APOC BUSINESS PROCESS REENGINEERING BIG DATA STUDY

-NOVEMBER 21, 2016-
authors

UCL School of Management, University College London
Bert De Reyck – bdercyck@ucl.ac.uk
Xiaojia Guo – x.quo.11@ucl.ac.uk

Darden School of Business, University of Virginia
Yael Grushka-Cockayne – GrushkaY@darden.virginia.edu
Kenneth C. Lichtendahl Jr. – LichtendahlC@darden.virginia.edu
Andrey Karasev – andreyjet@gmail.com

Heathrow Airport
Tom Garside – Tom_Garside@heathrow.com
Neville Coss – Neville_Coss_supplier@heathrow.com
Frederick Tasker – Frederick_Tasker@heathrow.com
ABSTRACT

Improving airport performance is at the heart of the SESAR’s Airport Operations Centre (AOC) solution. By providing access to real-time data from various data sources of different AOC stakeholders, airports can make accurate predictions about their operations, including passenger movements. In this study, we review AOC roles and responsibilities, identify the key AOC processes that could be enhanced by data-driven predictions and machine learning techniques (DDP&ML), and demonstrate a case study of how shared data and advanced analytics can be used to make predictions of passengers’ connection times. In the case study, a regression tree model is fitted to a large training set with 3.7 million passenger records. This predictive model is applied to generate forecasts (together with prediction intervals) of each passenger’s connection time and the passenger flows during an eight-hour live trial. The real-time predictions generated by the model could be used to inform Target-Off-Block Time (TOBT) adjustments and determine transfer security resourcing levels.
# CONTENTS

1. **INTRODUCTION**..........................................................- 4 -

2. **APOC OVERVIEW** ......................................................- 6 -
   2.1 Purpose of APOC, Roles and Key Decision Processes ............- 6 -
   2.2 Data Sources ..................................................................- 9 -
   2.3 Frame and Identify Selected DDP&ML Activity ......................- 9 -

3. **THE TRANSFER PASSENGER PROBLEM** .................- 11 -
   3.1 Heathrow Transfer Passenger Journey ............................- 11 -
   3.2 Problem Framing ............................................................- 12 -

4. **THE PREDICTIVE MODEL** ..............................................- 15 -
   4.1 Data Collection ..............................................................- 16 -
   4.2 Training the Model .........................................................- 18 -
   4.3 Generating Forecasts from the Model ...............................- 22 -
   4.4 Accuracy Assessment and Discussion ...............................- 26 -
   4.5 Retraining the Model .......................................................- 29 -

5. **LIVE TRIAL AND ASSESSMENTS** ......................- 30 -
   5.1 Real-time Input Data ....................................................- 30 -
   5.2 The Live Trial ..............................................................- 30 -
   5.3 Discussion ........................................................................- 33 -

6. **PRESCRIPTIVE ANALYSIS** ...........................................- 35 -
   6.1 TOBT Adjustment According to Passenger Delays .............- 35 -
   6.2 Security Lane Resourcing ...............................................- 37 -
   6.3 Inter-Terminal Coach Frequency and Routing ....................- 38 -

7. **CONCLUSIONS AND FUTURE WORK** ......................- 39 -
Figures and Tables

Figure 1. APOC Roles and Airport Interfaces. - 6 -
Figure 2. The Heathrow Connection Passenger Journey. - 11 -
Figure 3. The DMAC system. - 13 -
Figure 4. Key Process to be Predicted and the Available Data Sets. - 13 -
Figure 5. System Architecture. - 15 -
Figure 6. Data Consolidation. - 17 -
Figure 7. Performance of Decision Trees with Different Tuning Parameters and Features. - 19 -
Figure 8. The Regression Tree Model for Predicting Passengers’ Connection Times. - 20 -
Figure 9. Predictors of the Model and Their Feature Importance. - 21 -
Figure 10. Distributions of the Instances Fall into Each Leaf. - 22 -
Figure 11. Examples of the Output Files. - 24 -
Figure 12. Predictions of the 15 min Connecting Passenger Flows from June 7 to 13, 2016. - 27 -
Figure 13. Interface of the Connection Time Forecasting Application. - 30 -
Figure 14. Example of Running Trials with a 2 hours Forecasting Window and a 5 minutes Updating Frequency. - 31 -
Figure 15. Example of Passenger Flow Predictions Generated by the Application. - 32 -
Figure 16. Assessing the Live Trial in the APOC Room. - 32 -
Figure 17. Real-time Monitoring of the T5 Flight Connection Area. - 33 -
Figure 18. PTM Timing Analysis of July 17th. - 34 -
Figure 19. An Example of the Improvement on TOBT Predictability. - 36 -
Figure 20. System Architecture. - 45 -
Figure 21. Examples of the BOSS, BDD and Conformance Data sets. - 44 -
Figure 22. Data Consolidation. - 47 -
Figure 23. Distributions of the Instances Fall into Each Leaf. - 49 -
Figure 24. Output Files Generated by Running the Application. - 53 -

Table 1. Data Sources. - 10 -
Table 2. Accuracy of Forecasts from the Regression Tree Model (RT) and Benchmark Method. - 28 -
Table 3. Hit Rates of the 80% Prediction Intervals for the 1-min and 5-min passenger flows. - 29 -
Table 4. Information of Passengers Connecting to Flight BA74. - 36 -
Table 5. Summaries of the BOSS, BDD, and Conformance Data. - 46 -
The Airport Operation Centre (APOC) has been defined in SESAR (Single European Sky ATM Research) and developed by major European Airports\(^1\). The APOC consolidates different stakeholders in a physical or virtual operations room where existing airport operations applications, such as gate management, de-icing, security, baggage, passenger processes and crisis management, are managed in a collaborative manner.

The APOC has been viewed as the principle support to all major airport decision-making processes. In support of these processes, different stakeholders in the centre integrate information from commonly shared data sets, develop dynamically joint plans, and execute plans that are within their respective area of responsibility. In their current form, existing processes in APOC have been formalized and are supported by Airport Collaborative Decision Making (A-CDM). There remains scope for a strategic assessment of APOC processes and further leveraging the power of operational data.

The main purposes of this project is threefold. First, to review the APOC processes. Second, to identify key processes that could be enhanced by data-driven predictions and by machine learning algorithms (DDP&ML). And finally, to provide a case study to illustrate how shared data and advanced analytics can be used to increase the benefits from an APOC. This project has been undertaken in Heathrow’s APOC, arguably the most advanced APOC in Europe.

Heathrow airport, the largest UK airport, carries over 70 million passengers each year to over 250 destinations worldwide\(^2\). The airport community employs over 70,000 people and operates collaboratively with over 200 stakeholder organizations. In 2015, airport traffic increased by 2.2% to 75 million passengers, with an average load factor of 76.5%\(^3\). Heathrow’s APOC went live on November 12\(^{th}\), 2014. The vision was to house the advanced support systems and processes to enable the deployment of the SESAR Airport Operations Management concept. One year on, Heathrow’s APOC has become the nerve centre of all CDM processes between stakeholders, capturing the benefits of close collaborations, planning, real-time monitoring, proactive decision making and flow management.

Heathrow is the only international hub in the UK, and the vast majority of the flights landing or departing from Heathrow have at least 25% transfer passengers. With so many transfers required, it is critical that the processes involved in the connecting journey are optimized to

---

1. **INTRODUCTION**

---

\(^1\) Eurocontrol, 2010
\(^2\) Heathrow Airport Limited, 2015a
\(^3\) Heathrow Airport Limited, 2015b
ensure Heathrow’s vision to ‘give passengers the best airport service in the world’. Better predictions of passengers’ transfer activities may also improve the accuracy and stability of the Target-Off-Block Time (TOBT), one of the most important parameters in airport’s planning processes. In addition, the APOC has the ability to improve passengers’ experience of the connection journey. Therefore, enhancing related data-intensive analytics activity is worth exploring.

This study begins with a general overview of Heathrow’s APOC processes. We characterize the typical APOC roles and responsibilities, focusing on the key decision processes of the flow managers. We also identify the processes that could benefit from making better use of data.

Based on data availability and the importance of the problem, we decided to focus the second half of the study on improving the processes related to connecting passenger’s activity. Specifically, we develop a prototype model using the regression tree method to forecast transferring passengers’ connection times at Heathrow. This model is built on 3.7 million passenger records collected from three datasets: the Business Objective Search System (BOSS) dataset, the Baggage Daily Download (BDD) dataset, and the Conformance dataset. To generate real-time predictions from the model, we developed an application using Python. Three outputs are provided by running the application: (1) the mean and quantiles of the passengers’ connection times; (2) expected number of late passengers for each outbound flight with its current TOBT and delayed TOBTs; (3) the mean and quantiles of the transfer passenger flow at the Conformance desk.

In addition to providing accurate forecasts, our model can support APOC decision-making processes in three additional ways. First, the regression tree model is helpful for the flow managers who are interested in understanding the key factors that influence passengers’ connection times. Second, the TOBT of an outbound flight could be adjusted according to the predictions of the number of late passengers. Third, these forecasts would allow APOC to make better transfer security lane plans.

Finally, we ran an 8-hour live trial at Heathrow’s APOC to test the feasibility and impact of our model in real time. During the live trial, we generated 2-hour window predictions every five minutes. The trial was successful and the model revealed the potential to better manage passenger flows, with a link to improving TOBT predictability and stability.

The remainder of the report is organized as follows. Section 2 will present an overview of APOC processes. In Section 3 we will provide the framing of the transfer passenger problem. In Section 4, we will elaborate on the predictive model for forecasting transfer passengers’ connection times. In Section 5, we will describe the live trial of the predictive model and report the accuracy of the predictions generated during the trial. In Section 6, we will discuss several decision making processes that could be enhanced by our predictive model. Finally, Section 6 will summarize the project and propose future research opportunities.

---

Heathrow Airport Limited, 2015.
2. APOC OVERVIEW

2.1 Purpose of APOC, Roles and Key Decision Processes

The APOC consists of a variety of airport teams, with the common goal of ensuring ‘Happy passengers, travelling on time, with their bags’. In support of the various decision-making processes, different stakeholders integrate information from commonly shared data sets to develop integrated plans and execute those plans within their respective areas of responsibility.

Roles within the APOC include the airport operations manager (AOM), aircraft flow manager, security flow manager, passenger flow manager, operations lead coordinator, engineering help centre advisor, airport control engineer and baggage service manager (BSM). The APOC roles and their interactions are shown in Figure 2.

In this subsection, we will focus on the key decision processes of the passenger flow manager, the security flow manager and the aircraft flow manager that allow APOC to influence the passenger journey. We assume as part of this study that the passengers will not be delayed by their bags. In most cases it is assumed that the connection time of the bag is faster than the passenger in all but exceptional cases, or that a baggage expedite process support their transfer.

Figure 1. APOC Roles and Airport Interfaces.

---

5 Heathrow document. Provided by Heathrow Integrated Planning and Performance Team.
Passenger Flow Manager

The passenger flow manager sits in the APOC and monitors the flow of passengers through Heathrow’s terminals. They work with terminal-based operations to deploy resources and optimize the passenger experience.

Objectives

The objective of a passenger flow manager is to reduce delay minutes with an efficient use of resources. One delay minute is defined as one passenger waiting for one minute on his journey.

Decisions

Once the passenger flow manager realizes that there is a potential passenger congestion, they will inform relevant teams such as the UK Border Force (UKBF) and Customer Service Team, and recommend actions to resolve the issue. The focus of the passenger flow managers is on providing a proactive response to anything that could happen during the day and ensuring that the relevant teams have enough time to mitigate a problem before it happens.

Information used in making decisions

- Integrated plan received a day before.
- Predictions generated by the Dynamic Modeling for Arrivals and Connections (DMAC).
- CCTV to monitor passenger flow.

Interactions with other partners

The passenger flow manager has two conference calls and two meetings within the APOC per day.

Forecasts of the passenger flows for the next day are received and reviewed each evening. The expected impacts of the passenger flows are communicated to operational teams by the flow manager.

Security Flow Manager

The security flow manager is a member of the APOC security planning team. They manage the resource plan during the day through assessment of resource levels and current forecast demand. Each terminal has its own demand profile which must be individually monitored and managed. This assessment is complicated by unexpected peaks of passenger demand, for example the arrival of a large group.

Objectives

The objective of a security flow manager is to minimize queue “breaches” with the available resource. A breach is categorized as a queue time longer than the area’s service level set out in CAA regulation.

Decisions

For each day, the security resourcing team uses predicted passenger demand to generate an appropriate resource plan. The plan takes account of the day’s resource availability in each terminal and minimize shortfalls through inter terminal moves. During the day, the security flow manager continually assesses the flow of the passengers and identifies any deviations from the plan. The manager then makes decisions, in real time, on how many lanes are required to be open and
whether staff could be better distributed between terminals, taking account of the inter terminal walking time.

**Information used in making decisions**

- Daily operational plan. A forecast is generated by the forecasting team using historical trends and airline booking information. This defines levels of demand for each 15-minute period in each security area. Those profiles are then passed on to the planning team who generate a daily operational plan that specifies how many lanes will be open.
- Predictions generated by DMAC.
- Transview. This British Airways’ tool provides predictions of the number of people who are connecting within British Airways or One World flights over the next 4.5 hours. Transfer passengers with different transfer times are shown in different colours. This system is linked to the Departure Control System (DCS) which records passengers’ boarding pass scans at the Conformance desk in real time.
- Operational Performance Monitor (OPM). The security flow manager uses these data to monitor real-time flow rates in search areas. He also uses this information alongside a record of the day’s rostered staff to make inter-terminal resourcing decisions.
- CCTV to monitor queuing and congestion.

**Interactions with other partners**

The resource plan made by the planning team feeds into the integrated plan. The security flow manager has daily meetings within the APOC and works closely with the passenger flow manager. The security flow manager also makes frequent calls to the resource planning and terminal security managers outside the APOC.

**Aircraft Flow Manager**

The role of the airport flow manager is to provide a continuous stream of aircraft and to balance demand against available capacity. The aircraft flow manager pays keen attention to the condition of global air traffic and weather conditions.

**Objectives**

The objective of an aircraft flow manager is to balance and optimize the flow of aircrafts arriving and departing from the airport.

**Decisions**

The aircraft flow manager can request airlines to change their schedule based on the airfield capacity and weather disruptions. They can also make decisions on amending the stand plan to assist airlines and handlers.

**Information used in making decisions**

- Local weather forecasts provided by the Met Office, including the jet stream, typhoon conditions, snow, fog, etc.
- Air traffic conditions and aircraft en-route performance.
**Interactions with other partners**

Operational conference calls are held with multiple stakeholders, including airlines, the Met Office, Air Traffic Control, and Network Operations Managers. Any requests for schedule change are made directly with airlines.

### 2.2 Data Sources

The data sources available at Heathrow and through APOC can be classified into three categories: flight level data, passenger level data, and forecasts provided by the forecasting team. Descriptions of the first two categories are shown in Table 1. A-CDM, Inter Terminal Operations (ITO) and PTM data sets can be accessed in real time.

We interviewed the forecasting team and learned about the forecasting models they use. This team is responsible for providing the forecasts for the Integrated Plan. (Appendix C shows an example of the connecting passenger flow and inter terminal arriving passenger volumes provided by the Integrated Plan.) They produce flight level forecasts, day level forecasts, transfer passenger flow profiles, and demand profiles, to enable airport stakeholders to make decisions. The flight level forecasts are generated using the seasonal schedule, daily adjustments based on historical data, seasonal trends identified from the last six weeks and any special events. Before a flight arrives at the airport, the number of passengers on the flight will be predicted based on the booking information that Heathrow would have received in the previous day, as well as the historical data from the Business Objective Search System (BOSS) dataset. A transfer flow profile predicting what flights passengers are connecting between is provided to calculate inter-terminal volumes. A security demand profile, forecasting how many passengers will arrive at security in a 15-minute period, is created as well.

### 2.3 Frame and Identify Selected DDP&ML Activity

In order to select the DDP&ML activity to be studied in more detail, we consider both data availability and the importance of the problem. Processes that could be enhanced through data-intensive analytics and decision support include:

- Baggage flow management. Identify processes to reduce the number of misconnecting bags.
- Stand allocation. Optimization of on-the-day stand allocation.
- Passenger flow management. Understand the key factors that influence passengers’ transfer journey. Improve passenger experience and airline departure punctuality.

Compared to departing, arriving, and transfer passengers arriving from domestic origins, transfer passengers arriving from an international origin have a more complex journey and a greater interaction with Heathrow stakeholders. For this reason, the flow management of international transfer passengers was chosen as a focus in this study.

While the impact of late passengers on TOBT at Heathrow appears to be marginal⁶, better predictions of passengers’ connection time could potentially improve the accuracy and stability of TOBT. Moreover, from airside operations’ point of view, more robust connection journeys enable a more stable and accurate TOBT⁶.

---

⁶ Through impact studies conducted by Heathrow Airside operations.
British Airways, which owns 52% of the landing rights at Heathrow, accounts for the largest proportion of connecting passengers at the airport. The data concerning both intra- and inter-terminal connections to Terminal 5 is rich and captures a large number of multi-stakeholder processes.

For the reasons above, the journeys of passengers arriving on international flights and connecting through Terminal 5 were chosen as the focus of this study. These passengers’ outbound flights can be both domestic and international.

*Table 1. Data Sources.*

<table>
<thead>
<tr>
<th>Data Source</th>
<th>Description</th>
<th>Provider</th>
</tr>
</thead>
</table>
| Flight Level Data            | **BOSS**  
• Validated flight information data.  
• Produced every Wednesday for the previous Monday to Sunday.                                                                                           | Heathrow         |
|                              | **CDM (IDAHO)**  
• Arrival  
• Delays  
• Flights  
• Tows  
• Records of all flights that landed or departed on the date.  
• Produced at 5:00 am daily on the subsequent day. Each day a csv file is created.  
• Not every delayed flight is included in the folder. The delayed flight will only be included if the airline specified a particular delayed reason. | Heathrow         |
|                              | **A-CDM (IDAHO)**  
• Departures  
• Arrivals  
• Real time data.  
• Includes the “Movement” and “Transfer” tables.                                                                                                   | Heathrow         |
| Passenger Transfer Conformance Data | **ITO**  
• Available next day.  
• This dataset provides the date, route, time, number of services and number of passengers in 15 minute intervals.                                         | Heathrow         |
|                              | **Passenger Transfer Conformance Data**  
• As the airport have no access to the DCS, the records of passengers’ boarding pass scans are obtained from BA conformance data. This dataset is an extract that is provided by British Airways on a daily basis at the end of the day, to get the time of boarding pass scans. | Heathrow/Airline |
|                              | **PTM**  
• Real time data. Sent when a flight takes off from the departure airport. Each flight sends at least one PTM to the arrival airport.  
• These messages are sent by an airline to the aircraft’s destination airport that list passengers who are known to be transferring onto another aircraft. | Airline          |
3. THE TRANSFER PASSENGER PROBLEM

3.1 Heathrow Transfer Passenger Journey

The transfer passenger journey describes the process of a passenger’s disembarkation at the airport until they arrive at the departure gate of the connecting flight. Based on the passengers’ first points of departure and their final destinations, we divide these journeys into three categories: (i) international inbound connecting to international outbound flights, (ii) international inbound connecting to domestic outbound flights and (iii) domestic inbound connecting to international outbound flights. The connection time of passengers in the first two categories is usually longer.

In 2015, 32% (24 million) of the passengers landing at Heathrow were transfer passengers. The problem we are addressing in this study considers 15%(3.7 million) of the transfer passengers arriving from international flights and departing from Terminal 5. Figure 2 shows an example of the transfer passenger journey and the data collected during the journey.

When a flight takes off from the departure airport, Heathrow receives a PTM that includes transfer passengers’ information. After the flight lands at Heathrow and the passengers disembark, transfer

*Figure 2. The Heathrow Connection Passenger Journey.*
passengers follow purple signs for flight connections. Those transferring to Terminal 5 from one of the other terminals use a shuttle bus. In each Terminal, apart from Terminal 5, a Flight Connections Validation (FCV) desk ensures that the passenger passes through security in the terminal of their departing flight. In Terminal 5, this function is performed by a ‘Ready-to-Fly’ desk.

The shuttle bus between other terminals and Terminal 5 runs approximately every 10 minutes and the capacity of the bus is 53 passengers. The number of passengers who board the bus is counted by sensors and reported as part of an ITO dataset.

Upon arrival at Terminal 5, passengers are required to go through a Conformance desk. If a passenger does not have a boarding pass for the connecting flight, they are provided with one at ticketing desks located in front of the Conformance desks. Staff at the Conformance desk will check the boarding pass to ensure that the passenger is in the right place, their hand baggage meets airline regulations, and they have enough time to catch their onward flight. In the meantime, the passenger’s travel information is recorded and sent to the airline’s DCS. If a passenger at the Conformance desk is unlikely to reach their outbound flight, they will be redirected to a ticket desk for assistance. Terminal 5 has two flight connection areas, one for the ‘International to Domestic’ passengers and the other for passengers departing on international flights.

Passengers connecting to domestic flights need to pass through immigration, and these controls are operated by Border Force. In Terminal 5 these queues are split in two – one for European Union (EU), European Economic Area (EEA), and Swiss nationals, and another for all other nationalities. Security staff checks passengers’ boarding passes, records their information, and enrol them to the common departure lounge process. After this point, passengers progress to the security screening.

Enrolment is not required for passengers connecting to international flights. After passing the Conformance desk, they progress directly to security screening.

Security queue time service levels are monitored through the queue time of a random passenger. Boarding pass information, where scanned, is recorded by airport temporarily for the purpose of queue time measurement.

After moving through airport security, passengers enter the departure lounge, and walk toward their boarding gates. When boarding commences, staff will scan a passenger’s boarding pass and that information will be checked against the airline’s DCS database. If a passenger is travelling to a domestic destination, their information will be checked against that recorded at the common departure lounge enrolment. After boarding is completed, transfer passengers are ready to begin their onward journey.

### 3.2 Problem Framing

Currently, the prototype Dynamic Model of Arrivals and Connections (DMAC) system provides a dynamic view of expected flows versus scheduled expected flows. This system could help managers make decisions for immigration and ITO operations (Figure 3).

As shown in Figure 4, in our chosen DDP&ML activity we focus on the first half of the connection journey. The reasons for this choice are threefold. First, forecasts of each passenger’s arrival time at
the Conformance desk can help airlines predict which passengers are at risk of missing their outbound flights. Given these predictions, the airline would be able to expedite late passengers and Figure 3. The DMAC system.

ensure that TOBT remains stable. Second, better predictions on passengers’ connection times could support making accurate TOBT amendments. Third, the forecasts of the number of passengers arriving at the Conformance desk can support making decisions on allocating security resources.

Next, we define the key objectives, performance indicators, and constraints of this study.

**Key Objectives:** The objective of this study is to develop a model using machine learning techniques to forecast the time difference between a passenger’s arrival at the airport and their arrival at Terminal 5 Conformance desk. Specifically, we focus on the connection journeys of passengers who arrive at the airport on international flights. This model should offer insight as to the key factors impacting passengers’ connection times.

*Figure 4. Key Process to be Predicted and the Available Data Sets.*

7 Heathrow prototype tool. Provided by Heathrow Integrated Planning and Performance Team.
After building this predictive model based on a large historical data set, the APOC can use it to produce both point forecasts and probabilistic forecasts in real time. In addition, distributions of the passenger flow at BA Conformance desk can be created by aggregating these individuals’ distributions.

Performance Indicators: The accuracy of the predictions will serve as the indicator of performance. We evaluate the accuracy of the point forecasts and probabilistic forecasts using the root mean square error (RMSE) and the hit rate, respectively.

Constraints: The first constraint is the availability of real-time data. A variable may appear to be an important predictor when we build the model, but it is not available in real time. Moreover, our forecasting horizons might be limited by the availability of real-time data. For example, suppose a flight is scheduled to land at Heathrow at 8:00 a.m., but the transfer passengers’ travel information is only sent from the airline after 7:00 a.m. Thus, the passenger flow forecasts generated before 7:00 a.m., for the time interval of 8:00 – 8:30 a.m., may not be accurate.

Second, a manager often needs some time to adjust the original plans based on the forecasts. For example, it usually takes a security flow manager more than half an hour to move staffs between terminals. To guarantee that a manager has enough time to make adjustments, the forecasts should be provided to the managers as soon as new data comes in. This requires an effective procedure that can produce accurate forecasts ahead of time.
4. THE PREDICTIVE MODEL

The predictive model is comprised of three parts: collecting historical data, training the model, and generating predictions from the model. Figure 5 gives an overview of the system. Appendix D provides a description of the system architecture.

Figure 5. System Architecture.
4.1 Data Collection

**Data Collection and Consolidation**

To share and analyse data, we set up an Azure standard DS4 Virtual Machine with 28GB memory and 4 cores. We collected data for all of 2015 to train the predictive model.

As shown in Figure 5, the predictive model is built on the data collected from the BOSS, BDD, and the Conformance databases. The BOSS database contains validated flight information data. It is produced on Wednesday for the previous Monday to Sunday. The BDD data is outputted daily from a Heathrow baggage database known as Merlin. It records every piece of connecting baggage through Heathrow on the previous day. Each of the records also contains passenger information, including the passenger’s outbound flight number, outbound seat number, as well as his inbound flight number. The BA Conformance data, provided by BA on a daily basis, is extracted from the DCS data set. As Heathrow APOC has no access to the DCS, the records of passengers’ boarding pass scans in this study are obtained from BA conformance data.

As shown in Figure 6, the three databases are consolidated as follows:

1. BDD data is exported from the MySQL server where the data is stored and managed.
2. BA conformance data is merged from daily CSVs into a single CSV in Microsoft Access.
3. BDD and BA conformance data are joined in Microsoft Access. The new table contains all rows from both data sets as long as there is a match between the dates, outbound flight numbers and passenger seat numbers. If there is no match, the data is discarded.
4. Flight-level information (e.g. body type and geographical region of inbound flight) are added to the new table by mapping inbound flight dates and flight numbers with those in BOSS.

Passengers’ arrival times at the airport (\(t_A\)) and at the Conformance desk (\(t_C\)) are approximated by the “on-chock time” in the BDD data set and the “local conformance time” in the BA Conformance data set, respectively. The target variable of the predictive model – passengers’ connection times \(\Delta t\) – can then be calculated as

\[
\Delta t = t_C - t_A
\]

**Data cleansing**

We notice that about 0.5% (18,893 records) of the passengers are rerouted, which means they travel on a flight other than their intended outbound flights. The BDD data set records the outbound flight seat number that is assigned to a passenger when he checks in. If this passenger is rerouted to another flight, their seat will be reassigned to another passenger. In this case, we may attach \(t_C\) to a wrong passenger, because the seat number recorded in the BA conformance data is collected when a passenger scans a boarding pass at the Conformance desk. Consequently, the calculation of these passengers’ \(\Delta t\) may not be accurate. In addition, the mean and median of these passengers’ \(\Delta t\) are -60min and -56min, respectively. These statistics give us more confidence when removing the rerouted records.

After removing all the rerouted records, we still have 1% (32,723 records) negative \(\Delta t\) in the data set. One possible explanation of these negative values is that a passenger may change the outbound flight seat number after they check in and before they get to the conformance desk. Thus, we remove
all the negative $\Delta t$ in the data file. The mismatch of seat numbers may also cause excessively large connection times. For this reason, we also remove records for which the connection time was greater than the $(1 - x/100)$ quantile, where $x$ is the percentage of negative $\Delta t$.

The size of the resulting data file is 1.2GB. It contains approximately 3.7 million passenger records and 35 variables. (We eliminate the columns that are not relevant to this study, such as inbound flight’s number of engines.) The median and mean of $\Delta t$ are 27.0 min and 30.5 min, respectively.

We also create seven features using our domain knowledge of the data. These new features are:

- **Inbound flight region and outbound flight region.** The airports are grouped into 16 categories based on their geographic regions.
- **Punctuality of the inbound flight.** Punctuality is defined as the time difference between inbound flight’s actual on-chock time and its scheduled arrival time.
- **Hour of the day the inbound flight arrives** at the airport.
- **Perceived connection time.** This feature is calculated as the time difference between the inbound flight’s on-chock time and the outbound flight’s scheduled departure time.
- **Inbound flight load factor and outbound flight load factor.** The load factor is calculated as the ratio of the actual number of passengers to the capacity of the flight.

A full list of the 42 variables is shown in Appendix D Section 5. Note that we only use 33 of them as predictors because (i) six of them (local conformance time, conformance location code, conformance location description, inbound flight’s off-chock time, inbound flight stand number, and outbound flight stand number) are not available in real time, and (ii) three of them (inbound flight number, outbound flight number, and outbound flight seat number) are only used to join different data sets.
4.2 Training the Model

The Classification And Regression Trees (CART) analysis was first introduced in mid 1980s\textsuperscript{8,9}. It uses a decision tree as a predictive model. A decision tree in which the target variable takes real numbers are called regression trees. In most cases, the interpretation of the results summarized in a regression tree is very simple\textsuperscript{10}. This simplicity is not only useful for the purpose of rapid prediction, but can also yield intuitive explanations on why observations are predicted in a particular manner\textsuperscript{9}.

A regression tree is generally built by determining a set of if-then split conditions. It recursively partitions the input space into mutually exclusive regions. Specifically, we start at the root node of a tree, and ask a sequence of questions about the predictors. Which question will be asked next depends on the answers to the previous questions. Note that the variables can be either continuous or categorical. In each iteration, we choose the variable and the split point to achieve the minimum mean squared error (MSE) between the predictions and the realizations. This process will continue until a stopping rule is applied. After a tree is grown, each of the leaves represents one of the partitions of the input space.

To provide a model that can generate accurate predictions and are not over-complicated, we need to find the optimal tuning parameters for the tree. The first tuning parameter for the tree is the maximum tree depth. A very large tree with many leaves might overfit the data, while a small tree might not be able to capture the important structure. The maximum tree depth can restrict the number of layers of a tree. The second tuning parameter is the minimum leaf size. For this study, we need to make sure there are enough data points in each leaf to create a distribution. The minimum leaf size can be used to stop the splitting process when the number of instances in a leaf is too small. In addition, if the tree contains too many variables, it is hard to interpret. Thus, although we have 33 predictor variables in the data set, we may only use a subset to train the model.

We use cross validation to select the maximum tree depth, $d_{\text{max}}$, and the minimum leaf size, $l_{\text{min}}$. One round of cross validation involves dividing the entire data set into training and testing sets, fitting the model to the training set, and validating the model in the testing set. This method is often used for parameter tuning in the field of machine learning. In this study, the tree is fit – for a range of values of the two parameters – to three quarters of the data, and the MSE of the predictions is computed on the remaining one quarter. This is done in turn for each quarter of the data, and the four MSE are averaged. If the average MSE does not reduce significantly when $d_{\text{max}} > d'_{\text{max}}$ and $l_{\text{min}} > l'_{\text{min}}$, then $d'_{\text{max}}$ and $l'_{\text{min}}$ are the optimal values for the tuning parameters.

Specifically, we first train the trees with all 33 variables and different settings of $d_{\text{max}}$ and $l_{\text{min}}$. As show in Figure 7, we find that the MSE drops dramatically as the tree depth increases from one to ten, regardless of the leaf size. After tree depth reaches ten, however, the MSE does not change significantly. On the other hand, a tree with 5,000 minimum leaf size performs slightly better than the trees with 10,000 and 20,000 minimum leaf sizes. We also try to set the minimum leaf size to be less than 5,000, but the model does not improve much. Moreover, if we further reduce the leaf size, we may not have enough instances in the leaves to fit a distribution.

\textsuperscript{8} Breiman et al., 1984.

\textsuperscript{9} Trevor et al., 2009. A detailed and comprehensive description of the CART method can be found in Chapter 9, Section 9.2.2 and 9.2.3.

\textsuperscript{10} James et al., 2013.
To determine the predictors in our final model, we calculate the feature importance of all 33 predictors, and select the top ten most important features as the final predictors. Feature importance is often calculated as the reduction of predictive accuracy when the predictor of interest is removed. Thus, the higher the feature importance, the more important the predictor. Specifically, we fit a tree to the entire data set with maximum tree depth and minimum leaf size set to 10 and 5,000, respectively. We then sort the predictors based on their feature importance and select the first ten as the final predictors.

Next, we retrain the tree with these ten variables. We change the values of $d_{\text{max}}$ and $l_{\text{min}}$ and repeat the cross-validation process described above. The tree with $l_{\text{min}}$ equals 5,000 still performs slightly better than the others, and the MSE does not change significantly after tree depth reaches six. Thus, our final model has ten predictors and is fitted with maximum tree depth and minimum leaf size set to 6 and 5,000, respectively.

Our model divides all the passengers into 47 segments. In other words, the regression tree shown in Figure 8 has 47 leaves. A summary of the 47 segments are provided in Appendix E. The ten predictors and their feature importance are shown in Figure 9. We can interpret the most important predictors as the major factors that play key roles in influencing transfer passengers’ connection times. The five most important factors are:

- **Whether or not the passenger arrives at Terminal 5.** This is the most important predictor in our model. Note that this study focuses on the activities of the transfer passengers whose connecting flight departs at Terminal 5. Thus, passengers who arrives at Terminal 5 certainly spend less time to get to the Conformance desk.

- **The inbound aircraft body type is narrow or wide.** Wide body aircrafts often have more passengers on board. Thus, it may take more time for the transfer passengers to disembark.
Figure 8. The Regression Tree Model for Predicting Passengers' Connection Times.
from the aircraft. This feature might also serve as a proxy for the aircraft stand. Some aircrafts can only park at certain types of stands, and wide-body aircraft at Heathrow typically park at gates in satellites B and C, which require additional travel time for passengers to the main hall of Terminal 5.

- **Perceived connection time.** If the time between connecting flights is short, the passenger will have a sense of urgency, moving through the airport more quickly. As a result, his connection time will be shorter.

- **Inbound flight travel class.** A first/business class passenger can get off the aircraft first. They are in the front of the plain regardless of the aircraft’s body type. They usually have less hand luggage and travel more on their own instead of groups. Thus, their connection times are usually shorter than those of the economy class passengers.

- **Inbound flight stand type.** Note that the inbound flight stand type is automatically recorded as “P” (Pier served stand) in IDAHO. It will be updated to “R” (Remote stand) only if the flight has arrived at Heathrow and the stand is confirmed to be “R”. Transfer passengers on a flight arriving on “R” stand need to walk or take a bus to the terminal building. Thus, they will often have a considerably longer transit time to the connection area.

Given the regression tree, we fit a Gumbel distribution to each leaf. We select the Gumbel distribution for the following reasons. First, most distributions of the instances within a leaf are right skewed. Second, the Gumbel distribution has simple expressions and is easier to fit in the software than the other skewed distributions, such as the Gamma distribution. For a Gamma distribution with a shape parameter $k$, we need to calculate the gamma function $\Gamma(k)$ to obtain its pdf. However, this calculation is limited in many softwares (in Excel the upper limit for $k$ is 109). Third, we have tried to fit several distributions, including the Gumbel, Gamma, and F distributions, and the Gumbel provides the best fit.

A Gumbel distribution has the pdf
\[
 f(x) = \frac{1}{\beta} e^{\left(\frac{x-\mu}{\beta} + e^{\frac{x-\mu}{\beta}}\right)}
\]
for $-\infty < x < \infty$ where $0 < \mu, \beta < \infty$. When we simulate connection times in the next subsection, we only keep positive values. In other words, the connection times are sampled from the Gumbel distributions truncated below zero. The truncated Gumbel distribution is known as the reflection of the Gompertz distribution\textsuperscript{11}. Note that the probability of sampling a negative value $P(x \leq 0)$\textsuperscript{12} ranges from $4 \times 10^{-35}$ (leaf 10) to $5 \times 10^{-4}$ (leaf 43) with an average of $2 \times 10^{-5}$. Thus, very few negative samples can be generated from the 47 distributions.

Figure 10 shows the 47 Gumbel distributions fitted to the terminal leaves. The grey bars represent the histogram of the 3.7 million $\Delta t$ in our training set. We note that the distribution of all the connection times are more spread out than the distributions of the leaves. The shapes of the leaves’ distributions are quite different from each other. In general, the distributions with lower medians are less spread out. This indicates that in these segments, the uncertainties of the passengers’ connection times are low. If there are a lot of passengers in these segments arrive at the airport, the managers should have more confidence on making adjustments to their plans.

The predictive model described in this Section can be obtained by running through the python script provided in Appendix D Section 6.

4.3 Generating Forecasts from the Model

The bottom box in Figure 5 describes how predictions are generated. The detailed algorithm is shown in Alg. 1. Suppose we are at time $t$ and make predictions for passengers’ connection times and the passenger flow in the next $k$ minutes. First, the real-time information of the passengers who arrived in the last 2.5 hours or will arrive in the next $k$ minutes are extracted from IDAHO.

Specifically, the Transfer and the Movement table in IDAHO are joined by matching flight numbers. We use 2.5 hours as a threshold because the maximum connection time in our training set is approximately 2.5 hours. It is fairly unlikely that a passenger will take more than 2.5 hours to

\textit{Figure 10. Distributions of the Instances Fall into Each Leaf.}

\textsuperscript{11} Gompertz, 1825.

\textsuperscript{12} This probability is calculated as the cumulative distribution function of the Gumbel function at 0:

$$F(0) = e^{-e^{\frac{\mu}{\beta}}}.$$
finish his transfer journey. The joined table includes all the predictors required by the predictive model. The algorithm generates three outputs as follows:

**Quantiles of each passenger's connection time**

Given a transfer passenger's real-time information, the regression tree model first determines which leaf this passenger belongs to. For example, if a passenger arrives at Terminal 3 and plans to take an international outbound flight that is scheduled to depart in one hour, then this passenger will fall into leaf 5. Thus, the median of his or her connection time is 34.04 min and the distribution of the connection time can be described by a Gumbel distribution with $\mu = 30.1$ and $\beta = 7.0$. We can also calculate the time at which they will arrive at the Conformance desk by adding the on-chock time of the inbound flight to the $\Delta t$.

**Quantiles of the transfer passenger flow**

A passenger flow profile with a time slice of $r$ minutes is a group of $k/r$ distributions. These distributions describe the number of passengers arriving at the Conformance desk during time intervals $(t, t + r], (t + r, t + 2r], \ldots, (t + k - r, t + k]$. The algorithm samples 500 connection times from each of the passengers' distributions, and calculates the number of passengers arriving at the desk $n_{i,j}$, where $i$ and $j$ denote the $i$-th sample and the $j$-th time interval respectively. The empirical distribution for the $j$-th time interval $(t + (j - 1)r, t + jr]$ are then created using $n_{1,j}, n_{2,j}, \ldots, n_{500,j}$. The quantiles of the number of passengers arriving between $t + (j - 1)r$ and $t + j r$ can be approximated by the quantiles of $n_{1,j}, n_{2,j}, \ldots, n_{500,j}$.

**Expected number of late passengers for each outbound flight**

We group the passengers by their outbound flights, and calculate how many of them are at risk of being late. A passenger is considered to be late if they arrive at the Conformance desk later than 30 min before the outbound flight’s scheduled departure time. We also calculate the number of passengers that would be still at risk if the airline delayed the departure time by 5, 10, 15, and 20 min.

Figure 11 provides an example of the outputs generated at 12:00 p.m. on July 1, 2016. The passenger flows were predicted for the next two hours ($k = 120$ min) and split into 5, 15 and 60-minutes intervals ($r = \{5, 15, 60\}$).
Figure 11. Examples of the Output Files.

Column A: outbound flight number
Column B, C, D and E: Expected number of late passengers with current scheduled time for departure (B) and current scheduled time for departure + 5min(C), +10 min(D), and +15 min(E).

“Late passengers” are defined as the passengers who arrive at the Conformance desk later than 30 min before the outbound flight’s scheduled departure time.

Output files of running the application

Column B: ID of the passenger
Column C: inbound flight on-chock time
Column D: median of the passenger’s arrival time at the Conformance desk.
Column E, F, G, and H: 5th, 10th, 90th and 95th quantiles of Passenger’s arrival time at the Conformance desk.
Column I: Probability of being late.

Predictions of the transfer passenger flow at the Conformance desk

95th quantile median
5th quantile for the number of transfer Passengers arriving in the 15 minutes time interval.

Column A and B: beginning and ending time of the 15 minutes time interval.
Column C: median for the number of transfer passengers arriving in the 15 minutes time interval.
Column D, E, F and G: 5th, 10th, 90th and 95th quantiles for the number of transfer passengers arriving in the 15 minutes time interval.
Alg. 1 Generate distributions of each passenger’s connection time and the number of passengers arriving at the Conformance desk from time \( h \) to time \( h + k \) with a resolution of \( r \).

**Input:**
- \( h \), Current time
- \( k \), Forecasting window
- \( r \), Resolution
- \( n \), Number of passengers who arrived at the airport in the last 2.5 hours or will arrive in the next \( k \) minutes
- \( m \), Number of simulations
- \( w \), Number of different outbound flights
- \( \delta \), The regression tree model
- \( x_1, x_2, \ldots, x_n \), Real-time predictors of each passenger’s connection time.
- \( t_1, \ldots, t_n \), On-chock time of each passenger’s inbound flight
- \( T_1, \ldots, T_w \), Scheduled departure time of each outbound flight

\[
\text{for } i = 1 \text{ to } n \text{ do} \\
\quad l_i \leftarrow \delta(x_1, x_2, \ldots, x_n), \quad \{l_i \text{ is } the \ leaf \ that \ passenger \ i \ belongs \ to\} \\
\quad f_i \leftarrow \text{distribution of leaf } l_i \\
\text{end} \\
\text{for } i = 1 \text{ to } k/r \text{ do} \\
\quad hs_i \leftarrow h + (i-1) \ast r \\
\quad he_i \leftarrow h + i \ast r \\
\text{end} \\
\text{for } i = 1 \text{ to } m \text{ do} \\
\quad \text{for } j = 1 \text{ to } n \text{ do} \\
\quad \quad s_j \leftarrow \text{sample one positive connection time from } f_j + t_j \\
\quad \text{end} \\
\quad \text{for } j = 1 \text{ to } k/r \text{ do} \\
\quad \quad c_{ij} \leftarrow \text{count if } hs_j < s \leq hs_j \\
\quad \text{end} \\
\quad \text{for } j = 1 \text{ to } w \text{ do} \\
\quad \quad \text{for } k \text{ in } [0, 5, 10, 15, 20] \text{ min do} \\
\quad \quad \quad z_{i,j,k} \leftarrow \text{count if } s > T_j + k - 30 \text{ min} \\
\quad \quad \text{end} \\
\quad \text{end} \\
\text{end} \\
\text{Output: mean and quantiles of each } f_1, \ldots, f_n \\
\text{mean and quantiles of each } c_{1,\ldots,k/r} \\
\text{mean and quantiles of each } z_{1,0}, \ldots, z_{w,20}
4.4 Accuracy Assessment and Discussion

In this section we report on the accuracy of the model, when tested on an out-of-sample test set. The test set window was one week long, from 7th to 13th of June 2016. This week is selected because of the availability of the Conformance data. The test set includes records for 36,358 passengers. We evaluate the accuracy of the forecasts using the RMSE and the hit rate. The RMSE is a widely used measure of the accuracy of point forecasts. Lower values of RMSE indicate better, or more accurate, predictions. The hit rate is the percentage of realizations that fall within a central prediction intervals\textsuperscript{13}. It measures the calibration of the probabilistic forecasts. For example, an 80% prediction interval is considered well calibrated if the intervals contain the realizations 80% of the time. A low hit rate indicates overconfidence while a high hit rate indicates underconfidence.

The point forecasts were benchmarked against the average connection times of the passengers arriving from the same inbound flight and same travel class on the same day of week in June 2015. For example, suppose there is a business class transfer passenger arriving from BA774 on Tuesday, June 7, 2016. To calculate her connection time using the benchmark method, we first identify all the business class passengers of BA744 landing on a Tuesday in June 2015. Next, we take the average of these passengers’ actual connection times as the benchmark prediction. Based on the benchmark predictions of each passenger’s connection time, we also count the number of passengers arriving at the Conformance desk within every 15 minutes as the benchmark prediction for the passenger flow.

Figure 1 presents the 15-min interval predictions as well as the actual passenger flows from the 7th to 13th of June 2016. Table 2 shows the RMSE for the point forecasts generated by the regression tree model and the benchmark method. The hit rates for the 80% prediction interval generated by the regression tree model are also provided in the table. The point estimates from the regression tree model are more accurate than the benchmark, across all seven days tested. The prediction intervals for passengers’ connection times are overconfident across all days except for the 8th of June. The current regression tree model was trained and tuned by minimising the errors of the point forecasts. Thus, to improve the hit rate of the model, we may need to use a different objective function that rewards calibration. Other machine learning algorithms, such as the quantile regression forest\textsuperscript{14}, could also be considered to provide better prediction intervals.

The aggregation of each passenger’s distribution may amplify the model’s overconfident effect. A passenger will have an impact on the variability of the number of passengers arriving at the Conformance desk during a time interval \((t_1, t_2]\) only if the range of their connection time contains any boundaries, \(t_1\) or \(t_2\). For example, a passenger whose simulated connection time is between 8:03 and 8:13 a.m. will not affect the variability of the passenger flow from 8:00 to 8:15 a.m., as we are sure that this passenger will arrive during this time interval.

\textsuperscript{13} Grushka-Cockayne et al., 2016.
\textsuperscript{14} Meinshausen, 2006.
Figure 12. Predictions of the 15 min Connecting Passenger Flows from June 7 to 13, 2016.
Table 2. Accuracy of Forecasts from the Regression Tree Model (RT) and Benchmark Method.

<table>
<thead>
<tr>
<th>Date</th>
<th>Accuracy of predictions on connection times</th>
<th>Accuracy of predictions on 15-min passengers flow</th>
<th>Number of PAX tested</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RMSE</td>
<td>Hit Rate (%)</td>
<td>RMSE</td>
</tr>
<tr>
<td></td>
<td>RT</td>
<td>Benchmark</td>
<td>80%</td>
</tr>
<tr>
<td>June 7</td>
<td>25.60</td>
<td>30.34</td>
<td>55</td>
</tr>
<tr>
<td>June 8</td>
<td>32.62</td>
<td>34.69</td>
<td>88</td>
</tr>
<tr>
<td>June 9</td>
<td>27.04</td>
<td>37.08</td>
<td>74</td>
</tr>
<tr>
<td>June 10</td>
<td>25.84</td>
<td>31.79</td>
<td>60</td>
</tr>
<tr>
<td>June 11</td>
<td>20.08</td>
<td>25.58</td>
<td>75</td>
</tr>
<tr>
<td>June 12</td>
<td>33.79</td>
<td>43.31</td>
<td>74</td>
</tr>
<tr>
<td>June 13</td>
<td>24.84</td>
<td>27.70</td>
<td>77</td>
</tr>
</tbody>
</table>

Table 3 provides the hit rates for the 80% prediction intervals of 1-min and 5-min passenger flows. These hit rates are generally higher than those in Table 2. Additionally, the hit rates for the 1-min predictions are higher than those for the 5-min predictions. Thus, it appears that the low granularity of the time interval causes the miscalibration of the probabilistic forecasts.

In addition, future work on improving predictions for passenger flows may consider the dependencies among passengers arriving from the same flight. Our current simulation approach assumes these passengers are all independent from each other. Although we have included flight level information in the predictive model, the passengers’ connection times, however, are likely to depend on other variables relating to their inbound flight, such as the aircraft’s stand number which is not available in real time. Thus, the distributions of different passengers arriving from the same flight predicted by our model may be correlated. One possible way to incorporate the correlation is to use a copula in the simulation. A copula, first introduced in 1970s, is a multivariate probability distribution for which the marginal probability distributions are uniform. Copulas are widely used to describe the dependence between random variables in business applications. Suppose we have n passengers whose connection times are correlated, the simulation of their connection times can be modified as

1. Generate a vector of n dependent random values \( U = \{u_1, u_2, ..., u_n\} \) from a copula.
2. Apply the Gumbel inverse cumulative distribution function on each \( u_i \) to obtain a set of the simulated connection times \( T = \{\Delta t_1, \Delta t_2, ..., \Delta t_n\} \), where \( \Delta t_i = \{\Delta t_i; F_{\text{Gum}}(\Delta t_i|\mu, \beta) = u_i\} \). \( \mu \) and \( \beta \) are parameters of the Gumbel distribution that passenger i’s connection time follows.

Table 3. Hit Rates of the 80% Prediction Intervals for the 1-min and 5-min passenger flows.

<table>
<thead>
<tr>
<th>Date</th>
<th>1-min passenger flow</th>
<th>5-min passenger flow</th>
</tr>
</thead>
<tbody>
<tr>
<td>June 7</td>
<td>67</td>
<td>47</td>
</tr>
<tr>
<td>June 8</td>
<td>66</td>
<td>39</td>
</tr>
<tr>
<td>June 9</td>
<td>66</td>
<td>46</td>
</tr>
<tr>
<td>June 10</td>
<td>66</td>
<td>47</td>
</tr>
<tr>
<td>June 11</td>
<td>66</td>
<td>41</td>
</tr>
<tr>
<td>June 12</td>
<td>70</td>
<td>50</td>
</tr>
<tr>
<td>June 13</td>
<td>71</td>
<td>46</td>
</tr>
</tbody>
</table>

4.5 Retraining the Model

The regression tree model can be easily re-trained once new passenger records are collected. New data is collected every day, as passengers transfer through the connection process, which can then periodically be used to update the model.

In order to update the model, one can simply add the new data to the current training set and run through the Python script provided in Appendix D Section 6. After an update, the optimal tuning parameters, most important predictors and the structure of the tree can be slightly different from those reported in Section 4.2. This updating process can be repeated every six months. Although it may take a few hours to run the entire Python scripts, it only takes a few seconds to fit a single tree. Thus, in addition to running through the entire process every six months, one could only update the regression tree with the tuning parameters and predictors selected in this project on a weekly or monthly basis.

The model will also need to be retrained if airport operations have undergone significant changes, or any new infrastructure has been built. In this case, more predictors would need to be included in the model, and the entire training process would need to be run through again. The resulting regression tree could be very different from the one we have developed in this project.

The procedure of generating predictions as proposed in Section 4.3 remains the same so long as users apply the regression tree method. They can still use the connecting time forecasting application to generate predictions, but the tree file “treeModel.pickle” and the file “coef.csv” that contains parameters of the Gumbel distributions will need to be replaced with the updated ones.

---

17 It takes a Macbook Pro laptop (i7 processor and 16GB RAM) about 20 seconds to (1) load the training set with only 10 significant predictors identified in Section 4.2 and (2) fit a single tree with maximum tree depth and minimum leaf size set to 6 and 5000, respectively.
5. LIVE TRIAL AND ASSESSMENTS

5.1 Real-time Input Data

The real-time data that are used to generate predictions are exported from the IDAHO system. The “Movement table” in IDAHO contains all the required flight level information. We collect the real-time passenger level information from the PTM files. These files are usually sent when a flight takes off from its departure airport. The airline lists passengers who are known to be transferring onto another aircraft in the destination airport. Currently, IDAHO is able to integrate all the PTMs in real time and stack them into a “Transfer table” in the system. We can, therefore, join the Transfer and the Movement table in IDAHO by matching flight numbers.

It should be noted that the actual on-chock time will not be available if the aircraft is still en-route. In this case, we use estimated on-chock time instead. If we have neither of these two variables, we use the actual time of operation (ATO), the estimated time of operation (ETO), or the scheduled time of operation (STO). If none of these five fields are presented in IDAHO, we drop the record.

5.2 The Live Trial

The purpose of the live run was to detect any operational issues that could interfere with the efficacy of the model. For example, we tested if the script was running fast enough and if all the data required by the model were readily available when we needed them.

To conveniently generate predictions in real time, we develop a Python GUI scripting interface that can work in most operation systems (Windows, Linux, Mac, etc.). Appendix D Section 7 and 8 provide the scripts that are coded for the application. Figure 13 shows the interface of the application. A detailed user manual is provided in Appendix D Section 4. We also set up a VBA script to dump and refresh the input file every five minutes.

*Figure 13. Interface of the Connection Time Forecasting Application.*
We update the predictions every 5 minutes because it usually takes a Heathrow machine (HP Elite book 8470p) about 2.5 min to produce the forecasts for the upcoming two hours. The default resolutions are 1, 5, 15, and 60 minutes. The starting time defaults to the current time if the user does not specify one. The ending time will be 24 hours after the starting time.

As shown in Figure 14, the predictions are generated on a rolling basis. Suppose the trial started at 8:00 a.m. We first collected data of the passengers who arrived at Heathrow after 5:30 a.m. or will arrive before 10:00 a.m., and then generate forecasts for the next two hours (8:00 – 10:00 a.m.). Five minutes later (8:05 a.m.), the second trial started. Similarly, we only considered the passengers who arrived at Heathrow after 5:35 a.m. or will arrive before 10:05 a.m., and generated forecasts for the time interval between 8:05 to 10:05 a.m.

Under the default setting, the application outputs one CSV file for the individual-level predictions and four CSV files for the passenger flow predictions. We save the predictions at different resolutions in different files. The predictions include the mean and five commonly reported quantiles: the 0.10, 0.25, 0.5(median), 0.75, and 0.90 quantiles. Plots are also generated to visualize these outputs.

The plots in Figure 15 presents the number of passengers that are expected to show up at the Conformance desk between 12:00 p.m. and 2:00 p.m. The 0.10, 0.50, and 0.90 quantiles are shown in each plot. Forecasts with different granularities are generated for different purposes. The 1- and 5-minutes interval forecasts provide detailed passenger flow profiles. It is clear that there is a small peak around 12:30 p.m. This peak, however, is not easy to see from the other two plots in Figure 15. The 15-minutes interval forecasts can be used to adjust resource plans. Note that the expected number of connecting passengers provided by the Integrated Plan is also offered for 15-minutes intervals. Finally, the predictions at the one-hour granularity provide us an overview of the situation.
We ran the live trial at Heathrow APOC on the 19th of July. It was a “C” day, which means that the expected passenger volume was less than 80% of the annualized average passenger volume. The trial began at 8:00 a.m. and ended at 4:00 p.m. We followed up the trial in the APOC room (Figure 16). The predictions were assessed visually with the camera looking at the flight connection area in Terminal 5 (Figure 17). There was no large discrepancy between the predictions and the realizations identified during the trial. The trial was successful and we were able to continuously generate predictions during the trial.

Figure 16. Assessing the Live Trial in the APOC Room.

Figure 15. Example of Passenger Flow Predictions Generated by the Application.
5.3 Discussion

During the live trial, we observed passenger flows captured by cameras in the connection area. We found that the busyness of the airport may significantly influence a passenger’s connection time. If there are many flights depart or arrive at the same time, the airport could be crowded with passengers. As a result, the transfer passengers may walk slower and wait longer to pass the Conformance desk. In our current model, we include the “hour of the day”, which could partly reflect the airport’s level of busyness, as a predictor. Other variables that could measure the busyness include the type of the day (“A”, “B”, or “C”) and the expected passenger volume. We believe that including such a new variable into our model could be a potential way to improve its forecasting accuracy.

At present, our model can only generate forecasts after Heathrow receives the PTM from the airlines. The PTM is the only real-time data source that contains transfer passengers’ information. Thus, how far in advance we are able to receive these messages may have a significant impact on the accuracy of our predictions. Our predictions may underestimate the passengers flow because of the missing PTMs. In a future model, it may be possible to use the forecasted number of transfer passengers on each inbound flight until the PTM arrives if a sufficiently robust forecast is available.

Figure 18 shows the histogram of the time between the PTM being received and the actual chocks-on time of the inbound flight on July 17th. The PTMs for 95% of the passengers were received with more than 60 minutes ahead of their actual arrivals, 82% with more than 90 minutes, and 68% with more than 120 minutes. Thus, although the forecasting window was set to two hours during the live trial, 90 min might be a better choice. In addition, some issues regarding the format and the quality of the messages may also affect the efficacy of our model. These issues include:

- PTMs are unnecessarily spread over multiple PTM parts.
• Connections for the following day is not always included.
• Connections to other airports, e.g. Gatwick, are included in some PTMs.

In order that Heathrow can optimize the security demand forecast and plan, BA provide Heathrow with conformance data from their Ready-to-Fly desks to inform the turn up profiles of transfers passengers. This conformance data is used to train the model in this study, and is derived from the airline’s departure control system, which during the period of this study was being replaced with a new system. As a result of the transition between the systems, conformance data was not available for the actual live trial day so we are currently unable to test the accuracy of the predictions generated during that day.

It is planned that once conformance data is regularly available, the model will be run again for a day to test both the overall accuracy for the prediction and how the accuracy of the prediction matures in the few hours ahead of time.

Figure 18. PTM Timing Analysis of July 17th.
6. PRESCRIPTIVE ANALYSIS

The prediction of passenger connection time allows for improvements of a number of decision making processes, based within and outside of APOC. Through discussions with APOC roles and Heathrow stakeholders, three key processes were identified for further investigation: TOBT adjustments to reflect passenger delays (including airline decision to depart without passengers), security lane resourcing levels, and inter-terminal coach frequency and routing.

The advantage to each of these processes of better predictions of the connection times, discussed in more detail below, has been assessed according to each process’ own objectives and the objective of reducing connection time variability. More predictable and stable connection time allows for a more predictable and stable input to TOBT decision making.

6.1 TOBT Adjustment According to Passenger Delays

The TOBT is one of the most important parameters in the airport’s planning processes. According to the definition given in SESAR Concept of Operations Step 2 Edition 2014 (Ed. 01.01.00), TOBT is a prediction from the Aircraft Operator (AO) or the Ground Handler (GH), which estimates when an aircraft will be ready to start up or push back immediately upon receiving clearance from the Tower. TOBT is usually generated automatically by the A-CDM platform or determined manually by the AO or GH. During the aircraft turn around, TOBT may be adjusted a few times. A stable and accurate TOBT will increase operational efficiency of the airport stakeholders and help achieve a stable pre-departure sequence.

The anticipated impact of passenger delays on TOBT can be summarized as follows. If the delay of a passenger is not currently predicted accurately, the airline may have to resubmit the TOBT for the outbound flight if the passenger does not arrive at the gate when expected. As a result, the dependent CDM processes are impacted, resulting in the potential for additional delays at the airport and in the network.

Figure 19 illustrates an example of how our predictive model could improve TOBT predictability. BA774 is an international flight going from London Heathrow to Stockholm Arlanda Airport. The scheduled departure time was 9:15 a.m. on July 19. The total number of passengers on the flight was 129, and 43 of them were international transfer passengers arriving from 9 different flights. The passengers on the first flight landed before 6:00 a.m. and had ample time to catch the onward flight. Some other passengers, however, only had limited time. Table 4 summarizes the number of transfer passengers on each flight, the arrival times of the flights, and the top 5 most important predictors for passenger’s connection time.

---

18 Vincent Tempelaar, 2009.
19 Airport A-CDM Team München, 2016.
20 Airport CDM @ FRA, 2016.
As shown in Figure 19 output 1, the predictions of each passenger’s connection time suggest 12% (5 passengers) of them will pass the Conformance desk after 8:45 a.m. (30 minutes before the scheduled departure time of BA744). These passengers will probably miss their connections. The airline can make a decision whether or not to change the TOBT based on the proportion of late passengers. For example, they can set a threshold of 20% and delay the TOBT if there are more than 20% transfer passengers are predicted to be late.

If the airline decides not to change the TOBT, they could identify late passengers by looking at individual connection times (output 2). For example, there is one passenger arriving from BA901 and we expect him or her to pass through the Conformance desk at 8:49 a.m., the median of the

<table>
<thead>
<tr>
<th>Flight</th>
<th>BA012</th>
<th>BA092</th>
<th>BA216</th>
<th>BA058</th>
<th>BA246</th>
<th>BA294</th>
<th>BA116</th>
<th>AA730</th>
<th>BA901</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arrival time</td>
<td>5:47</td>
<td>6:24</td>
<td>6:34</td>
<td>6:55</td>
<td>7:00</td>
<td>7:47</td>
<td>8:01</td>
<td>8:04</td>
<td>8:22</td>
</tr>
<tr>
<td>Number of transfer PAX</td>
<td>4</td>
<td>3</td>
<td>6</td>
<td>1</td>
<td>10</td>
<td>10</td>
<td>6</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>Passenger travel class</td>
<td>E</td>
<td>E</td>
<td>E</td>
<td>E</td>
<td>E</td>
<td>E:2 PAX B:2 PAX</td>
<td>E</td>
<td>E</td>
<td>E</td>
</tr>
<tr>
<td>Departure airport</td>
<td>SIN</td>
<td>YYZ</td>
<td>IAD</td>
<td>CPT</td>
<td>GRU</td>
<td>ORD</td>
<td>JFK</td>
<td>CLT</td>
<td>FRA</td>
</tr>
<tr>
<td>Arriving terminal</td>
<td>T5</td>
<td>T5</td>
<td>T5</td>
<td>T3</td>
<td>T5</td>
<td>T5</td>
<td>T5</td>
<td>T3</td>
<td>T5</td>
</tr>
<tr>
<td>Perceived connection time (min)</td>
<td>208</td>
<td>171</td>
<td>161</td>
<td>140</td>
<td>135</td>
<td>88</td>
<td>74</td>
<td>71</td>
<td>53</td>
</tr>
<tr>
<td>Stand type</td>
<td>Pier</td>
<td>Pier</td>
<td>Pier</td>
<td>Pier</td>
<td>Pier</td>
<td>Pier</td>
<td>Pier</td>
<td>Pier</td>
<td>Pier</td>
</tr>
</tbody>
</table>

Table 4. Information of Passengers Connecting to Flight BA74.
connection time. The probability of arriving after 8:45 a.m. is as high as 0.8. Thus, the airline may need to rebook the passenger onto a later flight.

If the airline decides to delay the TOBT, output 3 in Figure 19 could be used for proposing an accurate TOBT amendment. With the current TOBT, 5 passengers are at risk of being late. If we delay the TOBT by 5 and 10 minutes, two and three more passengers would probably have enough time to catch the flight, respectively. Similarly, with TOBT delayed by 10 min, only 1 passenger would be late. If we amend the TOBT by 20 minutes, however, this passenger would still not be able to catch the flight. Thus, 15 minutes could be a reasonable TOBT amendment for BA774. If our predictions are accurate, the passengers will get to the Conformance desk at the time we expected. Thus, no further changes on TOBT are to be expected because of late transfer passengers, and the stability of TOBT could be improved.

Our predictions can also be used to facilitate early rebooking to other outbound flights for the passengers who are unlikely to meet an acceptable TOBT. For example, output 2 shows that although the passenger arriving from BA901 will probably miss the original connecting flight, he or she has a 90% chance of passing though the Conformance desk before 9:08 a.m. Consequently, the airline could rebook this passenger to the first flight departing after 9:43 a.m.

6.2 Security Lane Resourcing

As described in section 2.1, the objective of the Security Flow Manager is to manage multi-terminal resources to minimize breaches of queue time service levels in security processing areas. These resourcing levels are often defined before the day of operation according to a forecast of demand in 15 min intervals.

Currently, the number of lanes resourced at a processing area is planned according to the predicted inflow of passengers ($n_{inflow}$) in a 15 minutes interval, and the processing speed of an individual security ‘lane’ ($v$):

$$Number\ of\ lanes\ to\ open = \frac{n_{inflow}}{v}$$

It is understood that the predicted inflow of passengers at transfers security is highly variable, because of its dependence on the arrival time of inbound aircraft and the flights that passengers have chosen to connect between. With a real-time input of PTMs and flight arrival information, the model as described in this report is expected to provide a much more accurate forecast of the passenger inflow. This accurate forecast would allow the dynamic planning of the security lane resourcing in Terminal 5 transfers security, by applying the latest forecast demand to the existing lane planning tool. These plans would be monitored and executed by the Security Flow Manager in APOC.

With the Conformance desk data used in this study available only for passengers transferring through Terminal 5, new data regarding connection time to other terminals would be required for its expansion to other security processing areas. To follow the same development methodology of this model, a scan of each passenger’s boarding pass would need to be collected prior to security processing in these areas.
6.3 Inter-Terminal Coach Frequency and Routing

Inter-terminal coaching exists to transport connecting passengers from the terminal of their inbound flight to the terminal of their outbound flight. Heathrow sub-contracts this bussing service, and the buses are operated on a fixed schedule which is adjusted according to historical loads on particular routes. There is limited visibility of real time information of the actual demand on these routes. The historical demand used to guide scheduling decision is calculated from data collected on the occupancy of buses.

Too few buses in a particular time slice on these routes can both increase passenger connection times and introduce large variability, as passenger arrival at outbound terminals is ‘bunched’ according to bus capacity.

Currently, there is no data collection upon arrival at a coaching pick up point. With data collected here, an almost identical model of demand could be created, based similarly on PTMs and real-time flight arrival information. This demand could be used to minimize passenger connection times within the constraints of the number of buses available. For example, low predicted demand on the route between Terminal 4 and 5 coinciding with high predicted demand between Terminals 3 and 5 could allow for the re-routing of a bus from the prior route to the latter.
7. CONCLUSIONS AND FUTURE WORK

7.1 Conclusions

This study demonstrates the use of machine learning techniques to forecast transfer passengers’ connection times. We first built a predictive model for the connection times using the regression tree method, and then developed an approach to generate distributions of each passenger’s connection time and the number of passengers arriving at the Conformance desk. We also developed an application for APOC to produce these forecasts. The application has been used to run an eight-hour live trial on the 19th of July. Finally, we demonstrated how to use the predictions generated from our model to inform TOBT adjustments and determine transfer security resourcing levels.

There are some advantages from using our model. First, the machine learning technique we used to build the model is intuitive and efficient. It can help the managers understand the driving features of the transfer passengers’ connection times. We presented our model to four groups of Heathrow stakeholders right after the live trial. The insights derived from our model have received their enthusiastic support. Second, our model is built on a large historical data set. More than 40 variables are available for selection as predictors. These variables also enable one to build new features using domain knowledge of the data. Third, our model can update the predictions in real time. The application we developed for APOC can easily extract real-time data from the IDAHO. The forecasting procedure is effective and the predictions can be generated in a short amount of time. A study of the PTM qualities shows that there is sufficient data to enable the predictive model to provide forecasts foresight.

Our model is the first to provide forecasts for each transfer passenger. It is also the first to provide probabilistic forecasts. The forecast of a passenger’s connection time may help airlines to predict whether the passenger is at risk of missing his outbound flight. If an airline can retrieve this information far in advance, they may be able to generate a more stable and accurate TOBT. They can also remind a passenger of his or her predicted connection time via the airlines-specific application installed on the passenger’s phone. Based on the forecasts provided by our model, we can easily derive the number of late passengers with the current TOBT and several delayed TOBTs. Given the numbers of late passengers, the airline can decide whether or not to update the TOBT, and how much they should delay the TOBT.

In addition, the passenger flow profiles generated by aggregating individuals’ distributions may help APOC managers make better decisions. A dynamic planning tool can be developed based on the distributions of passenger flows. The outputs of the tool may include a resource plan for the security team as well as a dynamic schedule of the connecting buses.

The benefits of deploying this model are that

- Passenger experience is improved through reduced queuing, as capacity on the inter...
terminal coach operation and at transfer security more closely matches the dynamic demand.

- Airport punctuality is improved as there is increased confidence that passengers will make tight connections.

We also have identified several critical factors that could affect the implementation of big data and machine learning models in the airport:

- Availability and quality of real time data, such as PTM messages, and the consolidation of data from multiple systems within a data warehouse or data lake. A central data management system would help the airport to maintain the consistency of the recorded data.

- Capability and airport understanding to build predictive models using machine learning methods.

- An initial robust forecast demand for both direct and connecting passenger in order to roster the right number of staff to meet the demand, as the real time model can only adjust the resourcing deployment on the day itself.

- Information collaboration across airports and airlines and the further development of industry standards for information exchanges. For example, there is not an industry standard for exchanging airline booking data. If the airline shares transfer passengers’ booking information with the airport, our model will be able to generate forecasts long before the passengers arrive at the airport. The predictions of our model may also add values to the Integrated Plan. More information sharing between the airlines and the airport could help to make stronger joint plans.

- Development of the operational processes and engagement with the operational teams is critical to achieving the business change.

In general, our study indicates that the use of big data and machine learning models have the potential to improve airport operations, through reducing the risk of passenger misconnects and improving TOBT stability and robustness. While our model is developed for a passenger centric problem, we believe the methodology proposed in this report can be easily applied to other airport processes.

### 7.2 Future Work

To achieve the benefits from this new model, it is necessary to develop the processes to respond to the insight provided by the model and to engage the teams to adopt the revised way of working.

To exploit the opportunity to enhance security planning, the security team is required to monitor the big data model, revise their security lane plan to reflect significant changes from the original plan, and then adjust their resource deployment dynamically on the day to deliver the security lane plan. Although the security flow manager can seek to do this subjectively, to fully realise the opportunity brought by this study, the proposed model needs to be integrated with the lane plan and resource management processes.
To exploit the opportunity to enhance the stability of the TOBT, Heathrow needs to work collaboratively with the airlines on the end to end process and team engagement. The methodology to accurately predict the end to end security process time needs to be developed to reflect the wait time, standard process time, possible bag search time and walk time to the gate. It is possible that a further big data machine learning model could be developed to refine this prediction from the simple methodology adopted to date.

If the tool is then providing predictions of passengers’ arrival times at the gate, this information needs to be made accessible to the BA team managing that specific departure, in order to take into account the new insight as they set the TOBT. This is a complex process in itself of which the passenger is just one dimension.

It should also be recognised that different airlines adopt different operational processes and have different priorities. For examples some airlines will prioritise on time departure, and hence will off load passengers which are running late, whilst others will delay departure until all passengers and bags have made their flight. Hence any collaborative process needs to be adaptable to the needs of different airlines.

Robust processes rely on timely quality information, whether that is a boarding pass scan at conformance or measurement of the security waiting time. Bringing all this information together builds more robust models and ensures decisions are made in a joined up manner. Collaboration between airport and airlines is critical to unlocking this information. This is clearly demonstrated for example with airline booking data, which is provided by most airlines to Heathrow to enhance the accuracy of the forecast demand for both direct and transfer passengers.

In conclusion, being able to deploy a big data and machine learning model into the operation is dependent on process and people change, and information collaboration. The live trial has shown the potential which can now be developed and embedded to improve the operation.

Additionally, there are a lot of other processes where big data models could be deployed to better understand the drivers of performance or to manage on the performance. One option being considered is to develop a machine learning model to study the drivers of car park occupancy, and to understand the combination of flights, punctuality and volume which cause car park congestion. Another option is to develop a machine learning model which provides the predicted flow to ticket presentation based on the arrivals turn up profile and check in concourse occupancy, and also potentially the level of congestion on the motorway network. More widely, big data and machine learning models are likely to support us to understand more about the relationships between passengers, bags and the airport performance, and help us identify complex dependencies and trends.

With improved operational models, there is then the potential to use the insight to provide improved information to our passengers, which puts passengers more in control of their journey. There is also an opportunity to use the models to better understand demand versus capacity within the airport.
# Appendix A: Glossary

<table>
<thead>
<tr>
<th>A</th>
<th>AFM</th>
<th>Aircraft Flow Manager</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AOM</td>
<td>Airport Operations Manager</td>
</tr>
<tr>
<td></td>
<td>APOC</td>
<td>Airport Operations Centre</td>
</tr>
<tr>
<td></td>
<td>ATM</td>
<td>Air Traffic Management</td>
</tr>
<tr>
<td></td>
<td>ACDM</td>
<td>Airport Collaborative Decision Making</td>
</tr>
<tr>
<td>B</td>
<td>BOSS</td>
<td>Business Objective Search System</td>
</tr>
<tr>
<td>D</td>
<td>DCS</td>
<td>Departure Control System</td>
</tr>
<tr>
<td></td>
<td>DDP&amp;ML</td>
<td>Data Driven Predictive Process and Machine Learning</td>
</tr>
<tr>
<td></td>
<td>DMAC</td>
<td>Dynamic Modeling for Arrivals and Connections</td>
</tr>
<tr>
<td>E</td>
<td>EU</td>
<td>European Union</td>
</tr>
<tr>
<td></td>
<td>EEA</td>
<td>European Economic Area</td>
</tr>
<tr>
<td>F</td>
<td>FCV</td>
<td>Flight Connections Validation</td>
</tr>
<tr>
<td>I</td>
<td>IATA</td>
<td>International Air Transport Association</td>
</tr>
<tr>
<td></td>
<td>ITO</td>
<td>Inter Terminal Operations</td>
</tr>
<tr>
<td></td>
<td>OPM</td>
<td>Operational Performance Monitor</td>
</tr>
<tr>
<td>P</td>
<td>PASS2</td>
<td>Passenger Authentication Scanning System</td>
</tr>
<tr>
<td></td>
<td>PFM</td>
<td>Passenger Flow Manager</td>
</tr>
<tr>
<td></td>
<td>PTM</td>
<td>Passenger Transfer Message</td>
</tr>
<tr>
<td>R</td>
<td>RMSE</td>
<td>Root mean square error</td>
</tr>
<tr>
<td>S</td>
<td>SESAR</td>
<td>Single European Sky ATM Research</td>
</tr>
</tbody>
</table>
Appendix B: References


Eurocontrol, 2010. The Potential Role of the Airport Operations Centre (APOC) in the SESAR Airport Concept.


APPENDIX C: INTEGRATED PLAN ON ITO AND CONNECTING PASSENGER FLOWS – AN EXAMPLE DAY FOR TERMINAL 5
Figure 20 gives an overview of the system architecture of the DCPFP (Dynamic Connecting Passengers Flow Predictor) system.

**Figure 20. System Architecture.**

APPENDIX D: DCPFP SYSTEM ARCHITECTURE
1. Data Collection

The DCPF system uses three databases to train the model, as described in Table 5. Figure 21 (a) shows part of the flight-level information contained in the Business Objective Search System (BOSS) data. Each record consists of 32 fields, such as the aircraft type and the actual departure/arrival time. A part of the Baggage Download Dataset (BDD) is shown in Figure 21 (b), which records every piece of connecting baggage through the airport. Each of the records also contains an element of passenger information. Figure 21 (c) shows some of the data in the Conformance database. Each row represents a passenger, whose flight information was collected when their boarding pass was scanned at the Conformance desk. A full list of the variables contained in the data set for training the predictive model is shown in Section 5.

Table 5. Summaries of the BOSS, BDD, and Conformance Data.

<table>
<thead>
<tr>
<th>Data File</th>
<th>Format</th>
<th>Description</th>
<th>Owned by</th>
</tr>
</thead>
<tbody>
<tr>
<td>BOSS</td>
<td>csv</td>
<td>• Produced every Wednesday for the previous Monday to Sunday.</td>
<td>Heathrow</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Validated flight information data.</td>
<td></td>
</tr>
<tr>
<td>BDD</td>
<td>MySQL server database</td>
<td>• Created once a day. &lt;br&gt; • Lists every bag that was loaded on the previous day. There is an element of passenger information in each of the bag record.</td>
<td>Heathrow</td>
</tr>
<tr>
<td>Conformance Data</td>
<td>csv</td>
<td>• Created once a day. &lt;br&gt; • Contains individual passengers’ arrival times at the flight connections conformance desk.</td>
<td>Heathrow Airlines</td>
</tr>
</tbody>
</table>

As shown in Figure 22, the three databases are consolidated as follows:

1. BDD data is exported from the MySQL server where the data is stored and managed.
2. Flight connections conformance data is merged from daily CSVs into a single CSV in Microsoft Access.
3. BDD and Conformance data are joined in Microsoft Access. The new table contains all rows from both data sets as long as there is a match between the dates, outbound flight numbers and passenger seat numbers. If there is no match, the data is discarded.
4. Flight-level information (e.g. body type and geographical region of inbound flight) are added to the new table by mapping inbound flight dates and flight numbers with those in BOSS.

Passengers’ arrival times at the airport ($t_A$) and at the Conformance desk ($t_{RTF}$) are approximated by the “on-chock time” in the BDD data set and the “local conformance time” in the Conformance data set, respectively. The target variable of the predictive model – the passengers’ connection times $\Delta t$ – can then be calculated as

$$\Delta t = t_{RTF} - t_A$$

Next, the records of the rerouted passengers\(^{21}\) \(^{21}\) and passengers with negative connection times\(^{22}\) are removed from the joint data set. Suppose $x\%$ of the connection times are negative. Then

\(^{21}\) The outbound flight seat numbers in the BDD and the Conformance data are collected when the passengers check in and arrive at the flight connections conformance desk, respectively. A rerouted passenger’s seat number will be reassigned to another passenger between these two time points. Their records in different data sets cannot be joined by their seat numbers.

\(^{22}\) Negative connection times point to incorrect matches between different data sets. One possible explanation of these wrong matches is that passengers may change their seat numbers after they check in at the departure airport.
passengers whose connection times are greater than the \((100 - x)\)th percentile are also removed from the joint data set\(^{23}\). In addition, the user can remove columns that are deemed not to be relevant.

Finally, seven new features are created from the existing variables as follows:

- **Inbound flight region and outbound flight region.** The airports are grouped into 16 categories based on their geographic regions.
- **Punctuality of the inbound flight.** Punctuality is defined as the time difference between inbound flight’s actual on-chock time and its scheduled arrival time.
- **Hour of the day the inbound flight arrives** at the airport.
- **Perceived connection time.** This feature is calculated as the time difference between the inbound flight’s on-chock time and the outbound flight’s scheduled departure time.
- **Inbound flight load factor and outbound flight load factor.** The load factor is calculated as the ratio of the actual number of passengers to the capacity of the flight.

---

\(^{23}\) Incorrect matches also cause excessively large connection times in the data. The system assumes that passengers whose connection times lie above the \((100 - x)\)th percentile are mismatched among different data sets, where \(x\) is the percentage of the negative connection times.
2. Training the model

The DCPF system uses regression trees to build a predictive model. The user can obtain the predictive model by running through the Python script provided in Section 6. The structure of the procedure, as shown in the middle box in Figure 20, goes as follows:

1. The tuning parameters (maximum tree depth $d_{max}$, minimum leaf size $l_{min}$, and number of predictors $n_{pred}$) are selected using cross validation analysis. One round of cross validation involves dividing the entire data set into training and testing sets, fitting the model to the training set, and validating the model using the testing set. Specifically, the tree is fit for a range of values of the three parameters to three quarters of the data, and the Mean Squared Error (MSE) of the predictions is computed on the remaining one quarter. This is done in turn for each quarter of the data, and the four MSEs are averaged. If the average MSE does not reduce significantly when $d_{max} > d_{max}'$ and $l_{min} < l_{min}'$, then $d_{max}'$ and $l_{min}'$ are the optimal values for the tuning parameters.

2. A tree is fitted to the entire data set with $d_{max}$ and $l_{min}$ set to the values determined in Step 1. Then the predictors are sorted based on their feature importance values and the top $n_{pred}$ are selected as the final predictors.

3. The regression tree is retrained with the $n_{pred}$ predictors selected in Step 2. The cross-validation method described in Step 1 is applied again to find the optimal values of $d_{max}$ and $l_{min}$.

4. A Gumbel distribution is fitted to the instances in each leaf. As an example, Figure 23 shows the 47 Gumbel distributions fitted to the leaves using the data for 2015.

3. Generating Predictions

The bottom box in Figure 20 describes how predictions are generated. The detailed algorithm is shown in Alg. 1. Suppose we are at time $t$ and make predictions for passengers’ connection times and the passenger flow in the next $k$ minutes. First, the real-time information of the passengers who arrived in the last 2.5 hours or will arrive in the next $k$ minutes are extracted from IDAHO. Specifically, the Transfer and the Movement table in IDAHO are joined by matching flight numbers. The joined table includes all the predictors required by the predictive model. The DCPF system generates three outputs as follows:

---

24 A detailed and comprehensive description of the regression tree method can be found in Chapter 9, Section 9.2.2 and 9.2.3. of the book "The elements of statistical learning: data mining, inference and prediction" by Trevor Hastie, Robert Tibshirani and Jerome Friedman.

25 A free machine learning library "scikit-learn" for the Python programming language was used to fit the tree.

26 The optimal $d_{max}$ and $l_{min}$ of the tree fitted to the data for 2015 are 10 and 5000, respectively. The average MSE does not change significantly after the tree depth reaches 10 and the leaf size reduces to 5000.

27 The top ten most important predictors determined based on the data for 2015 include inbound flight terminal, inbound aircraft body type, passenger inbound flight travel class, inbound flight body type, punctuality of the inbound flight, hour of the day the inbound flight arrives, outbound flight destination, inbound flight load factor and inbound flight region.

28 The final predictive model fitted to the data for 2015 has 10 predictors and is fitted with $d_{max}$ and $l_{min}$ set to 6 and 5000, respectively. The regression tree has 47 leaves.
• Quantiles of each passenger's connection time. Given a transfer passenger's real-time information, the regression tree model first determines which leaf this passenger belongs to. The passenger’s connection time $\Delta t$ is predicted by the distribution assigned to this leaf. The time at which this passenger will arrive at the Conformance desk is calculated by adding the on-chock time of the inbound flight to the $\Delta t$.

• Quantiles of the transfer passenger flow. A passenger flow profile with a time slice of $r$ minutes is a group of $k/r$ distributions. These distributions describe the number of passengers arriving at the Conformance desk during time intervals $(t, t+r], (t+r, t+2r], \ldots, (t+k-r, t+k]$. The algorithm samples 500 connection times from each of the passengers' distributions, and calculates the number of passengers arriving at the desk $n_{i,j}$, where $i$ and $j$ denote the $i$-th sample and the $j$-th time interval respectively. The empirical distribution for the $j$-th time interval $(t+(j-1)r, t+jr]$ are then created using $n_{1,j}, n_{2,j}, \ldots, n_{500,j}$. The quantiles of the number of passengers arriving between $t+(j-1)r$ and $t+jr$ can be approximated by the quantiles of $n_{1,j}, n_{2,j}, \ldots, n_{500,j}$.

• Expected number of late passengers for each outbound flight. (A passenger is considered to be late if they arrive at the Conformance desk later than 30 min before the outbound flight's scheduled departure time.) The algorithm also calculates the number of passengers who will still be late if the airline delays the departure time by 5, 10, 15, and 20 minutes.
Alg. 1 Generate distributions of each passenger’s connection time and the number of passengers arriving at the Ready to Fly desk from time $h$ to time $h + k$ with a resolution of $r$.

**Input:**
- $h$, Current time
- $k$, Forecasting window
- $r$, Resolution (time slice)
- $\pi$, Number of passengers who arrived at the airport in the last 2.5 hours or will arrive in the next $k$ minutes
- $m$, Number of simulations
- $w$, Number of different outbound flights
- $\delta$, The regression tree model
- $x_1, x_2, \ldots, x_n$, real time input of passengers’ information
- $\mu, \beta$, Parameters of distributions for each leaf
- $t_1, \ldots, t_n$, On-chock time of each passenger’s inbound flight
- $T_1, \ldots, T_w$, Scheduled departure time of each outbound flight
- $x'_1, x'_2, \ldots, x'_n \leftarrow$ exclude domestic arrival passengers from $x_1, x_2, \ldots, x_n$, and add the same new features as those created in the training set.

for $i = 1$ to $n$ do
  $l_i \leftarrow \delta(x'_1, x'_2, \ldots, x'_n)$, \{ $l_i$ is the terminal leaf that passenger $i$ belongs to \}
  $f_i(\mu_i, \beta_i) \leftarrow$ distribution of leaf $l_i$
end

for $i = 1$ to $k/r$ do
  $hs_i \leftarrow h + (i - 1) \times r$
  $he_i \leftarrow h + i \times r$
end

for $i = 1$ to $m$ do
  for $j = 1$ to $n$ do
    $s_j \leftarrow$ sample one positive connection time from $f_j + t_j$
  end
  for $i = 1$ to $k/r$ do
    $c_{i,j} \leftarrow$ count if $hs_j < s \leq he_j$
  end
  for $j = 1$ to $w$ do
    for $k$ in $[0, 5, 10, 15, 20]$ min do
      $z_{i,j,k} \leftarrow$ count if $s > T_j + k - 30$ min
    end
  end
end

**Output:** save the mean and quantiles of each $f_1, \ldots, f_n$ to a CSV file
save the mean and quantiles of each $c_1, \ldots, c_{k/r}$ to a CSV file
save the mean and quantiles of each $z_{1,0}, \ldots, z_{w,20}$ to a CSV file
plot the mean and quantiles of the passenger flow

To conveniently generate predictions in real time, the system has a Python GUI scripting interface that can work in most operation systems (Windows, Linux, Mac OS, etc.). Instructions on how to use the interface are shown below:

1. Create two new folders named “input” and “output” under the same directory where the script files “executeGUI.py” and “functions.py” are. (These two scripts are provided in Section 7 and 8.) Paste three CSV files into the “input” folder: a) aircraft_type.csv, b) regions.csv, c) UK_IATA.csv.

2. If this is the first time you run this application, open a command prompt (for Windows) or a Terminal window (for Linux and MAC OS). Type pip install apscheduler

and hit enter. Now library “apscheduler” are installed on your machine.

3. Every time you update the regression tree model with a new training set, you will need to copy and paste the pickle file “treeModel.pickle” that stores the tree model, and the CSV file “coef.csv” that contains parameters of the Gumbel distributions to the “input” folder.

4. Join the Transfer and the Movement table in IDAHO by matching flight numbers, and save the new table into a spreadsheet named “input.xlsx”. This spreadsheet should only contain records of the passengers who arrived in the last 2.5 hours or will arrive in the next \( k \) minutes, where \( k \) is the forecasting window selected by the user. Copy and paste the spreadsheet into the “input” folder. Now the “input” folder (See below) contains all the data files required by the model.

5. Open a command prompt (for Windows) or a Terminal window (for Linux and MAC OS), set your working directory as the path of the “executeGUI.py” file. Specifically, type cd /<your directory>

and hit enter.

6. Run the application from the command prompt or the Terminal window. Type python executeGUI.py

and hit enter (See below Figure a). Then a window will pop out and ask you to set several parameters. These parameters include: the forecasting window, number of simulations,

---

29 aircraft_type.csv is used to decide inbound aircrafts’ body types (Wide or Narrow); regions.csv maps the flight’s IATA code to its region; UK_IATA.csv contains all the UK airports’ IATA code and is used to exclude the domestic arriving flights.
quantiles, update frequency, forecast resolution (the time slice), starting time of the first forecasting window (must be a time point in the future and the seconds need to be 00), and ending time of the last forecasting window. The default settings are shown in Figure b. After setting all the parameters, hit the “start” button and wait about 200 seconds before collecting the first set of outputs.

Under the default setting on the interface, the application will update the predictions in every 15 minutes. In each iteration, it will generate predictions for the next two hours. Figure b shows an example of generating 2-hour ahead predictions from 12:00 to 1:00 p.m. on July 1, 2016. The first iteration starts at 12:00 p.m. and generates predictions for 12:00 to 2:00 p.m. The second iteration starts 15 minutes later at 12:15 and generates predictions for 12:15 to 2:15 p.m. Since the “Resolutions” are set to 5, 15, and 60, the expected transfer passenger flow is split into 5, 15, and 60-minutes intervals. The outputs of each iteration include five CSV files and three figures of the expected transfer passenger flow. See Figure 24 for a detailed description of the outputs.

Figure a. Snapshot of the Terminal

Figure b. Interface of the Application
Figure 24. Output Files Generated by Running the Application.

**Expected number of late passengers for each outbound flight**

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ob_displays</td>
<td>+0 min</td>
<td>+5 min</td>
<td>+10 min</td>
</tr>
<tr>
<td>2</td>
<td>BA005</td>
<td>18</td>
<td>11</td>
<td>15</td>
</tr>
<tr>
<td>3</td>
<td>BA007</td>
<td>22</td>
<td>20</td>
<td>17</td>
</tr>
<tr>
<td>4</td>
<td>BA009</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>BA011</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Column A: outbound flight number
Column B, C, D and E: Expected number of late passengers with current scheduled time for departure (B) and current scheduled time for departure + 5min(C), +10 min(D), and +15 min(E).

"Late passengers" are defined as the passengers who arrive at the Conformance desk later than 30 min before the outbound flight’s scheduled departure time.

**Output files of running the application**

![Image of output files]

Column B: ID of the passenger
Column C: inbound flight on-chock time
Column D: median of the passenger’s arrival time at the Conformance desk.
Column E, F, G, and H: 5th, 10th, 90th and 95th quantiles of Passenger’s arrival time at the Conformance desk. Column I: Probability of being late.

**Figures of the transfer passenger flow**

- 95th quantile median
- 5th quantile for the number of transfer passengers arriving in the 15 minutes time interval.

![Image of figures]

**Predictions of the transfer passenger flow at the Conformance desk**

![Image of predictions]

Column A and B: beginning and ending time of the 15 minutes time interval.
Column C: median for the number of transfer passengers arriving in the 15 minutes time interval.
Column D, E, F and G: 5th, 10th, 90th and 95th quantiles for the number of transfer passengers arriving in the 15 minutes time interval.
5. Descriptions of the Variables Used in the Model

<table>
<thead>
<tr>
<th>Variable name</th>
<th>Description</th>
<th>Data set</th>
</tr>
</thead>
<tbody>
<tr>
<td>local_conform_time</td>
<td>Timestamp of when a passenger arrives at Conformance desk.</td>
<td>Conformance Data</td>
</tr>
<tr>
<td>conform_location_code</td>
<td>Code of the conformance desk.</td>
<td></td>
</tr>
<tr>
<td>conform_location_descrp</td>
<td>Terminal number, international or domestic connecting flight, and</td>
<td></td>
</tr>
<tr>
<td></td>
<td>conformance desk number.</td>
<td></td>
</tr>
<tr>
<td>BDDExport_ob_flight_no</td>
<td>Outbound flight number.</td>
<td>Conformance Data</td>
</tr>
<tr>
<td>BDDExport_passenger_seat_number</td>
<td>Outbound flight seat number.</td>
<td>BDD</td>
</tr>
<tr>
<td>passenger_travel_class</td>
<td>Passenger’s inbound flight travel class. There are five types of class</td>
<td>BDD</td>
</tr>
<tr>
<td></td>
<td>in this data sets: C (Business Class), F (First Class), J (Business Class</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Premium), M (Economy/Coach Discounted), and W (Economy/Coach Premium).</td>
<td></td>
</tr>
<tr>
<td>ib_flight_no</td>
<td>Inbound flight number.</td>
<td>BDD, BOSS</td>
</tr>
<tr>
<td>ib_terminal</td>
<td>Inbound flight terminal number.</td>
<td></td>
</tr>
<tr>
<td>off_chocks_time</td>
<td>Inbound flight off-chock time.</td>
<td></td>
</tr>
<tr>
<td>ib_schedule_date</td>
<td>Inbound flight date.</td>
<td></td>
</tr>
<tr>
<td>ob_schedule_date</td>
<td>Outbound flight date.</td>
<td></td>
</tr>
<tr>
<td>ib_stand_type</td>
<td>Stand type of the inbound flight: P (Pier served stand) or R (Remote stand).</td>
<td></td>
</tr>
<tr>
<td>ob_stand_type</td>
<td>Stand type of the outbound flight.</td>
<td></td>
</tr>
<tr>
<td>on_chocks_time</td>
<td>An approximation of passenger’s disembarkation time.</td>
<td>BOSS</td>
</tr>
<tr>
<td>ib_aircraft_type</td>
<td>Inbound aircraft type.</td>
<td></td>
</tr>
<tr>
<td>ob_aircraft_type</td>
<td>Outbound aircraft type.</td>
<td></td>
</tr>
<tr>
<td>ib_aircraft_body</td>
<td>Inbound aircraft body type: W (wide) or N (narrow).</td>
<td></td>
</tr>
<tr>
<td>ob_aircraft_body</td>
<td>Outbound aircraft body type: W (wide) or N (narrow).</td>
<td></td>
</tr>
<tr>
<td>ib_aircraft_class</td>
<td>Inbound aircraft class.</td>
<td></td>
</tr>
<tr>
<td>ob_aircraft_class</td>
<td>Outbound aircraft class.</td>
<td></td>
</tr>
<tr>
<td>ib_orig_dest</td>
<td>Inbound flight original destination.</td>
<td></td>
</tr>
<tr>
<td>ob_orig_dest</td>
<td>Outbound flight original destination.</td>
<td></td>
</tr>
<tr>
<td>ib_passenger_capacity</td>
<td>Capacity of the inbound flight.</td>
<td></td>
</tr>
</tbody>
</table>

30 Not a predictor. We use it to calculate Δt.
31 Not available in real time.
32 Not a predictor. We use it to join data sets.
33 We use it to calculate Δt, and it is a predictor.
<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ob_passenger_capacity</td>
<td>Capacity of the outbound flight.</td>
</tr>
<tr>
<td>ib_passenger_total</td>
<td>Number of passengers on the inbound flight.</td>
</tr>
<tr>
<td>ob_passenger_total</td>
<td>Number of passengers on the outbound flight.</td>
</tr>
<tr>
<td>ib_passenger_transfer</td>
<td>Number of transfer passengers on the inbound flight.</td>
</tr>
<tr>
<td>ob_passenger_transfer</td>
<td>Number of transfer passengers on the outbound flight.</td>
</tr>
<tr>
<td>ib_runway_number</td>
<td>Inbound flight runway number.</td>
</tr>
<tr>
<td>ob_runway_number</td>
<td>Outbound flight runway number.</td>
</tr>
<tr>
<td>ib_schedule_time</td>
<td>Inbound flight scheduled time.</td>
</tr>
<tr>
<td>ob_schedule_time</td>
<td>Outbound flight scheduled time.</td>
</tr>
<tr>
<td>ib_stand_number&lt;sup&gt;34&lt;/sup&gt;</td>
<td>Stand number of the inbound flight.</td>
</tr>
<tr>
<td>ob_stand_number&lt;sup&gt;34&lt;/sup&gt;</td>
<td>Stand number of the outbound flight.</td>
</tr>
<tr>
<td>ob_int_dom</td>
<td>Outbound flight destination: I (International destination) or D (Domestic destination).</td>
</tr>
<tr>
<td>lb_region</td>
<td>Inbound flight region.</td>
</tr>
<tr>
<td>Ob_region</td>
<td>Outbound flight region.</td>
</tr>
<tr>
<td>ib_PlanVsOn_chock</td>
<td>Punctuality of the inbound flight.</td>
</tr>
<tr>
<td>InBoundHour</td>
<td>Hour of the day when the inbound flight arrives at the airport.</td>
</tr>
<tr>
<td>perceived_connection_</td>
<td>Time difference between the inbound flight’s on-chock time and the</td>
</tr>
<tr>
<td>_time</td>
<td>outbound flight’s scheduled departure time.</td>
</tr>
<tr>
<td>ibFlightLoad</td>
<td>Ratio of the actual number of passengers to the capacity of the flight</td>
</tr>
<tr>
<td>obFlightLoad</td>
<td>for inbound flight and outbound flight.</td>
</tr>
</tbody>
</table>

<sup>34</sup> Not available in real time.
6. PYTHON Script — fit the model

# Import libraries.
import pandas as pd
import numpy as np
import pickle
from sklearn.tree import DecisionTreeRegressor, export_graphviz
from scipy.stats import beta, gamma, exponweib, gumbel_r, gompertz, weibull_min
from io import StringIO
import pydotplus
import matplotlib.pyplot as plt
from IPython.display import Image

# Load in the data.
totalDf = pd.read_excel('data-clean1.xlsx')
totalDf2 = pd.read_excel('data-clean2.xlsx')
totalDf3 = pd.read_excel('data-clean3.xlsx')
totalDf4 = pd.read_excel('data-clean4.xlsx')
totalDf = totalDf.append(totalDf2)
del totalDf2
totalDf = totalDf.append(totalDf3)
del totalDf3
totalDf = totalDf.append(totalDf4)
del totalDf4

# Divide 2015 into four quarters.
quart1 = (totalDf['ib_schedule_date'] <= '2015-03-31')
quart2 = (totalDf['ib_schedule_date'] >= '2015-04-01') & (totalDf['ib_schedule_date'] <= '2015-06-30')
quart3 = (totalDf['ib_schedule_date'] >= '2015-07-01') & (totalDf['ib_schedule_date'] <= '2015-09-30')
quart4 = (totalDf['ib_schedule_date'] >= '2015-10-01') & (totalDf['ib_schedule_date'] <= '2015-12-31')

# Create new features
regionsDf = pd.read_excel('regions.xlsx', index_col='IATA')
regDict = regionsDf.to_dict()
totalDf['ib_Region'] = totalDf.ib_orig_dest.apply(lambda x: regDict['Retail Markets'][x])
totalDf['Ob_Region'] = totalDf.ob_orig_dest.apply(lambda x: regDict['Retail Markets'][x])
totalDf['ib_PlanVsOn_chock'] = (totalDf['on_chocks_time'] - totalDf['NewIbSch']).astype('timedelta64[m]')
totalDf['InBoundHour'] = totalDf.on_chocks_time.dt.hour
totalDf['spread'] = (totalDf['NeOlbSch'] - totalDf['on_chocks_time']).astype('timedelta64[m]')
totalDf['ibFlightLoad'] = totalDf.ib_passenger_total / totalDf.ib_passenger_capacity
totalDf['obFlightLoad'] = totalDf.ob_passenger_total / totalDf.ob_passenger_capacity

# Convert categorical variables into dummy variables
DummDf = pd.get_dummies(totalDf, sparse=False)
DummDf = DummDf.drop(['ib_schedule_date', 'ob_schedule_date', 'on_chocks_time'], 1)

del totalDf

# Find optimal tuning parameters
maxDepth = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
minNodeSize = [5000, 10000, 20000]
mse = np.zeros((12,3))
for i in [0,1]:
    for j in [0,1]:
        tree = DecisionTreeRegressor(max_depth = maxDepth[i], min_samples_leaf = minNodeSize[j],
        random_state = 687)
        trainData = DummDf.drop(DummDf.loc[quart1].index)
        model1 = tree.fit(trainData.drop('Delta', axis=1), trainData.Delta)
        testData = DummDf.loc[quart1]
        pred1 = tree.predict(testData.drop('Delta',1))
        mse1 = (np.mean((pred1 - testData['Delta'])**2))**0.5
        del model1; del pred1; del trainData
        trainData = DummDf.drop(DummDf.loc[quart2].index)
        model2 = tree.fit(trainData.drop('Delta', axis=1), trainData.Delta)
        testData = DummDf.loc[quart2]
        pred2 = tree.predict(testData.drop('Delta',1))
        mse2 = (np.mean((pred2 - testData['Delta'])**2))**0.5
        del model2; del pred2; del trainData
        trainData = DummDf.drop(DummDf.loc[quart3].index)
        model3 = tree.fit(trainData.drop('Delta', axis=1), trainData.Delta)
        testData = DummDf.loc[quart3]
        pred3 = tree.predict(testData.drop('Delta',1))
        mse3 = (np.mean((pred3 - testData['Delta'])**2))**0.5
        del model3; del pred3; del trainData
        trainData = DummDf.drop(DummDf.loc[quart4].index)
        model4 = tree.fit(trainData.drop('Delta', axis=1), trainData.Delta)
        testData = DummDf.loc[quart4]
        pred4 = tree.predict(testData.drop('Delta',1))
        mse4 = (np.mean((pred4 - testData['Delta'])**2))**0.5
        del model4; del pred4; del trainData
        mse[i][j] = (mse1 + mse2 + mse3 + mse4)/4
mse

# Print out MSEs for different parameter settings. Find the minimum MSE.
maxDepth = 10
minNodeSize = 5000
# Fit a regression tree model to the entire data set. Find top 10 important features.
tree = DecisionTreeRegressor(max_depth = maxDepth, min_samples_leaf = minNodeSize, random_state = 687)
model1 = tree.fit(DummDf.drop('Delta', axis=1), DummDf.Delta)
# Create a data frame with only the top 10 important predictors.
features=[pair for pair in zip(DummDf.drop(['Delta'],axis=1).columns,model1.feature_importances_) if pair[1]>0]
features = features[0:9]
features
# Go back and repeat Loop a. Find optimal maxDepth and minNodeSize.
maxDepth = 6
minNodeSize = 5000

# Given the optimal tuning parameters and the new data frame, fit a regression tree.
tree = DecisionTreeRegressor(max_depth = maxDepth, min_samples_leaf = minNodeSize, random_state = 687)
modelFinal = tree.fit(DummDf.drop('Delta', axis=1), DummDf.Delta)
# Visualize the tree.
rotation=1
dot_data = StringIO()
out = export_graphviz(modelFinal, out_file=dot_data,
    feature_names=DummDf.drop('Delta',axis=1).columns,rounded=True,
    special_characters=True,proportion=True,rotate=rotation,filled=True,node_ids=True)
graph = pydotplus.graph_from_dot_data(dot_data.getvalue())
#graph.write_pdf("regressionTree.pdf") # Save the figure to your local folder.
Image(graph.create_png())
# Fit Gumbel distributions to the leaves

indexes=modelFinal.apply(DummDf.drop('Delta', axis=1))
DeltasDf=DummDf[['Delta']]  
DeltasDf['leafs']=modelFinal.apply(DummDf.drop('Delta', axis=1))  
params=[{'leaf':[], 'loc':[], 'scale':[]}]
for i, ind in enumerate(set(indexes)):
    #print ('fitting leaf '+ str(ind))
    temp = DeltasDf[DeltasDf.leafs==ind]
    coefs = gumbel_r.fit(temp.Delta)
    params['leaf'].append(ind)
    params['loc'].append(coefs[0])
    params['scale'].append(coefs[1])
paramDf=pd.DataFrame(params)

# Save the tree model and the parameters of the Gumbel distributions.
paramDf.to_csv('coef.csv',index=False)
with open('treeModel.pickle', 'wb') as f:
pickle.dump(modelFinal, f,4)
7. PYTHON Scripts – functions for producing predictions

import pandas as pd
import numpy as np
import pickle
from sklearn.tree import DecisionTreeRegressor
from scipy.stats import gumbel_r
from bokeh.plotting import figure, show, vplot, hplot, ColumnDataSource, output_file, gridplot
from bokeh.models import HoverTool
from pandas.tseries.offsets import *

pd.options.mode.chained_assignment = None

def dist_generator(leaf, trial_number, coefs):
    
    Args:
    leaf: terminal node ID
    trial_number: number of simulations
    coefs: parameters of the Gumbel distribution for each leaf

    Returns:
    an array with shape (1, trial_number) where each value is a random draw from the distribution of the leaf

    
    g_model = gumbel_r(loc=coefs['loc'][leaf], scale=coefs['scale'][leaf])
    sim_results = g_model.rvs(size=trial_number)

    # Find and replace the non-positive samples.
    for i in range(trial_number):
        if sim_results[i] <= 0:
            sim_results[i] = g_model.rvs(size=1)

    return np.ceil(sim_results)

def stat_calc(quantiles, sim_df, freq, columns):
    
    Args:
    quantiles: target quantiles
    sim_df: dataFrame with results of simulation (datetime in rows, number of passengers in columns
    freq: forecast resolution

    Returns:
    DataFrame with statistics from simulation, and some variables needed for making a chart

    
    indexes = []
    for k, ind in enumerate(sim_df.index):
        if k % freq == 0:
            index = ind
            indexes.append(index)
        df_copy = sim_df.copy()
        df_copy.index = indexes
        df_group = df_copy.groupby(df_copy.index).sum()
        df_group['inter_beginning'] = df_group['inter_beginning'].sum()
        # statistics
        quant = quantiles.split(',')
```python
colnames = ['median', 'p'+str(quant[0]), 'p'+str(quant[1]), 'p'+str(quant[2]), 'p'+str(quant[3])]
quant = [float(i) for i in quant]
colnames = ['median', 'p'+str(quant[0]), 'p'+str(quant[1]), 'p'+str(quant[2]), 'p'+str(quant[3])]
df_group[colnames[0]] = df_group[columns].median(axis=1)
df_group[colnames[1]] = df_group[columns].quantile(q=quant[0]/100, axis=1)
df_group[colnames[2]] = df_group[columns].quantile(q=quant[1]/100, axis=1)
df_group[colnames[3]] = df_group[columns].quantile(q=quant[2]/100, axis=1)
df_group[colnames[4]] = df_group[columns].quantile(q=quant[3]/100, axis=1)
df_group.drop(columns, axis=1, inplace=True)
# vars for chart
df_group['hour'] = df_group.index.hour
df_group['minute'] = df_group.index.minute
df_group['inter_end'] = df_group.index + Minute(freq)
df_group['end_hour'] = df_group['inter_end'].dt.hour
df_group['end_minute'] = df_group['inter_end'].dt.minute
return df_group
```
temp = df[df[classPax] > 0]
if len(temp) == 0:
    continue

globals()['df_' + classPax] = pd.DataFrame()
for n in range(1, temp[classPax].max() + 1):
    tempClass = temp[temp[classPax] == n]
    globals()['df_' + classPax] = globals()['df_' + classPax].append([tempClass] * n, ignore_index=True)
globals()['df_' + classPax]['PaxClass'] = i + 1
longDf = longDf.append(globals()['df_' + classPax])

# second set of features for the tree model
longDf['IDAHO_pax_class_3'] = longDf.PaxClass.apply(lambda x: 1 if x == 3 else 0)
longDf['Ib_Region_N America'] = longDf.Ib_Region.apply(lambda x: 1 if x == 'N America' else 0)
longDf['ib_aircraft_body_W'] = longDf.ib_aircraft_body.apply(lambda x: 1 if x == 'W' else 0)
longDf['Ob_Region_Domestic'] = longDf.OB_DICE.apply(lambda x: 1 if x == 'D' else 0)

inputDf = longDf[['TRANSFER_REC_ID', 'ibFlightLoad', 'InBoundHour', 'spread', 'IDAHO_pax_class_3',
                 'IB_TERM5_FLAG', 'ib_aircraft_body_W', 'IB_STAND_TYPE_P_FLAG',
                 'Ib_Region_N America', 'Ob_Region_Domestic']]

passengers = longDf[['TRANSFER_REC_ID','on_chock_best_approx', 'ib_displayed_flyt_no',
                     'ob_displayed_flyt_no', 'spread']]
passengers['leaf'] = model.apply(inputDf.drop('TRANSFER_REC_ID'))

return coefs, passengers

def sim_run(BigStartTime, trialNumber, passengers, columns, coefs, forecast_window, quantiles):
    
    Function sim_run performs simulation and saves individual results
to two csv files

    Args:
    BigStartTime: starting time of the forecast window
    trialNumber: number of simulations
    passengers: pre-processed data frame
    columns: list of columns
    coefs: dictionary with parameters of the distribution for each leaf
    forecast_window: size of forecasting window in minutes

    Returns:
    DataFrame with statistics from simulation, and some variables needed for making a chart.
    Two CSV files - one for each passenger's connection time and another for the expected number
    of late passengers of each outbound flight - will be saved in the "output" folder.

    quant = quantiles.split(',
IbUpperLimit = BigStartTime + Minute(forecast_window)
IbLowerLimit = BigStartTime - Minute(150)
chartDf = pd.DataFrame(index=pd.DatetimeIndex(start=BigStartTime, end=IbUpperLimit, freq='min'))

workingDf = passengers[((passengers.on_chock_best_approx >= IbLowerLimit) & (passengers.on_chock_best_approx <= IbUpperLimit))]
workingDf.reset_index(drop=True, inplace=True)

deltaMins = list(map(lambda x: dist_generator(x, trialNumber, coefs), workingDf.leaf.tolist()))
simOut = pd.DataFrame(deltaMins, dtype=int)
temp = pd.concat([workingDf[['TRANSFER_REC_ID','on_chock_best_approx', 'spread',
                        'ob_displayed_flyt_no']]]);
def results(sim_results, columns, StartTime, quantiles, resolutions=[1, 5, 15, 60]):
    
    Function results provides the aggregated predictions - the transfer passenger flow at the RtF desk. It saves the results in separate CSV files.
Args:
sim_results: data frame with results of simulation (date/time in rows, number of passengers in columns
resolutions: list with time resolutions
columns: list of columns
StartTime: starting time of a forecast window

Returns:
CSV files and some variables that are useful for generating the passenger flow figures.

```python
for r, res in enumerate(resolutions):
globals()[f'chartDf%sm% %res] = stat_calc(quantiles, sim_results, res, columns)
quant = quantiles.split('·')
quant = [float(i) for i in quant]
colnames = ['median', 'p' + str(quant[0]), 'p' + str(quant[1]), 'p' + str(quant[2]), 'p' + str(quant[3])]
columns.insert(0, 'inter_end')
globals()[f'chartDf%sm% %res][colnames].to_csv('output/outPut_%smin_%s.csv' % (res, str(StartTime).replace('"', '_').replace('·', '_'))

return [globals()[f'chartDf%sm% %res for res in resolutions]

def chart(dfs, StartTime, resolutions, quantiles):

Function generates html dashboard

Args:
dfs: data frames with simulation statistics
StartTime: starting time of the forecast window
resolutions: list with time resolutions

```
```python
globals()['p'+str(i)].quad(top='median', bottom=0, left='left', right='right', source=source, color="#6a51a3", fill_alpha=0.80)
globals()['p'+str(i)].quad(top='p05', bottom=0, left='left', right='right', source=source, color="#3f007d", fill_alpha=0.80)
globals()['p'+str(i)].xgrid.grid_line_color = None
globals()['p'+str(i)].ygrid.grid_line_alpha = 0.5
globals()['p'+str(i)].ygrid.grid_line_dash = [6, 4]

hover = globals()['p'+str(i)].select_one(HoverTool)
hover.point_policy = "follow_mouse"
hover.tooltips = [ ("Time interval", " @hour:@minute - @hour_end:@minute_end "),
                    (quant[0] + "% chance to have less than", " @p95(0) pass.")],
                    ("50% chance to have at least", " @median(0) pass.")],
                    (quant[3] + "% chance to have less than", " @p05(0) pass.")]  

plot_list = [globals()['p'+str(plot)] for plot in range(len(dfs))]
plot_list_trans = []
for plot_number, plot in enumerate(plot_list):
    if plot_number % 2 == 0:
        try:
            plot_list_trans.append([plot_list[plot_number], plot_list[plot_number+1]])
        except:
            plot_list_trans.append([plot_list[plot_number]])
    else:
        continue

grid=gridplot(plot_list_trans)

output_file("output/plot%s.html" % str(StartTime).replace(" ", ".replace(".", ",").replace(" ", "_").replace(" ", ""))
show(grid)
```

8. PYTHON Script – create gui application

```python
import pandas as pd
import functions as fc
import time
from apscheduler.schedulers.blocking import BlockingScheduler  # need to install with pip
from pandas.tseries.offsets import *
from tkinter import *
import tkinter as tk
from tkinter import ttk

time.time()
sched = BlockingScheduler()
counter = 0

def get_values():
    forecast_window = forecast_windowVar.get() - 1 # to make window more intuitive. Use can put 120 min.
    trialNumber = trialNumberVar.get()
    resolutions = resolutionsVar.get().split(',')
    resolutions = [int(number) for number in resolutions]
    start = startVar.get()
    end = endVar.get()
    frequency = frequencyVar.get()
    quantiles = quantilesVar.get()

    @sched.scheduled_job('interval', minutes=frequency, start_date=start, end_date=end)
    def main():
        t0 = time.time()
        columns = list(range(trialNumber))
        timePoint = pd.DatetimeIndex(freq='%smin' % frequency, start=start, end=end)
        global counter
        StartTime = timePoint[counter] + Hour(4)  # Day(2)
        print('Working on output for %s' % str(StartTime))
        coefs, passengers = fc.data_processing()
        chartDf = fc.sim_run(StartTime, trialNumber, passengers, columns, coefs, forecast_window, quantiles)
        dfList = fc.results(chartDf, columns, StartTime, quantiles, resolutions)
        fc.chart(dfList, StartTime, resolutions, quantiles)
        t1 = time.time()
        counter = counter + 1
        print('Done with outputs for %s in %s seconds' % (str(StartTime), int(t1 - t0)))

    root.destroy()

root = Tk()
root.wm_title("Heathrow_app")

forecast_windowVar = IntVar()
forecast_windowVar.set(120)

quantilesVar = StringVar()
quantilesVar.set('5,10,90,95')

trialNumberVar = IntVar()
trialNumberVar.set(500)

resolutionsVar = StringVar()
resolutionsVar.set('5,15,60')
```

- 65 -
frequencyVar = IntVar()
frequencyVar.set(15)

startVar = StringVar()
startVar.set(str(pd.Timestamp.now()).split('.')[0])

dayVar = StringVar()
dayVar.set(str(pd.Timestamp.now()+Day()).split('.')[0])

for i in range(7):
globals()['spacer'+str(i)] =ttk.Label(root, text=" " ).grid(column=i, row=0)

for i in range(12):
globals()['spacerR'+str(i)] =ttk.Label(root, text=" ").grid(column=0, row=i)

s = ttk.Style()
s.configure('.', font=('Helvetica', 16))
title = ttk.Label(root, text="Parameters: ", font=('Helvetica', 20)).grid(column=1, row=1, columnspan=1)
title1 = ttk.Label(root, text="Forecasting window (min) ", font=('Helvetica', 14)).grid(column=1, row=3, columnspan=4)
title2 = ttk.Label(root, text="Number of simulations ", font=('Helvetica', 14)).grid(column=1, row=4, columnspan=4)
title3 = ttk.Label(root, text="Resolutions (numbers separated with commas) ", font=('Helvetica', 14)).grid(column=1, row=5, columnspan=4)
title4 = ttk.Label(root, text="Quantiles (numbers separated with commas) ", font=('Helvetica', 14)).grid(column=1, row=6, columnspan=4)
title5 = ttk.Label(root, text="Update frequency (min) ", font=('Helvetica', 14)).grid(column=1, row=7, columnspan=4)
title6 = ttk.Label(root, text="Starting time (YYYY-MM-DD HH:MM:SS) ", font=('Helvetica', 14)).grid(column=1, row=8, columnspan=4)
title7 = ttk.Label(root, text="Ending time (YYYY-MM-DD HH:MM:SS) ", font=('Helvetica', 14)).grid( column=1, row=9, columnspan=4)

F_W = ttk.Entry(root, textvariable=forecast_windowVar, font=('Helvetica', 14)).grid(column=5, row=3, columnspan=1)
T_N = ttk.Entry(root, textvariable=trialNumberVar, font=('Helvetica', 14)).grid(column=5, row=4, columnspan=1)
RS = ttk.Entry(root, textvariable=resolutionsVar, font=('Helvetica', 14)).grid(column=5, row=5, columnspan=1)
QUANTILES = ttk.Entry(root, textvariable=quantilesVar, font=('Helvetica', 14)).grid(column=5, row=6, columnspan=1)
FQ = ttk.Entry(root, textvariable=frequencyVar, font=('Helvetica', 14)).grid(column=5, row=7, columnspan=1)
START = ttk.Entry(root, textvariable=startVar, font=('Helvetica', 14)).grid(column=5, row=8, columnspan=1)
END = ttk.Entry(root, textvariable=endVar, font=('Helvetica', 14)).grid(column=5, row=9, columnspan=1)

RUN = tk.Button(root, text='START', relief='raised', command=get_values, font=('Helvetica', 20)).grid( column=5, row=10, columnspan=1, rowspan=1)

root.mainloop()
print('Script is started at %s' % pd.Timestamp.now())
sched.start()
APPENDIX E: DESCRIPTIONS OF THE 47 PASSENGER SEGMENTS

<table>
<thead>
<tr>
<th>Leaf number</th>
<th>Description of the segment</th>
<th>Median of the Connection time</th>
<th>Fraction of training samples</th>
<th>Parameters of the Gumbel distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>Not arrive at Terminal 5; inbound flight travel class is first/business; connect to an international outbound flight; the connecting flight will depart within the next 88.5 min.</td>
<td>34.04</td>
<td>0.6%</td>
<td>30.1 7.0</td>
</tr>
<tr>
<td>6</td>
<td>Not arrive at Terminal 5; inbound flight travel class is first/business; connect to an international flight; connecting flight will depart between the next 88.5 min and 115.2 min.</td>
<td>35.6</td>
<td>0.7%</td>
<td>31.2 7.3</td>
</tr>
<tr>
<td>7</td>
<td>Not arrive at Terminal 5; inbound flight travel class is first/business; connect to an international outbound flight; connecting flight will depart after 115.2 min.</td>
<td>38.4</td>
<td>0.7%</td>
<td>32.9 8.6</td>
</tr>
<tr>
<td>10</td>
<td>Not arrive at Terminal 5; inbound flight travel class is first/business; connect to a domestic flight; inbound aircraft body type is Wide; connecting flight will depart within the next 108.5 min.</td>
<td>35.7</td>
<td>0.6%</td>
<td>31.7 7.3</td>
</tr>
<tr>
<td>11</td>
<td>Not arrive at Terminal 5; inbound flight travel class is first/business; connect to a domestic flight; inbound aircraft body type is Wide; connecting flight will depart after the next 108.5 min.</td>
<td>38.2</td>
<td>0.5</td>
<td>33.4 7.9</td>
</tr>
<tr>
<td>13</td>
<td>Not arrive at Terminal 5; inbound flight travel class is first/business; connect to a domestic flight; inbound aircraft body type is Narrow; connecting flight will depart within the next 125.5 min.</td>
<td>40.2</td>
<td>1.1%</td>
<td>35.5 8.5</td>
</tr>
<tr>
<td>Row</td>
<td>Description</td>
<td>Code</td>
<td>Delay (min)</td>
<td>Probability (%)</td>
</tr>
<tr>
<td>-----</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>------</td>
<td>-------------</td>
<td>----------------</td>
</tr>
<tr>
<td>14</td>
<td>Not arrive at Terminal 5; inbound flight travel class is first/business; connect to a domestic flight; inbound aircraft body type is Narrow; connecting flight will depart after the next 125.5 min.</td>
<td>44.1</td>
<td>1.1%</td>
<td>38.0</td>
</tr>
<tr>
<td>18</td>
<td>Not arrive at Terminal 5; inbound flight travel class is economy; Inbound flight arrives between 9:30 and 14:30; connecting flight will depart within the next 42.5 min.</td>
<td>38.9</td>
<td>0.9%</td>
<td>34.7</td>
</tr>
<tr>
<td>20</td>
<td>Not arrive at Terminal 5; inbound flight travel class is economy; Inbound flight arrives before 9:30; connecting flight will depart after the next 42.5 min.</td>
<td>43.2</td>
<td>2.9%</td>
<td>37.9</td>
</tr>
<tr>
<td>21</td>
<td>Not arrive at Terminal 5; inbound flight travel class is economy; Inbound flight arrives between 9:30 and 14:30; connecting flight will depart after the next 42.5 min.</td>
<td>47.1</td>
<td>3.8%</td>
<td>41.1</td>
</tr>
<tr>
<td>23</td>
<td>Not arrive at Terminal 5; inbound flight travel class is economy; Inbound flight arrives between 14:30 and 16:30; connecting flight will depart within the next 147.5 min.</td>
<td>39.4</td>
<td>0.7%</td>
<td>34.9</td>
</tr>
<tr>
<td>24</td>
<td>Not arrive at Terminal 5; inbound flight travel class is economy; Inbound flight arrives after 16:30; connecting flight will departure within the next 147.5 min.</td>
<td>36.5</td>
<td>0.8%</td>
<td>32.5</td>
</tr>
<tr>
<td>28</td>
<td>Not arrive at Terminal 5; inbound flight travel class is economy; Inbound flight not departures from North American; Inbound flight arrives before 10:30; connecting flight will depart after the next 147.5 min.</td>
<td>47.0</td>
<td>1.8%</td>
<td>40.6</td>
</tr>
<tr>
<td>29</td>
<td>Not arrive at Terminal 5; inbound flight travel class is economy; Inbound flight not departures from North American; Inbound flight arrives between 10:30 and 14:30; connecting flight will depart after the next 147.5 min.</td>
<td>51.1</td>
<td>1.6%</td>
<td>43.6</td>
</tr>
<tr>
<td>30</td>
<td>Not arrive at Terminal 5; inbound flight travel class is economy; Inbound flight not departures from North American; Inbound flight arrives after 14:30; connecting flight will depart after the next 147.5 min.</td>
<td>41.8</td>
<td>0.6%</td>
<td>36.5</td>
</tr>
<tr>
<td>33</td>
<td>Not arrive at Terminal 5; inbound flight travel class is economy; Inbound flight departures from North American; connecting flight will depart between the next 147.5 min and 266.5min; inbound flight load factor is less than 0.87.</td>
<td>48.4</td>
<td>0.9%</td>
<td>41.3</td>
</tr>
<tr>
<td>34</td>
<td>Not arrive at Terminal 5; inbound flight travel class is economy; Inbound flight departures</td>
<td>53.5</td>
<td>1.3%</td>
<td>45.9</td>
</tr>
<tr>
<td>No.</td>
<td>Conditions</td>
<td>Departure Time</td>
<td>Load Factor</td>
<td>Delay</td>
</tr>
<tr>
<td>-----</td>
<td>-----------------------------------------------------------------------------------------</td>
<td>----------------</td>
<td>-------------</td>
<td>-------</td>
</tr>
<tr>
<td>35</td>
<td>Not arrive at Terminal 5; inbound flight travel class is economy; Inbound flight departures from North American; connecting flight will depart after the next 266.5 min.</td>
<td>56.9</td>
<td>0.9%</td>
<td>14.5</td>
</tr>
<tr>
<td>39</td>
<td>Arrive at Terminal 5; inbound aircraft body type is Narrow; inbound flight travel class is first/business; Inbound flight parks at a Pier stand.</td>
<td>19.8</td>
<td>0.7%</td>
<td>5.5</td>
</tr>
<tr>
<td>42</td>
<td>Arrive at Terminal 5; inbound aircraft body type is Narrow; inbound flight travel class is first/business; Inbound flight parks at a Remote stand; connect to an international outbound flight; connecting flight will depart after the next 266.5 min.</td>
<td>11.4</td>
<td>1.5%</td>
<td>3.8</td>
</tr>
<tr>
<td>43</td>
<td>Arrive at Terminal 5; inbound aircraft body type is Narrow; inbound flight travel class is first/business; Inbound flight parks at a Remote stand; connect to an international outbound flight; connecting flight will depart after the next 266.5 min.</td>
<td>13.2</td>
<td>1.7%</td>
<td>4.9</td>
</tr>
<tr>
<td>45</td>
<td>Arrive at Terminal 5; inbound aircraft body type is Narrow; inbound flight travel class is first/business; Inbound flight parks at a Remote stand; connect to a domestic outbound flight; connecting flight will depart after the next 266.5 min.</td>
<td>13.9</td>
<td>0.9%</td>
<td>4.7</td>
</tr>
<tr>
<td>46</td>
<td>Arrive at Terminal 5; inbound aircraft body type is Narrow; inbound flight travel class is first/business; Inbound flight parks at a Remote stand; connect to a domestic outbound flight; connecting flight will depart after the next 266.5 min.</td>
<td>15.7</td>
<td>1.9%</td>
<td>5.3</td>
</tr>
<tr>
<td>50</td>
<td>Arrive at Terminal 5; inbound aircraft body type is Narrow; inbound flight travel class is economy; connecting flight will depart within the next 102.5 min; Inbound flight parks at a Remote stand; inbound flight load factor is less than 0.74.</td>
<td>24.0</td>
<td>0.6%</td>
<td>5.9</td>
</tr>
<tr>
<td>51</td>
<td>Arrive at Terminal 5; inbound aircraft body type is Narrow; inbound flight travel class is economy; connecting flight will depart within the next 102.5 min; Inbound flight parks at a Remote stand; inbound flight load factor is greater than 0.82.</td>
<td>21.2</td>
<td>0.5%</td>
<td>6.8</td>
</tr>
<tr>
<td>53</td>
<td>Arrive at Terminal 5; inbound aircraft body type is Narrow; inbound flight travel class is economy; connecting flight will depart within the next 21.5 min; Inbound flight parks at a Pier stand.</td>
<td>15.1</td>
<td>2.1%</td>
<td>4.5</td>
</tr>
<tr>
<td>Flight</td>
<td>Inbound Aircraft</td>
<td>Inbound Flight Travel Class</td>
<td>Connecting Flight Details</td>
<td>Inbound Flight Load Factor</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------</td>
<td>-----------------------------</td>
<td>---------------------------</td>
<td>----------------------------</td>
</tr>
<tr>
<td>54</td>
<td>Narrow</td>
<td>Economy</td>
<td>Depart between 21.5 min and 102.5 min; Pier stand.</td>
<td>0.82%</td>
</tr>
<tr>
<td>57</td>
<td>Narrow</td>
<td>Economy</td>
<td>Depart after 102.5 min; Remote stand; Load factor less than 0.81.</td>
<td>0.7%</td>
</tr>
<tr>
<td>58</td>
<td>Narrow</td>
<td>Economy</td>
<td>Depart after 102.5 min; Remote stand; Load factor greater than 0.81.</td>
<td>0.6%</td>
</tr>
<tr>
<td>60</td>
<td>Narrow</td>
<td>Economy</td>
<td>Depart after 102.5 min; Pier stand; Load factor less than 0.76.</td>
<td>0.36%</td>
</tr>
<tr>
<td>61</td>
<td>Narrow</td>
<td>Economy</td>
<td>Depart after 102.5 min; Pier stand; Load factor greater than 0.76.</td>
<td>0.6%</td>
</tr>
<tr>
<td>66</td>
<td>Wide</td>
<td>First/Business</td>
<td>Connect to international; Depart within 92.5 min.</td>
<td>0.21%</td>
</tr>
<tr>
<td>67</td>
<td>Wide</td>
<td>First/Business</td>
<td>Connect to international; Depart between 92.5 min and 152.5 min.</td>
<td>0.15%</td>
</tr>
<tr>
<td>69</td>
<td>Wide</td>
<td>First/Business</td>
<td>Connect to domestic; Depart within 22.5 min.</td>
<td>0.07%</td>
</tr>
<tr>
<td>70</td>
<td>Wide</td>
<td>First/Business</td>
<td>Connect to domestic; Depart between 22.5 min and 152.5 min.</td>
<td>0.68%</td>
</tr>
<tr>
<td></td>
<td>Event Description</td>
<td>Delay (min)</td>
<td>Probability (%)</td>
<td>Total Delay (min)</td>
</tr>
<tr>
<td>---</td>
<td>-----------------------------------------------------------------------------------</td>
<td>-------------</td>
<td>------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>73</td>
<td>Arrive at Terminal 5; inbound aircraft body type is Wide; inbound flight travel class is economy; connecting flight will depart within the next 19.5 min.</td>
<td>22.8</td>
<td>2.1%</td>
<td>19.0</td>
</tr>
<tr>
<td>74</td>
<td>Arrive at Terminal 5; inbound aircraft body type is Wide; inbound flight travel class is economy; connecting flight will depart between the next 19.5 min and 94.5 min.</td>
<td>28.5</td>
<td>10.9%</td>
<td>23.6</td>
</tr>
<tr>
<td>76</td>
<td>Arrive at Terminal 5; inbound aircraft body type is Wide; inbound flight travel class is economy; connecting flight will depart after the next 94.5 min; Difference between actual on-chock time and scheduled arrival time is less than 30.5 min.</td>
<td>29.3</td>
<td>4.2%</td>
<td>24.2</td>
</tr>
<tr>
<td>77</td>
<td>Arrive at Terminal 5; inbound aircraft body type is Wide; inbound flight travel class is economy; connecting flight will depart after the next 94.5 min; Difference between actual on-chock time and scheduled arrival time is greater than 30.5 min.</td>
<td>35.4</td>
<td>4.4%</td>
<td>28.6</td>
</tr>
<tr>
<td>81</td>
<td>Arrive at Terminal 5; inbound aircraft body type is Wide; inbound flight travel class is first/business; connect to an international outbound flight; Difference between actual on-chock time and scheduled arrival time is less than 27.5 min.</td>
<td>23.7</td>
<td>1.0%</td>
<td>19.3</td>
</tr>
<tr>
<td>82</td>
<td>Arrive at Terminal 5; inbound aircraft body type is Wide; inbound flight travel class is first/business; connect to an international outbound flight; Difference between actual on-chock time and scheduled arrival time is greater than 27.5 min.</td>
<td>26.3</td>
<td>1.0%</td>
<td>21.0</td>
</tr>
<tr>
<td>84</td>
<td>Arrive at Terminal 5; inbound aircraft body type is Wide; inbound flight travel class is first/business; connect to a Domestic outbound flight; inbound flight load factor is less than 0.87.</td>
<td>27.0</td>
<td>0.7%</td>
<td>22.2</td>
</tr>
<tr>
<td>85</td>
<td>Arrive at Terminal 5; inbound aircraft body type is Wide; inbound flight travel class is first/business; connect to a Domestic outbound flight; inbound flight load factor is greater than 0.87.</td>
<td>29.7</td>
<td>1.3%</td>
<td>24.5</td>
</tr>
<tr>
<td>88</td>
<td>Arrive at Terminal 5; inbound aircraft body type is Wide; inbound flight travel class is economy; connecting flight will depart within the next 281.5 min; Difference between actual on-chock time and scheduled arrival time is less than 27.5 min.</td>
<td>35.4</td>
<td>4.5%</td>
<td>28.2</td>
</tr>
<tr>
<td>89</td>
<td>Arrive at Terminal 5; inbound aircraft body type is Wide; inbound flight travel class is economy; connecting flight will depart within the next 281.5 min; Difference between actual on-chock time and scheduled arrival time is greater than 27.5 min.</td>
<td>40.0</td>
<td>4.6%</td>
<td>31.7</td>
</tr>
</tbody>
</table>
on-chock time and scheduled arrival time is greater than 27.5 min.

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
<th>Time</th>
<th>Load Factor</th>
<th>Departure Time</th>
<th>Load Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>91</td>
<td>Arrive at Terminal 5; inbound aircraft body type is Wide; inbound flight travel class is economy; connecting flight will depart after the next 281.5 min; inbound flight load factor is less than 0.80.</td>
<td>39.1</td>
<td>8.0%</td>
<td>30.9</td>
<td>12.5</td>
</tr>
<tr>
<td>92</td>
<td>Arrive at Terminal 5; inbound aircraft body type is Wide; inbound flight travel class is economy; connecting flight will depart after the next 281.5 min; inbound flight load factor is greater than 0.80.</td>
<td>44.1</td>
<td>3.0%</td>
<td>35.2</td>
<td>14.0</td>
</tr>
</tbody>
</table>