literature study, two different kind of optimisation methods are considered: exact solution methods and meta-heuristic methods. Exact solution methods, which are described in section 5.1, are optimisation techniques that are guaranteed to find the global optimal solution to a problem, given enough computational resources and time. These methods are generally more accurate and reliable than meta-heuristic methods, but they can be computationally expensive and may not be able to find solutions in a reasonable amount of time for large or complex problems. Meta-heuristic methods, on the other hand, are optimisation techniques that do not guarantee finding the global optimal solution, but are able to find good solutions in a relatively short amount of time. These methods, described in section 5.2, are based on the imitation of natural processes and are inspired by natural phenomena such as evolution, physics, and chemistry. These methods are generally less accurate than exact solution methods, but they are able to find solutions quickly for large or complex problems, and also they can be applied to problems where the mathematical models are not well known or hard to define.

5.1. Exact Solution Methods

There are several exact solution methods that can be used for resource allocation optimisation. The most common ones are described in this section. Other exact solution methods that can be looked at when the described methods are not sufficient are: Nonlinear programming, Mixed Integer programming, and Constraint programming.

5.1.1. Linear Programming

Linear programming (LP) is a method for optimisation that can be used to allocate resources in an efficient manner. It is based on the idea of finding the best solution to a problem by minimising or maximising a linear objective function subject to a set of linear constraints [42]. The method involves the use of a mathematical model that describes the problem, with variables representing the resources to be allocated and the objective function representing the goal of the optimisation. The constraints are used to represent the limitations on the resources, such as availability or budget. [27]

Finding the solution is typically done using a technique called simplex algorithm, which is an iterative method for solving LP problems [15]. The simplex algorithm works by starting with a feasible solution and then repeatedly moving to adjacent solutions that improve the objective function until an optimal solution is found. The algorithm terminates when no further improvement can be made, which means the optimal solution has been found. [27, 42, 110]

LP is a powerful tool for resource allocation scheduling and has many practical applications, including in operations research, finance, and logistics. Al-Rawi and Mukherjee [87] studies a method for solving labour scheduling problems encountered in a construction company by applying LP techniques. With the set up of the right objective function, constraints, and decision variables, the LP model showed a solution which maximised the fairness of the schedule while considering all constraints of efficient time and effort management and workload balance. The result provided a more contended and effective outcome. Another study conducted by Zhang et al. [114] utilised an LP model to minimise costs in a container oriented job scheduling. It showed good results which lowered the total costs significantly.

5.1.2. Integer Programming

Integer Programming (IP) is an extension of LP that allows for integer variables in the optimisation problem. IP can be used to solve problems where the resource allocation variables must be integers, such as scheduling problems with distinct time periods or problems with binary decision variables. LP is a technique for the optimisation of a linear objective function, subject to constraints represented by linear equations or inequalities [15]. The variables in LP are continuous, meaning they can take on any real value. However, in many practical problems, the variables must take on integer values, which makes LP inapplicable. In IP, the decision variables are restricted to take on integer values. This restriction can be imposed by adding integrality constraints to the problem, which forces the decision variables to take on integer values. IP can be used to solve problems where the resource allocation variables must be integers, such as scheduling problems with distinct time periods or problems with binary decision variables. [5, 51]

IP can be used to solve a wide range of problems in various fields such as transportation, production, logistics, and scheduling. For example, in scheduling, IP can be used to solve problems such as crew scheduling, shift planning, workforce planning, and skill-based routing [15]. In these problems, the decision variables are often binary, indicating whether a certain resource is assigned to a certain task or not. A disadvantage of IP is that is an NP-hard problem, which makes it computationally expensive to solve [51].

In the research of Liu et al. [68], a mathematical model based on IP is developed to quantify the overall baggage handling time in a collaborative work system, utilising piece wise functions for various allocation strategies. The model incorporates optimisation of the queue system to adhere to passengers' wait time expectations, ultimately leading to a significant enhancement in system efficiency. In the study of Ip et al. [53], an IP approach is proposed for the optimisation of aircraft maintenance planning and scheduling. The primary objective of this research is to improve the efficiency and effectiveness of the planning and scheduling processes. Additionally, a set of computational schedules for maintenance manpower is developed to accommodate all scheduled flights. The proposed methodology utilises a balance of workloads among groups and fulfilment of constraints through rounding, resulting in an optimal solution while maintaining computationally feasible time. Furthermore, the results obtained are presented in a clear and easily understandable format and the methodology is designed to minimise complexity and simplify the process.

5.1.3. Mixed Integer Linear Programming

Mixed Integer Linear Programming (MILP) is a type of optimisation problem that combines elements of both LP and IP. In MILP, some of the variables are restricted to take on integer values, while the remaining variables are allowed to take on any value within a specified range. The objective function and constraints are all linear. MILP can be used to solve a wide range of resource allocation problems, such as production scheduling, project management, and financial planning, where the decision variables must take on integer values. It can also handle problems that have both discrete and continuous variables. MILP is considered to be an exact solution method as it can provide an optimal solution that satisfies all the constraints, and it can handle a wide range of problems. [33]

MILP can be solved using specialised algorithms and software, such as branch-and-bound algorithm, that is designed specifically for this type of problem [9]. Commercial solvers such as Gurobi, CPLEX, and Xpress, are popularly used to solve MILP problems. [33]

The paper of Wahyudin et al. [107], presents a MILP model for aircraft line maintenance resource allocation optimisation, focused on the allocation of workforce, material, and tools. It is mentioned that many previous studies do not take the relationship of a hub and spokes station in terms of resource transfer into consideration, which in this case is taken into consideration. The paper shows the capability of the MILP model to find an optimal solution for allocating workforce, material, and tools. Another example of a research that utilised MILP is the study of Kuhn and Loth [61]. This study examines algorithms for scheduling airport service vehicles. A MILP is proposed to minimise fuel costs for service providers and delays for air carriers. The IP problem is formulated to facilitate solution search strategies. A genetic algorithm heuristic, borrowed from aircraft arrival scheduling, is implemented to find approximate solutions efficiently, in addition to an exact solution method that utilises branch and bound techniques specifically tailored for this problem. The results indicate that planning service vehicle routes based on future demands rather than reacting to them as they occur leads to significant benefits. The proposed method can reduce both the delay absorbed by aircraft and the distances travelled by service vehicles by 20% or more.

Branch and Bound

Branch and bound is a method that can be used to solve MILP problems. MILP problems are computationally expensive to find an exact solution. Branch and bound is a technique that can be used to find an exact solution to MILP problems in a more efficient way than other methods such as an exhaustive search. The basic idea behind branch and bound is to divide the problem into smaller sub-problems by introducing new constraints to the problem and then solving each sub-problem separately [63]. The method starts by relaxing the integrality constraints of the problem, and solving the resulting LP problem to get an initial solution. Then, the method generates new constraints by branching on the fractional variables, which are the variables that have fractional values in the LP solution. Each new constraint creates a new sub-problem, and each sub-problem is solved using LP. The solution of each sub-problem is then compared to the current best known solution and if it is better, it becomes the new best solution. The method continues to branch and solve sub-problems until an optimal integer solution is found. The method also uses a bounding function to prune the search tree by eliminating sub-problems that are guaranteed not to improve the current best solution. [46]

An example of using Brand and Bound method to solve a MILP problem, is the Parallel Machine Scheduling (PMS) model [92]. PMS is a type of exact solution model that can be used for resource allocation optimisation, specifically for scheduling problems involving the use of multiple machines or resources. The method is based on mathematical models and algorithms that are designed to find the optimal schedule for a set of jobs or tasks on a set of parallel machines, given a set of constraints and an objective function. The objective of PMS is to assign the tasks to the machines in such a way that certain criteria, such as minimising the make-span (the completion time of the last task), minimising the total completion time, or maximising the total throughput, are met [78]. PMS can be formulated as an optimisation problem, and several of the methods mentioned above can be used to solve it. For example, LP can be used to formulate and solve parallel machine scheduling problems where the objective function is linear and the constraints are also linear. IP can be used when the number of machines is restricted to integers. However, also approximate (meta-heuristic) methods can be used for solving these models. In the context of airport scheduling, the assignment of scheduling planes to gates can be conceptualised as a PMS problem, where planes represent jobs and gates represent machines. The feasibility of assigning a particular job (i.e., a plane) to a specific machine (i.e., a gate) is dependent on the size of the job and the capacity of the machines. Specifically, gates may have a maximum capacity that cannot accommodate larger-sized jobs (or in the airport case, planes of larger sizes) [92]. In the case of the current research, this model could be used to optimally allocate human resources from service companies, e.g. (external) handlers, where the checked baggage on laterals correspond to jobs and the handlers correspond to machines.

5.1.4. Dynamic programming

Dynamic Programming (DP) is a mathematical optimisation method that is used to solve problems with a recursive structure and overlapping sub-problems. The method is based on the idea of breaking a complex problem into smaller sub-problems, and solving each sub-problem separately. The solutions to the sub-problems are then combined to give a solution to the original problem [46]. The key idea behind DP is to store the solutions to the sub-problems in a table so that they can be reused, rather than recomputing them each time they are needed. This approach is called memorisation. DP can be used to solve problems with recursive dependencies, such as resource allocation in networks, and problems with a temporal aspect, such as the scheduling of resources over time. These types of problems can be modelled as a sequence of states, each state representing the current status of the system. The problem is then to find the optimal sequence of states that leads to the desired final state. [15, 46]

The paper of Dell'Olmo and Lulli [28] presents a novel approach to addressing the issue of airport capacity. A DP formulation and a corresponding backward solution algorithm are proposed, which are simple, robust, and efficient. The performance of the algorithm is evaluated on various realistic scenarios and compared to a commonly used greedy decision policy. The deviation between the greedy and DP optimal solutions ranges from 3.98% to 35.64%, indicating a significant improvement in almost all instances.

5.2. Meta-heuristic Methods

Meta-heuristic methods are a class of optimization methods that use high-level, problem-independent strategies to find approximate solutions to complex optimization problems. They are called ”meta” because they work on top of other optimization methods, guiding and enhancing the optimization process. The most common methods are described in the sub sections below. Other methods are: Tabu Search, Simplex search, Particle Swarm Optimisation, and Large Neighbourhood search.

5.2.1. Genetic Algorithm

The concept of the Genetic Algorithm (GA) is rooted in the principles of evolution as outlined by Charles Darwin, and is among a category of computational techniques referred to as evolutionary computation. The algorithm was first formulated by John Holland in the 1960s [48]. GA is a heuristic optimisation method that is based on the concepts of natural selection and genetics. It is a meta-heuristic method, meaning that it is a general-purpose optimisation technique that can be applied to a wide range of optimisation problems. GA works by creating a population of potential solutions, called individuals, to the optimisation problem. Each individual is represented by a set of parameters, called chromosomes, that define the solution. The chromosomes are usually encoded as a binary string, but other encoding such as real numbers or permutations can also be used [77]. The GA then iteratively combines, mutates, and selects the best individuals to create new generations of solutions. The combination of individuals, called crossover, is performed by selecting two parent individuals and creating two offspring by combining their chromosomes at a randomly selected point. The mutation operator is used to introduce small random changes to the chromosomes of the individuals. The selection operator is used to select the individuals that will be used as parents for the next generation. The selection is usually performed based on the fitness of the individuals, which is a measure of the quality of the solution represented by the individual. The GA continues to evolve the population for a fixed number of generations or until a stopping criterion is met, such as reaching a certain level of fitness. The best individual in the final population is the solution to the optimisation problem. [58, 46]

The research of Shiu and Szeto [97] presents a new type of adaptive evolutionary algorithm that combines two GAs using a mutation matrix, which is based on an adaptive allocation of CPU time. The algorithm is evaluated on the airport scheduling problem with constraints. The results of the application to airport scheduling show that the self-adaptive mutation-only GA is able to efficiently produce high-quality solutions. Liu et al. [67] examined the use of a GA in the scheduling of airport terminal area traffic. The research combines GA theory with analysis of airport terminal area traffic and focuses on the design and implementation of a traffic scheduling algorithm module for a GA-based traffic scheduling system. The study found that using the GA to optimise for the shortest total flight landing time alone is an oversimplified goal, and an adaptive GA was used to address the limitations of the GA such as instability and prematureness. The study was implemented using the platform support of the national ATC scenario simulation system and the improved sorting algorithm was applied. The results show that the improvement of the algorithm is effective and achieves the research objectives. Guo et al. [41] proposed a novel GA for airport baggage transport vehicle scheduling to improve customer service and provide efficiency and safety for airport operations. By considering population diversity and population fitness simultaneously the proposed GA considers both exploitation and exploration abilities. Real data is used in the proposed algorithm to evaluate its effectiveness and feasibility. The simulation results indicate that the proposed algorithm is able to achieve competitive performance in addressing the airport baggage transport vehicle scheduling problem.

5.2.2. Simulated Annealing

Simulated Annealing (SA) is a meta-heuristic optimisation method that is based on the concept of annealing in metallurgy. The method simulates the process of heating and cooling a material to find the optimal solution [90]. SA is often used to solve complex scheduling and resource allocation problems, and it can find near-optimal solutions. The basic idea behind SA is to generate a sequence of solutions to the optimisation problem, where each solution is generated by making a small random change to the current solution. The new solution is accepted if it improves the objective function, and rejected otherwise. However, SA also allows for the acceptance of worse solutions with a certain probability, which is called the acceptance probability. The acceptance probability is determined by a temperature parameter, which is gradually decreased during the optimisation process. The temperature parameter controls the randomness of the search and is used to balance the exploration and exploitation of the search space. At high temperatures, the acceptance probability is high, and the search is more random, allowing for the exploration of the entire search space. At low temperatures, the acceptance probability is low, and the search is more deterministic, allowing for the exploitation of the best solutions found so far. [90, 60]

The main advantage of SA is its ability to escape from local optima and to find near-optimal solutions [60]. SA can also handle problems with constraints and problems with non-differentiable or non-continuous objective functions. However, SA also has some limitations, such as the difficulty of specifying the appropriate parameters for the algorithm, such as the cooling schedule and the initial temperature. [90]

In the study of Yan and Shi [112], an optimisation approach to the operating efficiency of baggage turntables for arriving passengers is proposed by utilising the SA algorithm in the allocation of baggage claim carousels. Simulation results demonstrate that the proposed algorithm yields improved performance compared to the first-come-first-served distribution method. The results indicate that the SA algorithm is a viable solution for addressing the problem of baggage claim carousel allocation and improving operational efficiency.

5.2.3. Ant Colony Optimisation

Ant Colony Optimisation (ACO) is a meta-heuristic optimisation method that is also inspired by nature by observing the behaviour of ant colonies [11]. The algorithm simulates the behaviour of ants as they search for food, and uses this behaviour to find an optimal solution to a problem. In ACO, a set of virtual ”ants” move through the solution space, each leaving a trail of ”pheromones” behind. The strength of the pheromone trail represents the quality of a particular solution. As the ants move through the solution space, they update the pheromone trails based on the quality of the solutions they find. Over time, the pheromone trails converge on the best solutions, and the ants are able to find the optimal solution more efficiently. [76]

ACO can be applied to resource allocation optimisation problems by representing each resource as a node in a graph, and the solution as a path through the graph. The ants move through the graph, selecting the next node to visit based on the pheromone trails and a probability that is influenced by the quality of the resource. The pheromone trails are updated based on the quality of the solutions found by the ants. ACO is a powerful optimisation method that can handle large-scale, real-life problems and can take into account the dynamic nature of airport operations. It is particularly useful for routing and scheduling problems, which are common in airport operations.[29]

An improved ACO algorithm is proposed by Du et al. [30] to optimise airport ground service scheduling. In this study, a multi-objective model for vehicle routing problem with tight time windows, short travel time and re-used vehicles is presented. The model aims to minimise the number of vehicles used, the total start time of serving flights, and the total flow time of vehicles. An ACO algorithm is proposed as a solution method for this discrete optimisation problem due to its ability to efficiently handle multi-objective problems. The results of numerical computations demonstrate that the proposed ACO algorithm is able to construct high-quality solutions in a reasonable time frame.

References

[1] Adetayo Adeniran, Olufunto Kanyio, and Adelanke Owoeye. “Forecasting Methods for Domestic Air Passenger Demand in Nigeria”. In: *JARIE* (2018). DOI: 10.22105/jarie.2018.133561.1038. URL: https://www.researchgate.net/publication/329236776.

[2] Zakhriya Alhassan et al. *Artificial Neural Networks and Machine Learning – ICANN 2018*. Ed. by Věra Kůrková et al. Vol. 11141. Springer International Publishing, 2018. ISBN: 978-3-030-01423-0. DOI: 10.1007/978-3-030-01424-7. URL: http://link.springer.com/10.1007/978-3-030-01424-7.

[3] Mahmoud Amal and Mohammed Ammar. *A Survey on Deep Learning for Time-Series Forecasting*. Vol. 77. Springer Science and Business Media Deutschland GmbH, 2021, pp. 365–392. DOI: 10.1007/978-3-030-59338-4_19.

[4] O. D. Anderson. *The Box-Jenkins Approach to Time Series Analysis (*) (I)*. 1977. URL: http://www.numdam.org/item/RO_1977__11_1_3_0.pdf.

[5] Jasbir S. Arora. *Discrete Variable Optimum Design Concepts and Methods*. 2nd ed. Elsevier, 2004, pp. 513–530. ISBN: 978-0-12-064155-0. DOI: https://doi.org/10.1016/B978-0-12-064155-0.X5000-9.

[6] Ben Baldanza. *Once Derided, Airline Fees For Large Carry-On Bags Now Becoming Mainstream*. https://www.forbes.com/sites/benbaldanza/2020/12/07/once-derided-airline-fees-for-large-carry-on-bags-now-mainstream/?sh=343d400476a6. (Accessed on 01/25/2023). 2020.

[7] Emilio Bartezzaghi, Roberto Verganti, and Giulio Zotteri. *A simulation framework for forecasting uncertain lumpy demand*. 1999, pp. 499–510.

[8] BBT. *Air France-KLM baggage fee rises on European flights | Business Travel News Europe*. https://www.businesstravelnewseurope.com/Air-Travel/Air-France-KLM-baggage-fee-rises-on-European-flights. (Accessed on 01/25/2023). 2016.

[9] M. Benichou et al. *EXPERIMENTS IN MIXED-INTEGER LINEAR PROGRAMMING **. 1971, pp. 76–94.

[10] Candice Bentéjac, Anna Csörgö, and Gonzalo Martínez-Muñoz. “A comparative analysis of gradient boosting algorithms”. In: *Artificial Intelligence Review* 54 (3 Mar. 2021), pp. 1937–1967. ISSN: 15737462. DOI: 10.1007/s10462-020-09896-5.

[11] Jacek Błażewicz. *Handbook on parallel and distributed processing*. Springer, 2000, p. 635. ISBN: 9783540664413.

[12] T. O. Blinova. “Analysis of possibility of using neural network to forecast passenger traffic flows in Russia”. In: *Aviation* 11 (1 2007), pp. 28–34. ISSN: 18224180. DOI: 10.1080/16487788.2007.9635952.

[13] Constantinos Bougas. *Forecasting Air Passenger Traffic Flows in Canada: An Evaluation of Time Series Models and Combination Methods*. Université Laval, 2013.

[14] G.E.P. Box and G.M. Jenkins. *Time series analysis, forecasting and control*. San Francisco, Holden Day, 1976.

[15] Stephen P. Bradley, Arnoldo C. Hax, and Thomas L. Magnanti. *Applied Mathematical Programming*. Addison-Wesley, 1977.

[16] Leo Breiman. *Random Forests*. 2001, pp. 5–32.

[17] Jason Brownlee. *Deep Learning for Time Series Forecasting: Predict the Future with MLPs, CNNs and LSTMs in Python*. 2018.

[18] Jason Brownlee. *Random Forest for Time Series Forecasting - MachineLearningMastery.com*. https://machinelearningmastery.com/random-forest-for-time-series-forecasting/. 2020.

[19] John C. Chambers, Satinder K. Mullick, and Donald D. Smith. "How to Choose the Right Forecasting Technique". In: *Harvard Business Review* (July 1971). URL: https://hbr.org/1971/07/how-to-choose-the-right-forecasting-technique.

[20] Chuen-Jyh Chen, Shih-Ming Yang, and Zi-Cheng Wang. "Development of a Neuro-Fuzzy Model for Airline Passenger Forecasting *". In: *Astronautics and Aviation. Series A* 44 (3 2012), pp. 169–176. DOI: 10.6125/JoAAA.201209_44(3).04.

[21] Shaowu Cheng, Qian Gao, and Yapping Zhang. "Comparative Study on Forecasting Method of Departure Flight Baggage Demand". In: 2014. ISBN: 9781479946990.

[22] Sun Choi and Young Jin Kim. "Artificial neural network models for airport capacity prediction". In: *Journal of Air Transport Management* 97 (Oct. 2021). ISSN: 09696997. DOI: 10.1016/j.jairtraman.2021.102146.

[23] Paul le Clerq. *Jacht op nieuw personeel voor Schiphol verloopt nog moeizaam | RTL Nieuws*. https://www.rtlnieuws.nl/economie/bedrijven/artikel/5319477/schiphol-banenmarkt-personeelstekort-beveiligers-bagage. (Accessed on 01/20/2023). July 2022.

[24] Sven F. Crone. *Stepwise Selection of Artificial Neural Network Models for Time Series Prediction*. 2005.

[25] J. D. Croston. *Forecasting and Stock Control for Intermittent Demands*. 1972, pp. 289–303.

[26] Jonathan D. Cryer and Kung-Sik Chan. *Time Series Analysis With Applications in R*. second. Springer, 2008.

[27] CueMath. *Linear Programming - Definition, Formula, Problem, Examples*. https://www.cuemath.com/algebra/linear-programming/. (Accessed on 01/11/2023).

[28] Paolo Dell'olmo and Guglielmo Lulli. *A dynamic programming approach for the airport capacity allocation problem*. 2003, pp. 235–249. URL: http://imaman.oxfordjournals.org/.

[29] Marco Dorigo, Mauro Birattari, and Thomas Stützle. "Ant Colony Optimization". In: IEEE, 2006, pp. 28–39. DOI: 10.1109/MCI.2006.329691.

[30] Yuquan Du, Qian Zhang, and Qiushuang Chen. "ACO-IH: An improved ant colony optimization algorithm for Airport Ground Service Scheduling". In: IEEE, Aug. 2008. ISBN: 9781424417063.

[31] Jamileh Farajzadeh and Farhad Alizadeh. "A hybrid linear–nonlinear approach to predict the monthly rainfall over the Urmia Lake watershed using wavelet-SARIMAX-LSSVM conjugated model". In: *Journal of Hydroinformatics* 20 (1 2018), pp. 221–231. ISSN: 14647141. DOI: 10.2166/hydro.2017.013.

[32] James Fernando. *Moving Average (MA): Purpose, Uses, Formula, and Examples*. Jan. 2023.

[33] Christodoulos A. Floudas and Xiaoxia Lin. *Mixed Integer Linear Programming in Process Scheduling: Modeling, Algorithms, and Applications*. 2005, pp. 131–162.

[34] Aurélien Géron. *Hands-on machine learning with Scikit-Learn and TensorFlow : concepts, tools, and techniques to build intelligent systems*. 2016. ISBN: 9781491962299.

[35] Adel A. Ghobbar and Chris H. Friend. *Evaluation of forecasting methods for intermittent parts demand in the field of aviation: a predictive model*. 2003, pp. 2097–2114.

[36] Benjamin Goehry et al. *Random Forests for Time Series*, pp. 0-000.

[37] Weiwei Gong. "ARMA-GRNN for passenger demand forecasting". In: vol. 3. 2010, pp. 1577–1581. ISBN: 9781424459612. DOI: 10.1109/ICNC.2010.5583711.

[38] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[39] Royal Schiphol Group. *Baggage at Schiphol*. https://assets.ctfassets.net/biom0eqyyi6b/1LFrmaEaaI2Kgsw0G8IYgC/48c4c91d2b45712d1bb32b4501ad34a8/Baggage_at_Schiphol.pdf. (Accessed on 28/10/2022). 2018.

[40] Liang Guo et al. “The hybrid PROPHET-SVR approach for forecasting product time series demand with seasonality”. In: *Computers and Industrial Engineering* 161 (Nov. 2021). ISSN: 03608352. DOI: 10.1016/j.cie.2021.107598.

[41] Weian Guo et al. “Scheduling for airport baggage transport vehicles based on diversity enhancement genetic algorithm”. In: *Natural Computing* 19 (4 Dec. 2020), pp. 663–672. ISSN: 15729796. DOI: 10.1007/s11047-018-9703-0.

[42] Md Mohedul Hasan and Md Rajib Arefin. *Application of Linear Programming in Scheduling Problem*. 2017, pp. 145–150.

[43] Nathaniel E Helwig. *Multivariate Linear Regression*. University of Minnesota, Jan. 2017.

[44] Emilcy Hernández et al. “Rainfall prediction: A deep learning approach”. In: vol. 9648. Springer Verlag, 2016, pp. 151–162. ISBN: 9783319320335. DOI: 10.1007/978-3-319-32034-2_13.

[45] Bertha Hidalgo and Melody Goodman. “Multivariate or multivariable regression?” In: *American Journal of Public Health* 103 (1 2013), pp. 39–40. ISSN: 15410048. DOI: 10.2105/AJPH.2012.300897.

[46] Frederick S. Hillier and Gerald J. Lieberman. *Introduction to Operations Research*. 11th ed. McGraw Hill, 2021.

[47] Sepp Hochreiter and Jürgen Schmidhuber. “Long Short-Term Memory”. In: *Neural Computation* 9 (8 Nov. 1997), pp. 1735–1780. ISSN: 08997667. DOI: 10.1162/neco.1997.9.8.1735.

[48] John H. Holland. “Genetic Algorithms”. In: 267 (1 1992), pp. 66–73. DOI: 10.2307/24939139.

[49] Charles C. Holt. “Forecasting seasonals and trends by exponentially weighted moving averages”. In: *International Journal of Forecasting* 20 (1 Jan. 2004), pp. 5–10. ISSN: 01692070. DOI: 10.1016/j.ijforecast.2003.09.015.

[50] Rob J. Hyndman and George Athanasopoulos. *Forecasting: Principles and Practice*. 2nd edition. Accessed on 21/12/2022. OTexts: Melbourne, Australia, Apr. 2018. URL: https://OTexts.com/fpp2/.

[51] IBM. *What is integer programming? - IBM Documentation*. https://www.ibm.com/docs/en/icos/12.8.0.0?topic=problem-what-is-integer-programming. (Accessed on 01/15/2023).

[52] IBM. *What is Supervised Learning? | IBM*. (Accessed on 03/06/2023).

[53] W. H. Ip, Nick Chung, and George Ho. *Using Integer Programming for Airport Service Planning in Staff Scheduling*. 2010, pp. 85–92.

[54] Arun Jagota. “Exponential Smoothing Approaches in Time Series Forecasting”. In: *Towards Data Science* (Sept. 2020). URL: https://towardsdatascience.com/exponential-smoothing-approaches-to-forecasting-time-series-34e4957ed1a.

[55] Yunjian Jia et al. “A Combined Forecasting Model for Passenger Flow Based on GM and ARMA”. In: *International Journal of Hybrid Information Technology* 9 (2 Feb. 2016), pp. 215–226. ISSN: 17389968. DOI: 10.14257/ijhit.2016.9.2.19.

[56] Katie Johnston. *Hate airline baggage fees? Needham man is to blame - The Boston Globe*. https://www.bostonglobe.com/business/2013/04/27/the-man-behind-bag-fees/JHLvuMZhXLOgBZgGzWKCbJ/story.html. (Accessed on 01/25/2023). 2013.

[57] Andreas Kanavos et al. “Deep learning models for forecasting aviation demand time series”. In: *Neural Computing and Applications* 33 (23 Dec. 2021), pp. 16329–16343. ISSN: 14333058. DOI: 10.1007/s00521-021-06232-y.

[58] Sourabh Katoch, Sumit Singh Chauhan, and Vijay Kumar. “A review on genetic algorithm: past, present, and future”. In: *Multimedia Tools and Applications* 80 (5 Feb. 2021), pp. 8091–8126. ISSN: 15737721. DOI: 10.1007/s11042-020-10139-6.

[59] Joos Korstanje. *Advanced Forecasting with Python*. Apress, 2021. DOI: 10.1007/978-1-4842-7150-6.

[60] C. Koulamas, S.R. Antony, and R. Jaen. *A Survey of Simulated Annealing Applications to Operations Research Problems*. Pergamon, 1994, pp. 41–56.

[61] Kenneth Kuhn and Steffen Loth. *Airport Service Vehicle Scheduling*. 2009.

[62] Jesus Lago, Fjo De Ridder, and Bart De Schutter. “Forecasting spot electricity prices: Deep learning approaches and empirical comparison of traditional algorithms”. In: *Applied Energy* 221 (July 2018), pp. 386–405. ISSN: 03062619. DOI: 10.1016/j.apenergy.2018.02.069.

[63] E. L. Lawler and D. E. Wood. *Branch-And-Bound Methods: A Survey*. 1966. URL: https://about.jstor.org/terms.

[64] Cheng Li. “Combined forecasting of civil aviation passenger volume based on ARIMA-REGRESSION”. In: *International Journal of System Assurance Engineering and Management* 10 (5 Oct. 2019), pp. 945–952. ISSN: 09764348. DOI: 10.1007/s13198-019-00825-6.

[65] Ziyu Li et al. “Analysis of airport departure baggage check-in process based on passenger behavior”. In: vol. 2. Institute of Electrical and Electronics Engineers Inc., Feb. 2018, pp. 204–207. ISBN: 9781538636749. DOI: 10.1109/ISCID.2017.149.

[66] Shaofu Lin and Hengyu Tian. “Short-Term Metro Passenger Flow Prediction Based on Random Forest and LSTM”. In: *2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*. Vol. 1. 2020, pp. 2520–2526. DOI: 10.1109/ITNEC48623.2020.9084974.

[67] Junhao Liu et al. “An Improved Genetic Algorithm-Based Traffic Scheduling Model for Airport Terminal Areas”. In: *Journal of Sensors* 2022 (2022). ISSN: 16877268. DOI: 10.1155/2022/7926335.

[68] Ying Liu et al. “Allocation and optimization of shared self-service check-in system based on integer programming model”. In: *Journal of Combinatorial Optimization* 44 (1 Aug. 2022), pp. 532–556. ISSN: 15732886. DOI: 10.1007/s10878-021-00839-6.

[69] Yunpeng Liu et al. “Multi-step ahead time series forecasting for different data patterns based on LSTM recurrent neural network”. In: vol. 2018-January. Institute of Electrical and Electronics Engineers Inc., Apr. 2018, pp. 305–310. ISBN: 9781538648063. DOI: 10.1109/WISA.2017.25.

[70] Andrew Loo. *Exponentially Weighted Moving Average (EWMA)*. (Accessed on 02/24/2023). Feb. 2023.

[71] Qun Ma et al. “Research on Prediction of Checked-baggage Departed from Airport Terminal Based on Time Series Analysis”. In: Institute of Electrical and Electronics Engineers Inc., 2021, pp. 264–269. ISBN: 9781665402323. DOI: 10.1109/ICNISC54316.2021.00055.

[72] Amal Mahmoud and Ammar Mohammed. *A Survey on Deep Learning for Time-Series Forecasting*. Vol. 77. Springer Science and Business Media Deutschland GmbH, 2021, pp. 365–392. DOI: 10.1007/978-3-030-59338-4_19.

[73] MathWorks. *Time Series Regression I: Linear Models - MATLAB & Simulink Example*. https://www.mathworks.com/help/econ/time-series-regression-i-linear-models.html. 2022.

[74] Sidra Mehtab and Jaydip Sen. *Analysis and Forecasting of Financial Time Series Using CNN and LSTM-Based Deep Learning Models*. 2021. URL: https://www.researchgate.net/publication/345970760.

[75] Sidra Mehtab and Jaydip Sen. *Stock Price Prediction Using Convolutional Neural Networks on a Multivariate Timeseries*. 2020.

[76] Seyedali Mirjalili. *Ant Colony Optimisation*. Springer, 2018. Chap. 3, pp. 33–42. ISBN: 978-3-319-93025-1. DOI: https://doi.org/10.1007/978-3-319-93025-1. URL: http://www.springer.com/series/7092.

[77] Seyedali Mirjalili. *Genetic Algorithm*. Springer, 2018. Chap. 4, pp. 43–55. ISBN: 978-3-319-93025-1. DOI: https://doi.org/10.1007/978-3-319-93025-1. URL: http://www.springer.com/series/7092.

[78] Ethel Mokotoff. “Parallel machine scheduling problems: A survey”. In: *Asia-Pacific Journal of Operational Research* 18 (2 Nov. 2001), p. 193.

[79] M. Navratil and A. Kolkova. “Decomposition and forecasting time series in business economy using prophet forecasting model”. In: *Central European Business Review* 8 (4 2019), pp. 26–39. ISSN: 18054862. DOI: 10.18267/j.cebr.221.

[80] Thomas H. Naylor, Terry G. Seaks, and D. W. Wichern. *Box-Jenkins Methods: An Alternative to Econometric Models*. 1972, p. 123.

[81] NOS. *Chaos op Schiphol door KLM-staking: reizigers opgeroepen niet meer te komen*. https://nos.nl/artikel/2426169-chaos-op-schiphol-door-klm-staking-reizigers-opgeroepen-niet-meer-te-komen. (Accessed on 01/27/2023). Apr. 2022.

[82] Nurhamidah Nurhamidah, Nusyirwan Nusyirwan, and Ahmad Faisol. "Forecasting Seasonal Time Series Data Using the Holt-Winters Exponential Smoothing Method of Additive Models". In: *Jurnal Matematika Integratif* 16 (2 Dec. 2020), p. 151. ISSN: 1412-6184. DOI: 10.24198/jmi.v16.n2.29293.151-157.

[83] Bogdan Oancea and Ştefan Cristian Ciucu. *TIME SERIES FORECASTING USING NEURAL NETWORKS*. 2014.

[84] Alex D. Papalexopoulos and Timothy C. Hesterberg. *A REGRESSION-BASED APPROACH TO SHORT-TERM SYSTEM LOAD FORECASTING*. 1990.

[85] Luca Di Persio and Oleksandr Honchar. "Artificial neural networks approach to the forecast of stock market price movements". In: *International Journal of Economics and Management Systems* (2016). URL: http://iaras.org/iaras/journals/ijems.

[86] Gábor Petneházi. "Recurrent Neural Networks for Time Series Forecasting". In: (Dec. 2018). URL: http://arxiv.org/abs/1901.00069.

[87] Osama Yaseen M. Al-Rawi and Taniya Mukherjee. "Application of Linear Programming in Optimizing Labour Scheduling". In: *Journal of Mathematical Finance* 09 (03 2019), pp. 272–285. ISSN: 2162-2434. DOI: 10.4236/jmf.2019.93016.

[88] J. J. Ruiz-Aguilar, I. J. Turias, and M. J. Jiménez-Come. "Hybrid approaches based on SARIMA and artificial neural networks for inspection time series forecasting". In: *Transportation Research Part E: Logistics and Transportation Review* 67 (2014), pp. 1–13. ISSN: 13665545. DOI: 10.1016/j.tre.2014.03.009.

[89] Asep Rusyana, Amiruddin Hasan, and Maulina Oktaviana. *Forecasting Passenger by Using Holt's Exponential Smoothing and Winter's Exponential Smoothing*. 2016. ISBN: 978-602-71798-1-3. URL: https://www.researchgate.net/publication/316349640.

[90] Rob A. Rutenbar. "Simulated Annealing Algorithms: An Overview". In: IEEE, Jan. 1989, pp. 19–26. DOI: 10.1109/101.17235.

[91] Merve Şahin, Recep Kızılaslan, and Ömer F Demirel. *Forecasting Aviation Spare Parts Demand Using Croston Based Methods and Artificial Neural Networks*. 2013, pp. 1–21.

[92] Ameer Salem, Georgios C. Anagnostopoulos, and Ghaith Rabadi. *A Branch-and-Bound Algorithm for Parallel Machine Scheduling Problems*. 2000.

[93] Albin Salmi. *The impact of COVID-19 on machine learning models in a commercial aviation use case*. 2022.

[94] Christophorus Beneditto Aditya Satrio et al. "Time series analysis and forecasting of coronavirus disease in Indonesia using ARIMA model and PROPHET". In: vol. 179. Elsevier B.V., 2021, pp. 524–532. DOI: 10.1016/j.procs.2021.01.036.

[95] Schiphol. *Schiphol | Traffic and transport figures per month*. https://www.schiphol.nl/en/schiphol-group/page/transport-and-traffic-statistics/. (Accessed on 25/10/2022).

[96] Rick Seaney. *History of Airline Fees: Bags, Food & More | FareCompare*. https://www.farecompare.com/travel-advice/airline-fees-bags-history/. (Accessed on 01/25/2023). 2017.

[97] King Loong Shiu and K.Y. Szeto. "Self-adaptive Mutation Only Genetic Algorithm: An Application on the Optimization of Airport Capacity Utilization". In: Springer, Nov. 2008, pp. 428–435.

[98] Xiaogang Su, Xin Yan, and Chih Ling Tsai. "Linear regression". In: *Wiley Interdisciplinary Reviews: Computational Statistics* 4 (3 May 2012), pp. 275–294. ISSN: 19395108. DOI: 10.1002/wics.1198.

[99] Xinxin Tang and Guangming Deng. "Prediction of Civil Aviation Passenger Transportation Based on ARIMA Model". In: *Open Journal of Statistics* 06 (05 2016), pp. 824–834. ISSN: 2161-718X. DOI: 10.4236/ojs.2016.65068.

[100] Sean J. Taylor and Benjamin Letham. *Forecasting at Scale*. 2017. DOI: 10.1080/0031305.2017.1380080.

[101] Wai Hong Kan Tsui and Faruk Balli. "International arrivals forecasting for Australian airports and the impact of tourism marketing expenditure". In: *Tourism Economics* 23 (2 2017), pp. 403–428. ISSN: 13548166. DOI: 10.5367/te.2015.0507.

[102] Wai Hong Kan Tsui et al. "Forecasting of Hong kong airport's passenger throughput". In: *Elsevier* (2014).

[103] Hristos Tyralis and Georgia Papacharalampous. "Variable selection in time series forecasting using random forests". In: *Algorithms* 10 (4 Dec. 2017). ISSN: 19994893. DOI: 10.3390/a10040114.

[104] Yugesh Verma. "Complete Guid to SARIMAX in Python for Time Series Modeling". In: *AIM* (July 2021). URL: https://analyticsindiamag.com/complete-guide-to-sarimax-in-python-for-time-series-modeling/#:~:text=SARIMAX(Seasonal%5C%20Auto%5C%2DRegressive%5C%20Integrated,average%5C%20component%5C%20in%5C%20the%5C%20model..

[105] Maftukhatul Qomariyah Virati, Diory Paulus Pamanik, and Setia Pramana. "Forecasting Number of Passengers of TransJakarta using SARIMAX Method". In: *OPEN ACCESS J DATA SCI APPL* 3 (1 2020), pp. 31–037. DOI: 10.34818/JDSA.2020.3.45.

[106] Robert L. Vislocky and J. Michael Fritsch. "Generalized Additive Models versus Linear Regression in Generating Probabilistic MOS Forecasts of Aviation Weather Parameters". In: (1995), pp. 669–680.

[107] Rillo S. Wahyudin et al. "Resource Allocation Model to Find Optimal Allocation of Workforce, Material, and Tools in an Aircraft Line Maintenance". In: IMECS, Mar. 2016. ISBN: 9789881404763.

[108] Liya Wang et al. *Deep Learning for Flight Demand Forecasting*. 2011.

[109] Shengyou Wang et al. "Traffic flow prediction using bi-directional gated recurrent unit method". In: *Urban Informatics* 1 (1 Dec. 2022), p. 16. ISSN: 2731-6963. DOI: 10.1007/s44212-022-00015-z. URL: https://link.springer.com/10.1007/s44212-022-00015-z.

[110] Lewis Woolfson. *Schedule Optimisation using Linear Programming in Python | by Lewis Woolfson | Towards Data Science*. https://towardsdatascience.com/schedule-optimisation-using-linear-programming-in-python-9b3e1bc241e1. (Accessed on 01/26/2023). 2021.

[111] Shuojiang Xu, Hing Kai Chan, and Tiantian Zhang. "Forecasting the demand of the aviation industry using hybrid time series SARIMA-SVR approach". In: *Transportation Research Part E: Logistics and Transportation Review* 122 (Feb. 2019), pp. 169–180. ISSN: 13665545. DOI: 10.1016/j.tre.2018.12.005.

[112] Jianying Yan and Lina Shi. "Analysis on the Allocation Optimization of Airport Baggage Turntable". In: vol. 1802. IOP Publishing Ltd, Mar. 2021. DOI: 10.1088/1742-6596/1802/3/032125.

[113] Jiangni Yu. "Short-term Airline Passenger Flow Prediction Based on the Attention Mechanism and Gated Recurrent Unit Model". In: *Cognitive Computation* 14 (2 Mar. 2022), pp. 693–701. ISSN: 18669964. DOI: 10.1007/s12559-021-09991-x.

[114] Dong Zhang et al. "Container oriented job scheduling using linear programming model". In: IEEE, 2017. ISBN: 9781509063062. DOI: 10.1109/INFOMAN.2017.7950370.

[115] Guoqiang Zhang, B. Eddy Patuwo, and Michael Y. Hu. *Forecasting with artificial neural networks: The state of the art*. 1998, pp. 35–62.

[116] Jian Zheng et al. "Electric load forecasting in smart grids using Long-Short-Term-Memory based Recurrent Neural Network". In: Institute of Electrical and Electronics Engineers Inc., May 2017. ISBN: 9781509047802. DOI: 10.1109/CISS.2017.7926112.

A
Gantt Chart

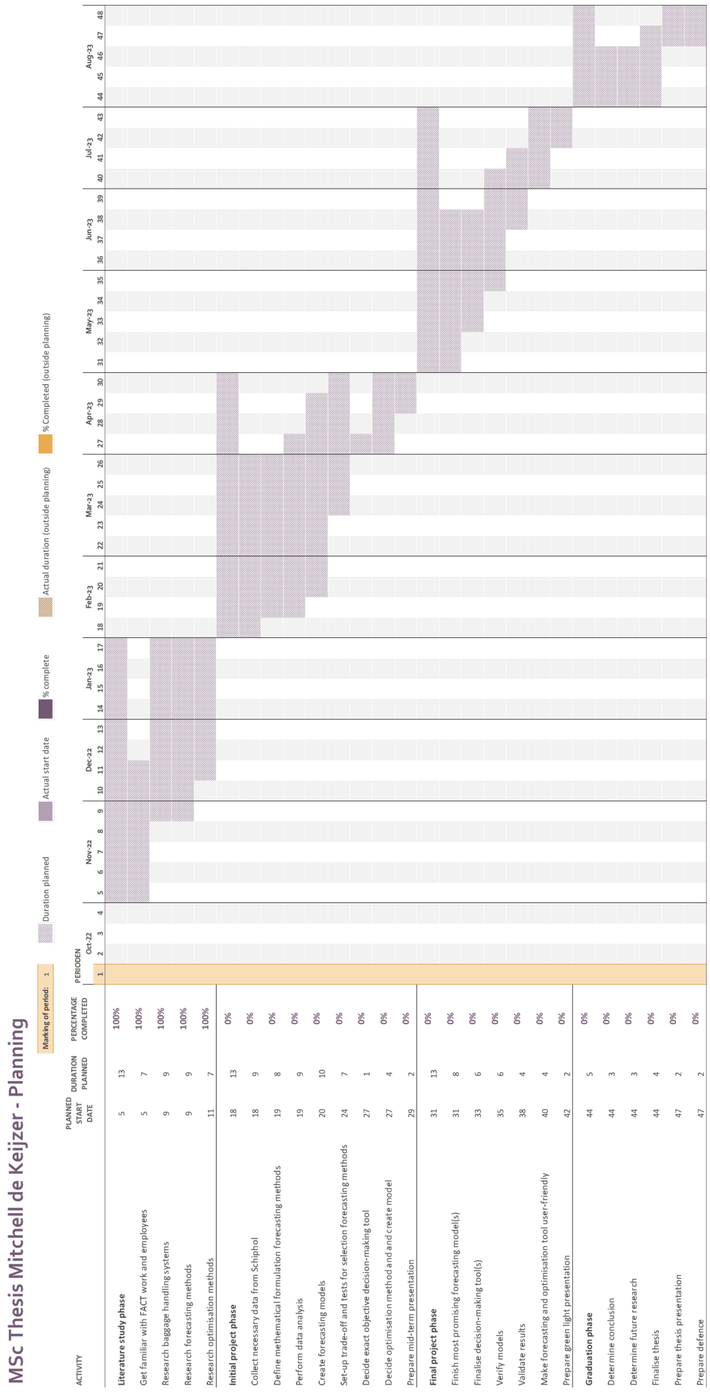


Figure A.1: Gantt Chart thesis project

III

Supporting work

1
Forecast Method Testing

In order to test the forecast methods that were identified during the literature study, a dataset was set-up to find out which methods are able to make an accurate prediction and what the differences are. In this chapter, section 1.1 explains what the dataset entails and how the forecast methods will be tested against each other. Subsequently, section 1.2 presents the results of each forecast method and the trade-off that has been made.

1.1. Used Data & Test Set-Up

In order to test the forecast methods based on data that could be real, a dummy dataset has been created by adjusting a real dataset that contains the average baggage factor (BF) per day of an airline for one year (2022). Each method will be tested for a forecast period of 7 and 30 days. The test set-up consists of the following metrics:

- Accuracy score (R-squared score): used to assess the goodness of fit of a regression model in forecasting. It indicates how well the model's predictions explain the variability observed in the actual data. It ranges from 0 to 1, with a higher value indicating a better fit of the model to the data.
- Ability to capture trend and seasonality: it is known that the BF has a certain trend and seasonality over the years and therefore it is important that the model is able to capture this to improve the performance.
- Use of explanatory variables: some forecast methods base their prediction only on the target variable, whilst others are able to use explanatory variables to improve the forecast performance by understanding why the target variable has certain values over time.
- Dealing with irregular time intervals: the dataset on which the final model must be able to make predictions will contain an arbitrary amount of flights on each day and therefore it is not possible to make predictions for each individual flight by prediction based on a regular time interval.
- Model complexity: the more complex a model is, the better the prediction could become. However, since employees of FACT at Schiphol must be able to use and most importantly understand it, a trade-off must be made between performance and model complexity.
- User friendliness: similar to the previous point, employees must be able to understand and use the tool. Therefore, it must be relatively easy to understand how the model works and how to adjust it if necessary.
- Computationally expensive: a model that takes some time to make its predictions is okay. However, since predictions are prone to errors, a model that is not too computationally expensive is required to be able to do multiple runs during a day if necessary.
- Overfitting: when developing models, there are measures against overfitting. However, it is still important to take this into account for each method to see what can be done against it and how sensitive it is.

1.2. Forecast Method Results

The forecast methods were subjected to testing by generating forecasts for both 7-day and 30-day periods. Given the similarity of the results, only the predictions for the 30-day period are depicted in the figures. The initial methods evaluated included the moving average (MA) and double and triple exponential smoothing (ES), with their corresponding results presented in Figures Figure 1.1, Figure 1.2, and Figure 1.3. The MA method exhibits a certain trend in its predictions but fails to capture the majority of the peaks observed in the data. On the other hand, the Holt ES method manages to capture the trend of a sudden increase in the BF during the last week, but it does not accurately replicate the underlying pattern. The Holt-Winters method, another form of ES, exhibits a slight improvement in capturing the pattern when combined with the trend component, although it still deviates significantly from the actual values.

Moving forward to the Box-Jenkins method, the methodology commences with Autoregression (AR), with the outcomes shown in Figure Figure 1.4. This particular method demonstrates proficiency in identifying patterns within the data; however, it struggles to immediately identify abrupt shifts due to insufficient information. By incorporating AR with MA and integrating the Integral part, the Autoregressive Integrated Moving Average (ARIMA) model is formed, as depicted in Figure Figure 1.5. The amalgamation of AR, MA, and the Integration component grants the model the ability to recognize both the pattern and trend, resulting in improved performance compared to the aforementioned methods. Nonetheless, the ARIMA model remains unable to capture the initial peak during the winter holiday season. In order to capture this peak and enhance model accuracy, the inclusion of Seasonal and exogenous factors becomes imperative, leading to the development of the SARIMAX model. Figure Figure 1.6 demonstrates that the addition of a holiday feature as an exogenous factor empowers the model to make highly accurate predictions.

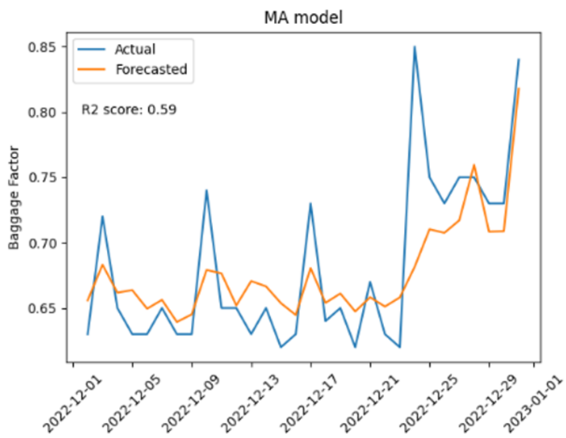


Figure 1.1: 30-day forecast with the MA method

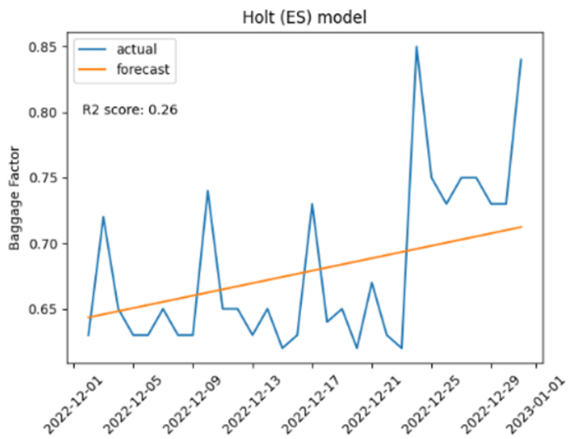


Figure 1.2: 30-day forecast with the Holt ES method

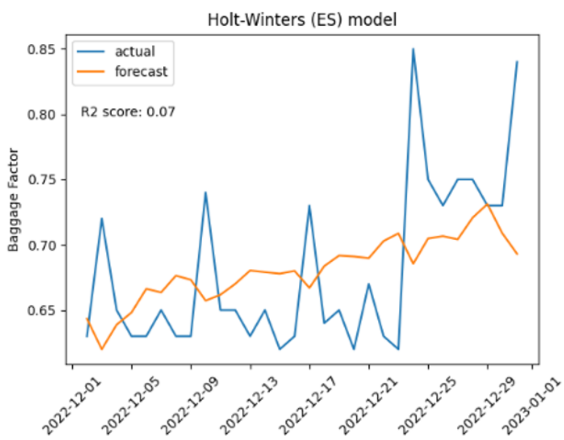


Figure 1.3: 30-day forecast with the Holt-Winters ES method

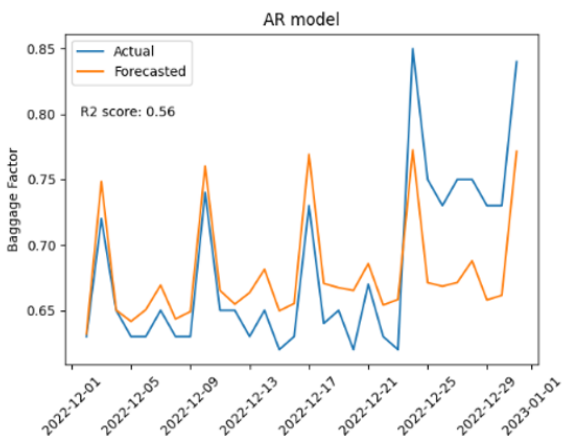


Figure 1.4: 30-day forecast with the AR method

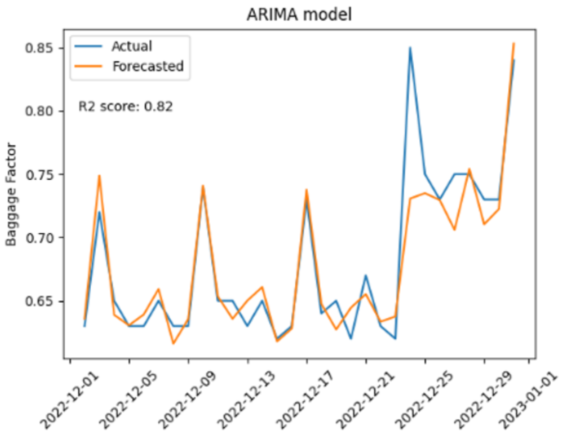


Figure 1.5: 30-day forecast with the ARIMA method

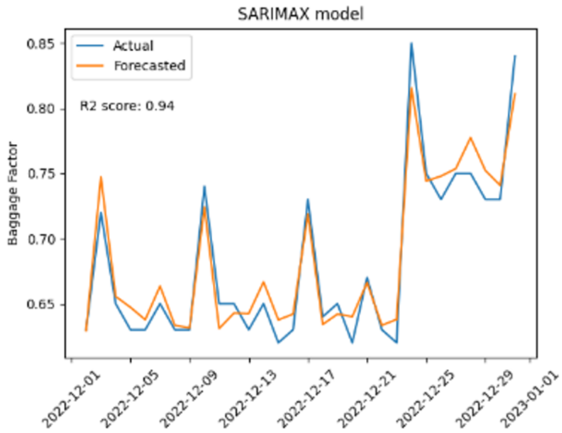


Figure 1.6: 30-day forecast with the SARIMAX method

Up to this point, the methods employed were primarily time series methods. However, the subsequent analysis focuses on supervised machine learning models. First, the results for the Multiple Linear Regression (MLR) method can be observed in Figure Figure 1.7. MLR models have the capability to incorporate exogenous variables, and the results demonstrate the beneficial impact of this feature. The accuracy achieved by the MLR model is remarkably high, with nearly perfect replication of the underlying pattern. Similarly, the decision tree-based methods, namely Random Forest (RF), XGBoost, and LightGBM, exhibit exceptional performance as depicted in Figures Figure 1.8, Figure 1.9, and Figure 1.10, respectively. An advantage of decision tree methods is that they are not contingent upon regular time intervals, as the decision tree approach does not rely on such temporal considerations. Conversely, when the MLR method was tested with irregular time intervals, the results were less accurate.

In addition to the aforementioned models, two advanced machine learning and deep learning models were developed: Prophet and DeepAR, with their corresponding outcomes presented in Figures Figure 1.11 and Figure 1.12, respectively. The Prophet model exhibits remarkable accuracy in its predictions, while the DeepAR model deviates significantly from the actual values. Both models have the ability to incorporate external features for enhanced prediction capabilities. It is worth noting that the lack of accuracy in the DeepAR model may stem from improperly set parameters for this specific type of data. In summary, the supervised machine learning models, such as MLR, decision tree-based methods (RF, XGBoost, LightGBM), and advanced models (Prophet, DeepAR), demonstrate varying degrees of accuracy and capability in capturing patterns. The MLR model and decision tree-based methods prove to be highly accurate and robust, while the advanced models yield mixed results, with Prophet exhibiting high accuracy and DeepAR falling short.

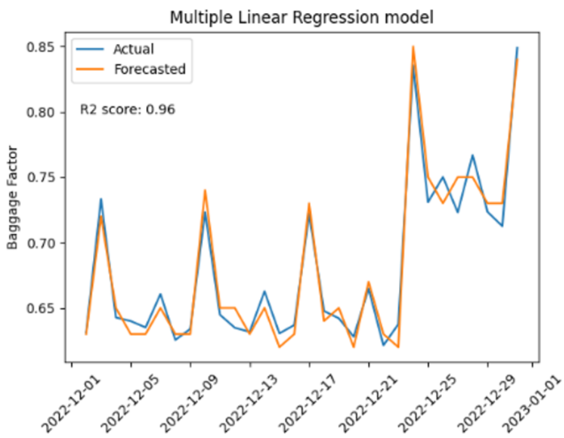


Figure 1.7: 30-day forecast with the MLR method

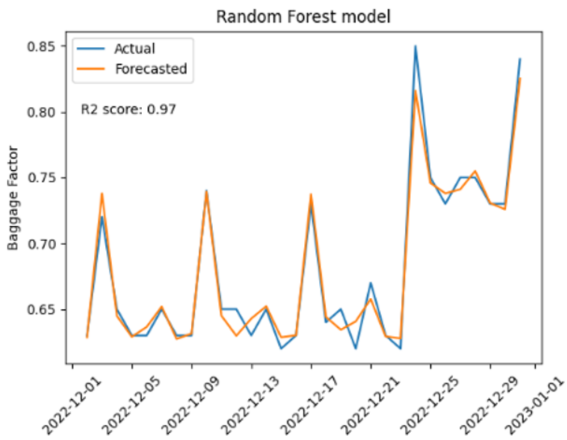


Figure 1.8: 30-day forecast with the RF method

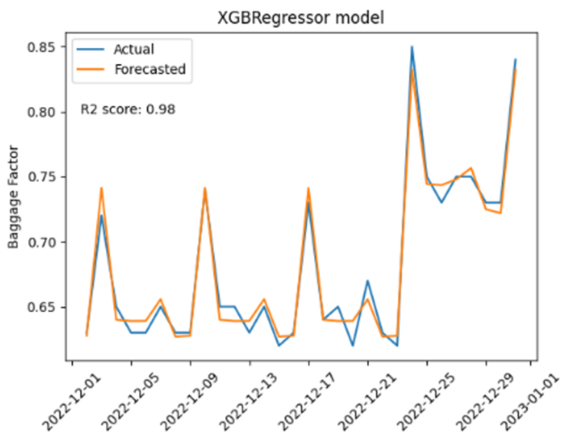


Figure 1.9: 30-day forecast with the XGBoost method

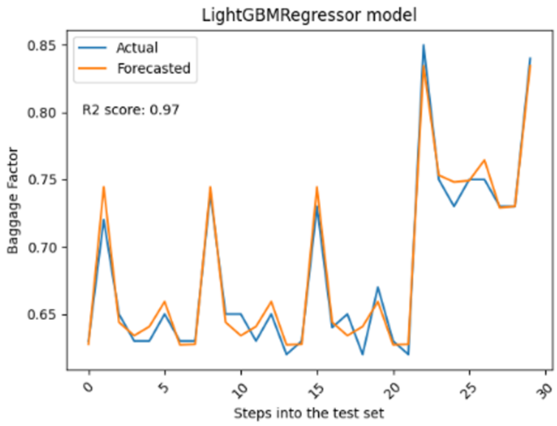


Figure 1.10: 30-day forecast with the LightGBM method

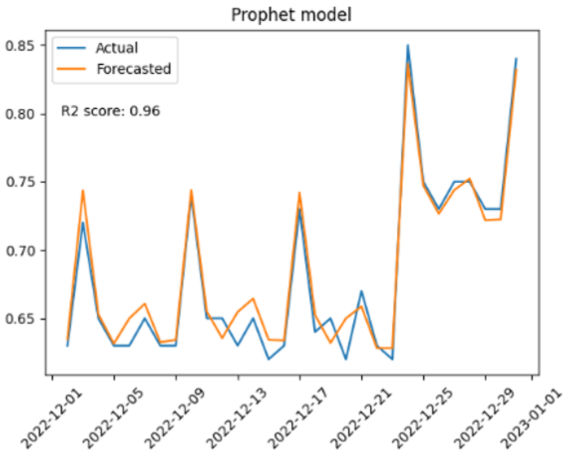


Figure 1.11: 30-day forecast with the Prophet method

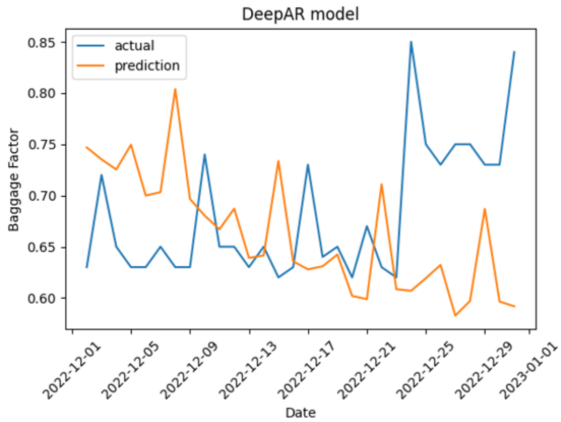


Figure 1.12: 30-day forecast with the DeepAR method

In conclusion, it is evident that the inclusion of exogenous factors in the final forecast model is crucial, and the utilisation of machine learning methods yields significantly higher accuracy compared to the time series models. Furthermore, all machine learning models demonstrate the ability to capture trend and seasonality, while also incorporating exogenous variables. Although most of these models can accommodate irregular time intervals, they generally possess greater complexity and are less user-friendly in comparison to the time series models. Consequently, the decision has been made to proceed with the utilisation of decision tree methods, specifically gradient boosting, for the development of the final forecast model. The primary reasons behind this choice are the models' capability to generate accurate predictions and their capacity to provide insights into the decision-making process (such as how trees are constructed and which features contribute to the highest model gain), thus enhancing user-friendliness. Additionally, the computational time required for these models is relatively fast when compared to other approaches. It is important to note that overfitting can be a potential issue with these models. However, this concern can be addressed through hyperparameter tuning and careful selection of features to ensure optimal model performance.

2

Initial Data Analysis

This chapter presents a preliminary analysis of the data, providing insights into the short-term and long-term patterns observed throughout the year 2022. Figure 2.1 illustrates the average BF per week. The results indicate a significant increase in BF during the winter season, compared to other times of the year. Moreover, a slight rise in the average BF is observed during the summer season. In contrast, Figure 2.2 displays the total number of checked baggage items per week. The pattern differs significantly from the previous figure. It is evident that a greater number of baggage items are transported during the summer season in comparison to other periods. This difference can be attributed to the increased number of flights departing during the summer season. Notably, the average BF remains relatively constant despite these variations.

Upon closer examination of the weekly data, Table 2.1 presents the average BF per weekday. The weekdays generally exhibit a relatively stable pattern in terms of BF, except for Saturday, which displays a modest increase of approximately 0.1 in BF compared to other weekdays. Further analysis focusing on daily variations is provided in Table 2.2, which showcases the disparities in average BF across different time periods. Two noteworthy observations emerge from the analysis. Firstly, the time period between 08:00 and 16:00 emerges as the busiest interval of the day, characterised by the highest average BF. Secondly, even though the number of departing flights decreases after 21:00, the BF remains considerably high in comparison to other time periods. This phenomenon can be attributed to the prevalence of intercontinental flights during this late period, which tend to involve a significant number of checked baggage items. Lastly, the analysis in Table 2.3 provides the average BF for both European flights and intercontinental flights. The results clearly demonstrate that intercontinental flights exhibit a significantly higher average BF compared to European flights.

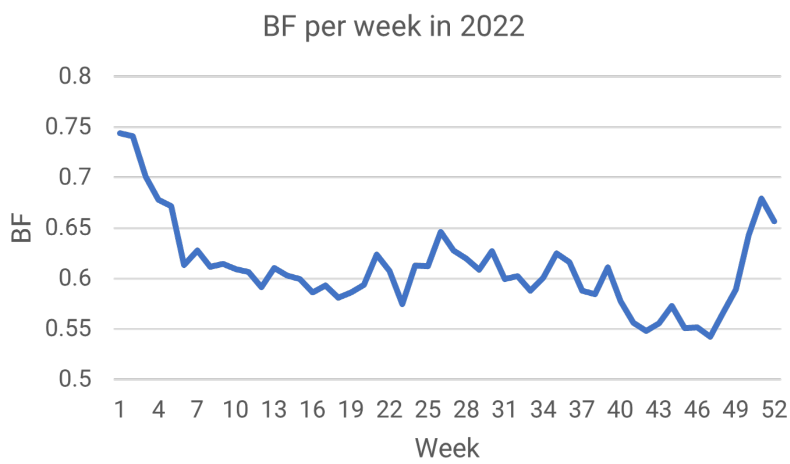


Figure 2.1: Average BF per week in 2022 of all flights

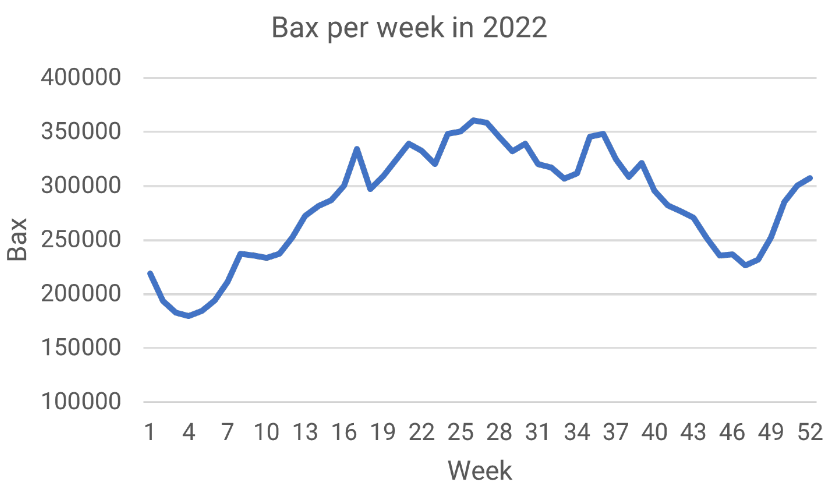


Figure 2.2: Total bax per week in 2022 of all flights

Table 2.1: Average BF per weekday in 2022 of all flights

Weekday	Average BF
Monday	0.592
Tuesday	0.615
Wednesday	0.611
Thursday	0.590
Friday	0.587
Saturday	0.692
Sunday	0.613

Table 2.2: Average BF per time period in 2022 of all flights

Time Period	Average BF	% of all flights
00:00-04:00	0.592	0.1
04:00-05:00	0.495	2.4
05:00-06:00	0.479	5.2
06:00-07:00	0.570	5.6
07:00-08:00	0.627	6.9
08:00-09:00	0.705	7.9
09:00-10:00	0.715	7.4
10:00-11:00	0.729	7.1
11:00-12:00	0.722	5.6
12:00-13:00	0.705	7.8
13:00-14:00	0.694	6.3
14:00-15:00	0.664	5.8
15:00-16:00	0.600	6.5
16:00-17:00	0.443	3.9
17:00-18:00	0.390	3.2
18:00-19:00	0.512	4.8
19:00-20:00	0.579	8.3
20:00-21:00	0.587	4.8
21:00-22:00	0.803	0.3
22:00-23:00	0.852	0.1
23:00-24:00	-	0.0

Table 2.3: Average BF per outbound range in 2022 of all flights

Outbound Range	**Average BF**
Europe	0.544
Intercontinental	0.971

Bibliography

[1] Royal Schiphol Group. Schiphol | traffic and transport figures per month. https://www.schiphol.nl/en/schiphol-group/page/transport-and-traffic-statistics/, 2019. (Accessed on 25/10/2022).