190810 - Advanced Profiling

Technical Report for CEBRA project 190810

Natasha Page[1], Nathaniel Bloomfield[1], Sam Zhao[2], Emmanuel Esguerra[2], Jose Arias[2], John Baumgartner[1], and Andrew Robinson[1]

[1]The Centre of Excellence for Biosecurity Risk Analysis, The University of Melbourne
[2]Compliance Division, Department of Agriculture, Water and the Environment

April 28, 2022

cebra
Centre of Excellence for
Biosecurity Risk Analysis

Contents

List of Figures

List of Tables

1. Executive summary

1.1. Key Points

1. Working with the department, CEBRA has developed a new way of analysing passenger inspection data to determine passenger cohort profiles.

2. The new way is easier to automate, easier to implement, and results in better profiles than the previous approach.

3. The department should adopt the new approach recommended here for profiling international passengers.

4. The approach recommended here should also be suitable for profiling international mail but formal demonstration is beyond the remit of the current project.

1.2. Background

International passengers and mail are two high-profile pathways that may carry actionable biosecurity material (ABM) across national borders. Both pathways enjoy considerable activity, with many millions of passengers and mail articles arriving every year. This activity presents a substantial challenge to the Department of Agriculture, Water and the Environment, upon which the responsibility for border biosecurity rests.

The department uses statistical models of the contamination probability associated with cohorts of passengers and mail to determine the best level of intervention across the pathway. These statistical models are constructed using a set of ad-hoc tools that were developed by the Australian Centre for Excellence in Risk Analysis (ACERA), and are described in Robinson *et al.* (2015).

A particular challenge addressed by these methods is missing data: in most ports, the number of each cohort of passengers or mail that undergo screening is unknown, so the probability that a cohort carries ABM cannot be directly calculated. Specifically, the nil-find screening and inspection outcomes are incompletely captured, which if ignored would lead directly to corrupt profiles. To correct for this shortcoming, the number screened is estimated by using the rates observed in the department's endpoint survey, scaled to the overall cohort and screening counts (via *raking*, see Appendix B for definition), and then the final interception rate estimates are smoothed to improve their predictive abilities (by a tool called *Empirical Bayes*).

1.3. Innovation

The project builds on the previous work as follows. Whereas previously, the focus was to try to estimate the number of passengers of each cohort within each intervention channel by proportional scaling, the recommended approach formally fits statistical models that predict the probability of being screened for individual cohorts. These probabilities are then used to estimate the number screened of each cohort, which are then used to construct models that predict the probability that the cohort is carrying ABM. These innovations enable the use of a more sophisticated and reliable suite of statistical tools in place of the previous approaches, with the new method finding approximately 20% more non-compliance. The new approaches are also deal with edge cases, such as cohorts that lack data, in less arbitrary ways.

Briefly, then, previously the set of processing steps was: (i) apply scaling to the endpoint survey data, (ii) construct cohort-level estimates of contamination rate and then (iii) smooth the cohort-level estimates (by Empirical Bayes). We now recommend the following steps to construct the profiles: (i) apply smoothing to the endpoint survey data by fitting a statistical model, (ii) apply scaling (via *raking*) to the predicted screening rates obtained from the endpoint survey data, (iii) construct cohort-level estimates of contamination rates and then (iv) smooth (by another statistical model). Figure 4.1 provides a comparison.

Our reasons are explained in detail in the report, but briefly, step (i) helps solve the problem of sparsity in the endpoint data that led to earlier difficulties with raking, step (ii) compensates for missing nil-finds in the MAPS dataset, step (iii) is the same as previous, and step (iv) provides data-driven support to find the best mix of profiling information. Furthermore, models help solve the problem of sparsity by using the data to identify the key attributes that contribute to non-compliance, and extrapolate these to less well represented combinations of attributes.

Also as a part of this project, we provide R code for profile automation to show how the new method can be integrated into the existing system as well as how the Department can follow the analysis techniques used in this project to consider alternative models or variables in the future.

1.4. Other Details

The proposed approach also avoids a problem that was characteristic of the previous approach, namely the endpoint survey data was commonly sparse, with many zeros, which made scaling by raking difficult.

We also explore an alternative approach that avoids attempting to estimate screening rates altogether, and uses the endpoint survey and a statistical model to directly estimate the non-compliance rates of passengers. Although this simpler method performs well, the multi step method has the stronger performance and so is our ultimate recommendation.

We consider a number of problem variations to determine the optimal model as well as a number of key aspects to practical implementation.

2. Introduction

Over 20 million international passengers arrive in Australia each year. Each of these passengers may be bringing with them actionable biosecurity material (ABM), which has the potential to detrimentally impact Australia's agricultural industry and environment. This project examined the problem of profiling air passengers with regard to their probability of carrying undeclared ABM. Passengers carrying undeclared ABM are considered to be non-compliant, as they have failed to declare that they are carrying items of biosecurity concern at the border.

The current approach is to divide the passengers into similar cohorts, using information on a passenger's flight number, passport country, age and gender. (For example, a cohort might be females between the age of 18–30, with a US passport on flight AB113.) Then, using previous inspection data, the likelihood of a passenger from each of these passenger cohorts being non-compliant is estimated. The analysis to construct the cohort profiles is carried out in advance of the flight arrivals to inform which passengers should be screened upon arrival each day. The goal of the project was to improve the method currently used to predict the non-compliance rate for a given cohort, using the same data inputs.

Profiles are constructed using statistical models of non-compliance likelihood fitted to border inspection data. Obtaining representative data is a significant challenge in the construction of profiles. If the data collection is biased, then profiles can either be misleading if negative finds are not taken into account, or suffer from poor representation of some regions (Robinson *et al.*, 2015; Suresh & Guttag, 2019). The biosecurity risk management approach for international passengers (and mail) is primarily tactical: the focus is on managing the contemporary biosecurity risk. This means that for the most part, intervention is carried out where the biosecurity risk is held to be the highest. As a consequence, the inspection data are not a representative sample of the population, and this issue is compounded by the fact that nil-finds (that is, inspections or screening that do not detect ABM) are not consistently recorded.

Therefore, in order to also secure strategic benefits from the inspection data, we need to recover the effort that has gone into inspecting each cohort. This can be estimated from summary statistics of the operation, such as passenger cohort counts, intervention counts, and so on. Previously this information gap has been remedied using a set of ad-hoc corrections as documented in Robinson *et al.* (2015).

Once the relative risk of different cohorts is known, the challenge is to allocate intervention resources among those cohorts. This is analogous to the challenge of exploitation and exploration (Chouldechova & Roth, 2018). Exploration vs. exploitation is a well-known problem in reinforcement learning (Sutton & Barto, 2011), where an algorithm must *explore* areas of the domain space that have not previously been searched as well as *exploiting* areas that appear to be optimal or close to optimal. This is a way of avoiding getting caught in local optima. Finding the equilibrium between the two is tricky and the ideal balance will vary between problems. This problem is parallel to ours because we only get new data to inform the model if a passenger is found to be carrying ABM or if they are selected in the endpoint survey. So, if we ignore the exploration side of the problem, then we may find ourselves trapped in a local minima whereby we continue to sample from the same cohorts and therefore find ABM only in these cohorts, potentially drawing to the mistaken conclusion that these are the cohorts that offer the most risk to the pathway — a *self-fulfilling prophecy*.

This project examined the utility of a variety of techniques for solving this problem. We used the existing method, which involves a combination of raking (Deming & Stephan, 1940) (see Appendix B for definition) and empirical Bayes estimation (Carlin & Louis, 2008) as a baseline, and we further considered the benefits of using the output from a Generalised Linear Model (Hastie, 1992) or a Gradient Boosting Machine (Friedman, 2001) as the seed for raking, followed by a statistical model to predict non-compliance. We also consider methods that avoid the issue of predicting a screening rate and simply use the endpoint survey to fit a model for non-compliance.

The remainder of this Chapter 2 formally specifies the problem and examines previous work that has been done in this area. Chapter 3 reports the datasets and the limitations that they bring to the problem. In Chapter 4, we define the models that we will be considering for profiling and compare their performance. Chapter 5 looks at how we might choose a cutoff for who should get screened while Chapter 6 looks at how varying the size of the endpoint survey affects our results and Chapter 7 examines ways that we might spot if we need to update the profiles. Chapter 8 describes how the profiles may be automated and the results of this project might be put into practice and Chapter 9 provides a summary of how version control can be used. Finally, Chapter 10 provides discussion and conclusions.

2.1. Problem Specification

We categorise passengers according to certain characteristics to define a cohort. The characteristics that we use to profile are:

- citizenship country,
- gender,
- age, and
- flight.

These are the passenger characteristics that are presently used for constructing cohorts in the existing workflow. It is of course possible to construct profiles with a wider range of features, but this was not in the remit of the project.

We might define a particular cohort as, for example, *Australian males, aged 30 on flight AB13*. We would say that a passenger belongs to that cohort if they fit each of those characteristics.

Upon arrival at the airport, each arriving passenger will go through one of:

1. the X-ray channel,
2. the detector-dogs unit (DDU) channel,
3. the manual inspection channel, or
4. the direct exit channel.

The channel nomination depends on (among other things) the cohort to which the passenger belongs, and the cohorts are prioritized for screening by means of *profiles*, which are statistical models of the biosecurity risk represented by the cohorts. For the purposes of this report, we describe channels 1 to 3 as *screening channels* and channel 4 as the *exit channel*.

We say that a passenger is *non-compliant* if they are carrying undeclared ABM. We wish to predict, for any given passenger, the likelihood that they are non-compliant based solely on the above characteristics. We do this by calculating a non-compliance rate (NCR) for each cohort. The NCR, in its simplest form, can be calculated as:

$$\text{NCR} = \frac{\text{number of non-compliant passengers found during screening}}{\text{number of passengers screened}}. \tag{2.1}$$

In Section 4.1.2, we investigate more sophisticated measures of non-compliance and compare them to this simple non-compliance rate. The definition of the non-compliance measure above as well as those in Section 4.1.2 are as described in previous ACERA project 1101D[1], although we use *non-compliance rate* where they use *non-compliance effectiveness*.

If we could calculate an NCR for each cohort, then in theory we could use a greedy allocation approach whereby we would simply work our way down the list of cohorts, going from highest NCR to lowest NCR, assigning passengers to the screening channels until we ran out of capacity on these channels, and then assign the remaining passengers to the exit channel. This approach, at the very least, takes care of the exploitation side of the problem.

Unfortunately, finding the number of passengers inspected (or *inspection effort*) for a cohort is not a simple task, because passengers are not identified by cohort within the intervention channel unless they are non-compliant. Consequently there is no way to estimate the cohort risk just using these data.

However, the department takes a survey of passengers after intervention, called an *endpoint survey*, which captures extra detail and further intervention data. We can use the endpoint survey data to predict which channel a passenger cohort would have gone through. The endpoint survey dataset is much smaller than the interception dataset, and so there will likely be advantages in using it to estimate the implied screening rates that we infer that the interception dataset would have been drawn from. We have information on the total inspection effort across the pathway as a whole as well as the overall volume of passengers in a cohort, and we can use this information to inform a prediction of cohort-specific screening effort, by raking which we define in Appendix B.

A further complication is that the data can not provide an unequivocal view of what might occur under different profiling regimes, hence, any desktop assessment of the profiles or profiling methodology is fraught. This is because the data capture is undertaken based on the profiles. The two main sources of inspection data are the routine screening of high-risk cohorts, and the manual inspection of items in the endpoint survey. The outcomes of the routine screening cannot be assumed to be representative of the unscreened population, for the very reason that the cohorts were selected because

[1] Robinson et al (2013) *Adoption of meaningful performance indicators for quarantine inspection performance*, ACERA Final Report for 1101D

they were expected to be higher risk. Furthermore, the endpoint survey data are collected using a more stringent intervention measure than just screening (that is, at least one bag is opened), whereas not all passengers that undergo screening are inspected.

2.2. Previous work

The methods examined in this project build upon the techniques already employed by the Department.[2] These were developed following Robinson *et al.* (2015) who demonstrated how raking can be used to improve the estimate of cohort counts within channels and how empirical Bayes smoothing can be used to reduce variability in estimates for cohorts with low representation in the endpoint survey. Robinson *et al.* (2015) applied the methods to predicting the NCR within arriving international mail; this problem is very similar to predicting the NCR within air passengers because much of the data is similar and many of the issues are the same such as sparse data within cohorts in the endpoint survey.

Later, Lane *et al.* (2017) evaluated the effectiveness of a variety of machine learning methods in predicting non-compliance in air passengers. They demonstrated these methods on a case study of two months worth of data from Sydney Kingsford-Smith International Airport and found that the methods performed similarly to each other and better than a random sample. For this case study, the staff at the airport carried out a census operation and so they were able to obtain full information on inspection effort by cohort.

The issue of fairness in machine learning models is well-recognised across many industries from healthcare (Gianfrancesco *et al.*, 2018; Rajkomar *et al.*, 2018) and policing (Joh, 2017; Rich, 2016) to chatbots (Garcia, 2016) and resume screening (Derous & Ryan, 2019). There is considerable attention being given to the topic of finding ways to combat such problems (Holstein *et al.*, 2019; Williamson & Menon, 2019; Friedler *et al.*, 2019). For a more in-depth analysis into the problem of fairness in machine learning, see Chouldechova & Roth (2018) and Barocas *et al.* (2017).

The current project builds on the previous work in the following important way. Whereas previously, the effort was to try to estimate the number of passengers of each cohort within each intervention channel, the current project tries to fit statistical models that predict the probability of screening for individual cohorts. This innovation invites the use of potentially more accurate statistical tools in place of the previous candidates, while also easing automation.

[2] Anonymous (2016) *Air traveller profiles*, Confidential Internal Document, Department of Agriculture, Water and the Environment.

3. Datasets

The datasets available to us were:

- cohort volumes (Australian Border Force, Section 3.1),
- flight schedules (Airport Coordination Australia, Section 3.2),
- channel volumes (Mail And Passenger System, Section 3.3),
- interception data (Mail And Passenger System, Section 3.4), and
- endpoint survey (Mail And Passenger System, Section 3.5).

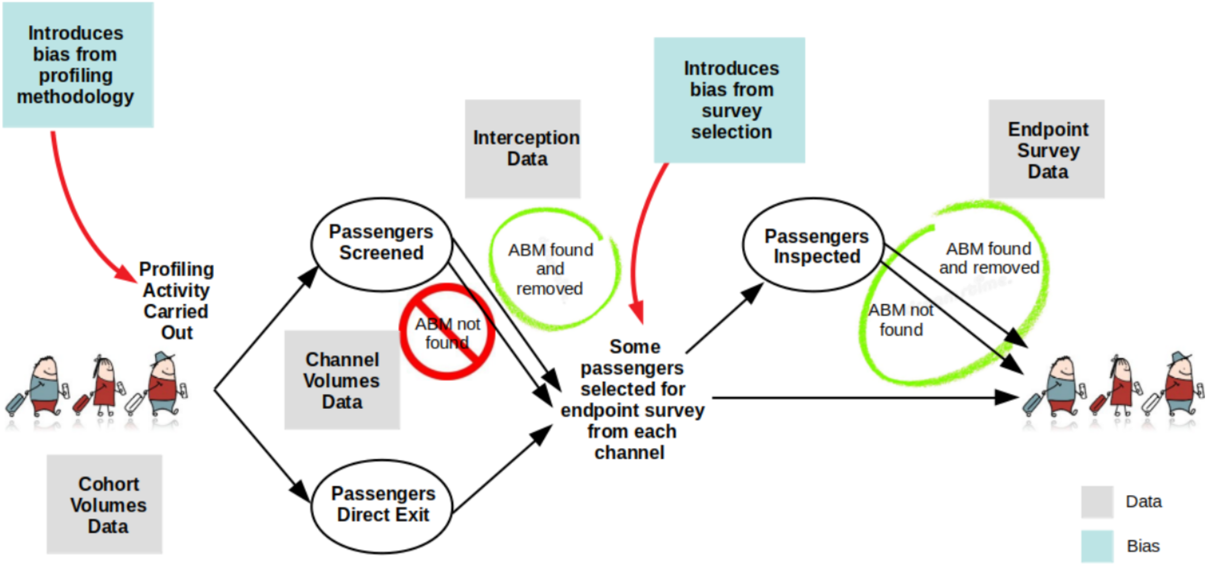


Figure 3.1.: Flow diagram that illustrates where in the passenger process the relevant datasets are captured.

Figure 3.1 shows where each of these datasets are captured with respect to a passenger's journey through the airport. *Of particular note is the lack of data for passengers that were screened but for which no non-compliance was found.* The endpoint survey has all information about passengers that were searched but that does not necessarily mean that the same ABM would have been found if they had gone through the screening channels. Indeed, many of them had already been through the screening channels and non-compliance was missed.

Additionally, the biases introduced by the profiling activity and the endpoint survey selection were also key elements that must be considered. Profiling introduces bias by design since we aimed to allocate higher screening resources to passenger cohorts with the highest risk score. Therefore, we could not assume that any findings from the screening channels were representative of passengers who went through the exit channel and vice versa. The endpoint survey also has bias, although not by design in this case, such as:

- biosecurity agents sampling passengers differently,
- some flights being more heavily sampled than others, depending on the number of passengers coming through the airport at that time, and
- staff allocation leading to some channels being more heavily sampled than others.

3.1. Cohort Volumes

We were provided cohort volumes for passenger movements from Jan 2015 to Nov 2019. There are 99,830,671 passenger movements in this dataset across 10 airports receiving international passengers (hereafter, "international airports"). We have passengers from 226 unique citizenships; most of these citizenships are countries but we also have codes for other passenger types such as stateless individuals and refugees. The dataset is highly imbalanced; the least represented nationalities that only appear once or twice in the dataset are: (Former) German Democratic Republic, France (Metropolitan), French Polynesia, (Former) Zaire, American Samoa, Montserrat, Norfolk Island, Puerto Rico and Sao Tome and Principe. Conversely, other citizenship countries make up a large proportion of the data, as summarised in Table 3.1.

Table 3.1.: Countries with the most rows in the cohort volumes data. Australia is by far the largest citizenship country, as we might expect, with 45% of the passenger movements being Australian citizens.

Citizenship	Movements	Proportion
Australia	44,997,781	0.45
China	9,527,972	0.10
New Zealand	9,216,423	0.09
UK	4,817,854	0.05
USA	3,452,806	0.03

The largest airports in Australia receive the bulk of the passengers; Sydney and Melbourne make up 66% of the data with 40,008,229 and 26,271,597 passenger movements respectively compared with Sunshine Coast which has 2,606,723 passenger movements. We use passengers aged between 1 and 100 for 3,643 unique flights. A sample of the dataset can be seen in Table 3.2.

Table 3.2.: Sample from cohort volumes dataset. The first two columns describe the date the passenger arrived, column 3 is the port that the passenger arrived into, columns 4–7 are the cohort characteristics and column 8 is the number of passenger movements within that cohort for that month.

Yr	Mth	Port	Flt	Citizenship	Age	Gender	Total
2019	10	ADL	MH139	AUS	67	F	16
2016	11	MEL	AI308	IND	25	M	27
2016	6	ADL	EK440	GHA	30	F	1
2018	6	BNE	NF24	AUS	40	F	2
2019	5	MEL	SQ247	MYS	31	M	1

3.2. Flight Schedules

The flight schedule data gives the route details and arrival times for each flight number. This is used during the preprocessing of the data to identify when a flight number might have been changed to represent a different route and can also be used for resource allocation in practice. It could also be used to derive additional features for our models, such as origin of flight, although this may not capture previous connecting flights. This dataset covers Jan 2014 to Oct 2019 for 8 international airports with 174,993 unique flights. The dataset covers 1,157 unique flight numbers and, like the cohort volume data, has 63% of the data coming from Sydney and Melbourne which have 67,556 and 42,120 flights respectively. A sample of the dataset can be found in Table 3.3.

Table 3.3.: Sample from flight schedules dataset. Columns 1–2 show the month that the data is referencing, column 3 shows the port, columns 4–5 break down the flight number, columns 6–8 and 11 describe the route that the flight takes, column 9 shows the estimated time of arrival and column 10 is the number of seats on the flight.

Yr	Mth	Port	Carr	Flt	Citizenship	Via	Dest	ETA	Seatsin	Rte
2016	10	MEL	QF	30	HKG	NA	NA	800	297	HKG-MEL
2016	1	SYD	QF	8	DFW	NA	NA	605	484	DFW-SYD
2014	2	MEL	VA	4152	DPS	NA	DPS	710	176	DPS-MEL
2017	5	PER	TZ	8	SIN	NA	SIN	1725	375	SIN-PER
2018	3	CNS	PX	90	POM	NA	POM	1055	100	POM-CNS

3.3. Channel Volumes

We have counts of passengers that went through each screening channel from Jan 2010 to Dec 2019. This dataset covers 21 airports with 173,286,183 passengers. 13 of these airports are small with relatively few passengers with the remaining 8 ports all having more than 1.7 million passengers. Overall, approximately 23.3% of passengers went through one of the screening channels. A sample of the dataset can be seen in Table 3.4. Table 3.5 provides a summary of passenger volumes for each channel for large airports. Note that there is a difference between screening and inspection — screening involves the passing of personal effects through x-ray, inspection involves the manual examination of effects. Relevant percentages can be found in Table 3.7.

Table 3.4.: Sample from channel volumes dataset. Columns 1–2 are the month that the data is describing, column 3 is the port, column 4 is the total number of passenger movements and column 5 is the channel.

Month	Year	Airport	Volume	Channel
10	2014	SYD	29380	Exit
4	2013	CNS	874	Detector Dogs
8	2014	CNS	93	Manual
9	2011	PER	3069	Manual
3	2013	ADL	6	Other

Table 3.5.: Summary of passenger volumes by channel for large airports from 2016–2018. Column 1 is the port, column 2 is the total number of passenger movements, columns 3–5 break this down by channel and column 6 shows the non-compliance that was found.

Airport	Total Passengers	Unscreened	Screened	Inspected	Non-Compliant
SYD	24,741,827	21,615,534	2,835,357	290,936	55,134
MEL	15,644,214	14,059,311	1,468,987	115,916	28,951
BNE	8,615,877	7,156,007	1,381,933	77,937	18,312
PER	6,488,158	5,121,672	1,234,669	131,817	23,344
OOL	1,605,999	1,310,539	276,736	18,724	3,971
ADL	1,441,930	1,138,535	249,601	53,794	8,166

3.4. Interception Data

The interception data shows full details for when a passenger was found to be non-compliant. This dataset covers Jan 2010 to Dec 2019 for 18 airports with 688,346 interceptions. There are many columns in this dataset but the key parts that we are interested in for the purposes of this project are about the details of the passenger, as shown in Table 3.6. The ABM is categorised either *risk* or *high-risk*; we look at only the passengers who are found to be carrying high-risk material. Additionally, the ABM is categorised either *declared*, *declared prompted* or *undeclared* and we look at only the passengers who are carrying undeclared or declared prompted material. Over a third of the interceptions come from Sydney (266,888 rows of data) but this number needs to be contextualized by considering how many passengers were screened at each port.

Table 3.6.: Sample from interceptions dataset. Column 1 is the channel that the passenger went through, columns 2–5 are their cohort characteristics, columns 6–7 describe the date and column 8 is the port.

Channel	Citizenship	Flightnumber	Gender	Age	Year	Month	Port
Manual	KOR	KE121	Female	32	2011	3	SYD
Manual	AUS	QF82	Female	36	2016	6	SYD
X-Ray	IND	D7232	Male	56	2017	7	PER
X-Ray	AUS	DJ4198	Female	52	2010	6	BNE

Table 3.7 shows the volume of passengers along with the volume that were screened and the volume of non-compliant passengers that were found for large airports in 2016–2018 (aggregated from the interception data and channel volumes dataset). We see that there is significant variation in the screening rate and the non-compliance rate between ports. Interestingly, there appears to be a negative correlation between airport size and screening rate — the smaller ports tend to screen a higher proportion of their passengers, but this does not necessarily equate to a higher NCR; for example, Gold Coast screens 18.4% of their passengers (1.2% are inspected and 17.2% are screened only). While the inspected non-compliance rate is rather high (15–25%), these inspections are likely only conducted if something has been discovered during screening, and drops to around 1–4% when the whole screening channel is considered.

Table 3.7.: Summary of passenger volumes, screening and non-compliance for large airports from 2016–2018. Column 1 is the port, column 2 is the number of passengers, column 3 is the % of passengers that went through the direct exit channel, column 4 is the % of passengers that were screened but not inspected and column 5 is the % of total passengers that were screened and inspected. Column 6 is the % of inspected passengers that were found to be non-compliant (in the general population, non-compliance is only found through inspection).

Airport	Total Passengers	% Unscrn	% Scrn	% Insp	Insp NCR
SYD	24,741,827	87.4%	11.5%	1.2%	19.0%
MEL	15,644,214	89.9%	9.4%	0.7%	25.0%
BNE	8,615,877	83.1%	16.0%	0.9%	23.5%
PER	6,488,158	78.9%	19.0%	2.0%	17.7%
OOL	1,605,999	81.6%	17.2%	1.2%	21.2%
ADL	1,441,930	79.0%	17.3%	3.7%	15.2%

3.5. Endpoint Survey

The endpoint survey data covers Jan 2010 to Dec 2019 and is a near-random sample of passengers who have already been through one of the channels. These people are selected by humans using a paper-based algorithm and so the sample will not be truly random, and also the proportion of passengers that are selected from each channel may not represent the true proportion of passengers that went through each channel. This introduces two forms of bias that need to be considered, namely cohort bias, where the selection of passengers may not be representative of all passengers entering Australia, and screening channel bias, where endpoint surveys are conducted more frequently from some screening channels. The significance of the first issue, cohort bias, will impact our confidence in non-compliance rates depending on how many samples have been collected from a particular cohort, while the screening channel bias can be corrected through re-sampling the data or conducting raking. The degree of cohort bias in the endpoint survey dataset can be determined, and Figure 3.2 compares the proportion of the endpoint survey that is made up of a given citizenship country to the proportion of the general population that is made up of that same country. There is clear correlation between the endpoint survey and the general population which is evidence that there is limited cohort bias in the endpoint survey.

The selected passengers are inspected for ABM (the second time for the passengers who went through one of the screening channels) by opening and manually inspecting all bags that have not already been opened.[1] It is important that the passengers who are coming from the x-ray, DDU or manual inspection channels have already been screened and may have been inspected. There is no way at this stage to determine whether or not the passenger has already had ABM confiscated, and this will lead to false negatives. However,the screening channel that the passenger passed through is known, including whether it was just screening or screening and manual inspection. Table 3.8 shows that 1-3% of passengers in the endpoint survey were screened and manually inspected, and 8-20% of passengers were screened only, depending on the airport. This means that the large proportion of the dataset that was not screened will not suffer from the issue of false negatives.

However, the endpoint survey does use a different screening approach than either the DDU or x-ray (one that is much more thorough), so the non-compliance rate obtained from the survey may not be representative of what would have been found had those passengers been sent through the screening channels.

This dataset covers 597,885 passengers across 13 airports. As with the interceptions dataset, there are many columns that are not relevant to our project at this stage and we will be focusing on the passenger details. A sample of the dataset can be found in Table 3.9.

[1] This policy may no longer be current.

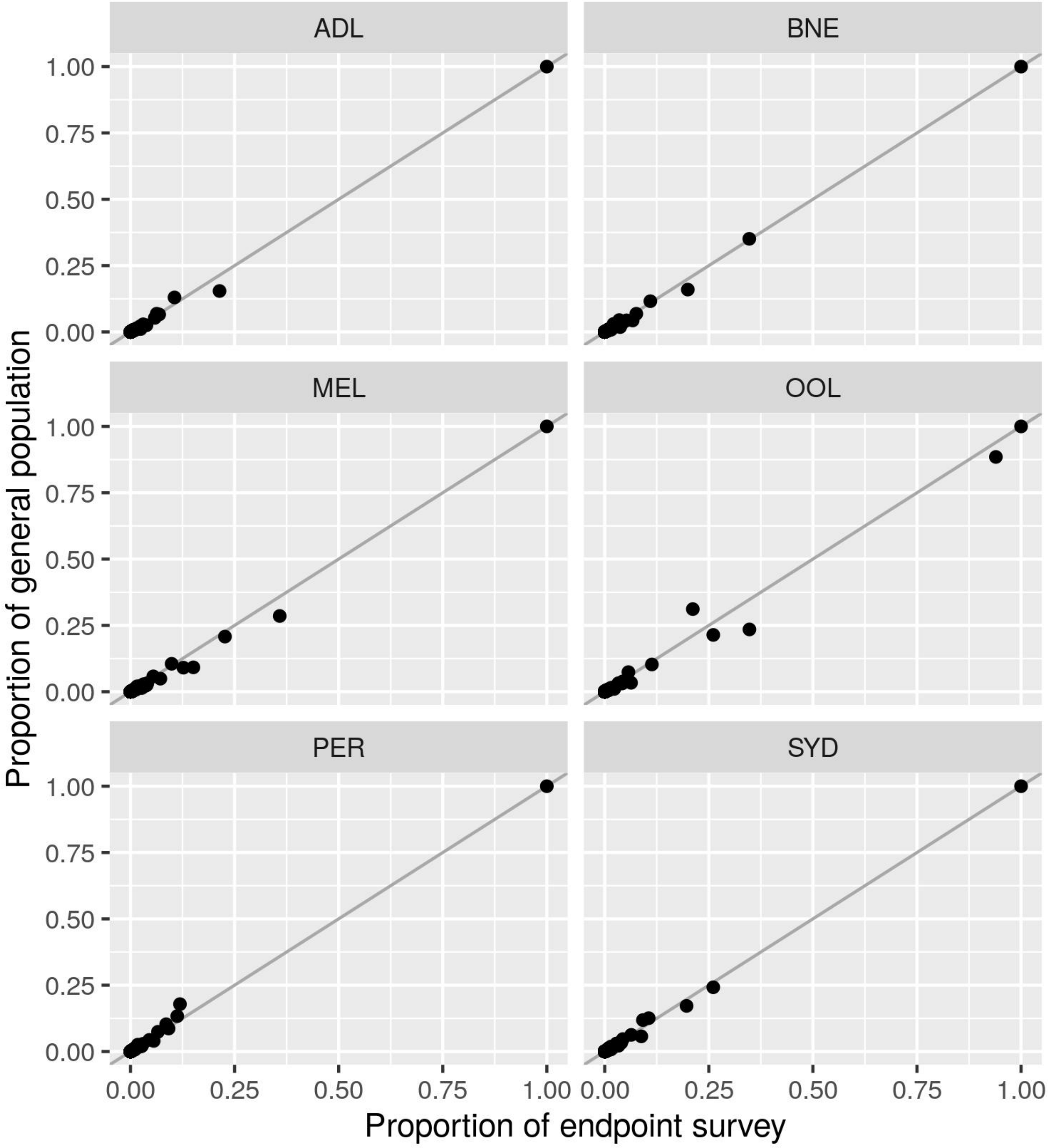


Figure 3.2.: Cohort bias scatter plot. The horizontal axis is the proportion of the endpoint survey that is made up of a given passenger cohort and the vertical axis is the proportion of the general population that is made up of that same cohort. The cohorts are grouped by citizenship and the data has been normalised for visibility (therefore Australian citizens show as making up 100% of both populations when in reality we know it is much less than this). The data are stratified by port.

Table 3.8.: Endpoint survey volumes for large airports from 2016–2018. Column 1 is the port and column 2 is the total number of endpoint surveys carried out at that port during the time period. Columns 3–5 are the percentage of these surveys that came from passengers that were unscreened, screened only or screened and inspected respectively. Columns 6–8 show the non-compliance that was found in the endpoint survey for each of the screening levels.

Airport	Survey Volume	% after Unscrn	% after Scrn Only	% after Insp	Unscrn NCR	Scrn NCR	Insp NCR
SYD	61,515	86.3%	12.5%	1.3%	2.6%	1.6%	5.5%
MEL	31,849	90.5%	8.3%	1.2%	2.7%	4.6%	7.6%
BNE	23,449	80.3%	16.7%	3.0%	1.7%	2.5%	3.8%
PER	21,082	77.2%	20.1%	2.8%	1.6%	2.0%	2.1%
OOL	8,847	78.4%	20.7%	0.9%	1.4%	1.8%	4.9%
ADL	8,231	79.1%	17.4%	3.5%	5.7%	5.6%	4.8%

Table 3.8 shows us the screening volumes and non-compliance rates in the endpoint survey. If we compare the screening and inspection rates to those shown in Table 3.7 then we see the screening channel bias in the endpoint survey. The *Inspected NCR* and *Screened NCR* columns can be described as the leakage for those channels since all of these passengers have either been screened or inspected. Note that the non-compliance rate behind the inspected passengers is unintuitively high relative to the other pathways. At face value this suggests that inspection is ineffective, but it is important to keep in mind that the inspected pathway has a much higher intrinsic contamination rate due to screening and profiling.

Table 3.9.: Sample from endpoint survey dataset. Column 1 shows the channel that the passenger came from with the post-fix O indicating that the passenger was screened only. Columns 2–5 are the cohort characteristics, columns 6–7 show the month that the survey took place and column 8 the port. Finally, column 9 is a binary variable indicating whether or not non-compliance was found.

Channel	Citizenship	Flightnumber	Gender	Age	Year	Month	Port	NC
X-Ray	JPN	PR218	Male	23	2017	3	CNS	0
Exit	NZL	DJ187	Female	21	2011	2	BNE	0
X-Ray	AUS	JQ120	Male	29	2011	5	SYD	0
Exit	USA	VA176	Female	18	2017	7	BNE	0
Manual	GBR	TG473	Male	43	2014	1	BNE	0

4. Choosing a Model for Cohort Risk

We considered a variety of techniques to calculate cohort risk. The primary goal overall was to choose optimal cohorts for screening so that we either (i) minimise the number of passengers carrying ABM that go unscreened or (ii) maximise the amount of ABM that is recovered. These goals can be achieved by selecting the passenger cohorts with the highest approach rate and hit rate respectively, as defined in Section 4.1.2 where we assess which objective function is preferable. If we just look at the ABM that was found without accounting for misses, we will find it leads to the self-fulfilling prophecy effect and an unfair solution; so our secondary goal was to avoid such bias in our predictions. In addition to methods that aid the building of profiles, we considered various metrics that may be used to rank the risk of these cohorts and discussed their merits.

When choosing the best model for this problem, we considered a number of subquestions:

1. How can we measure performance in data with missing information?
2. Is the existing solution structure appropriate? Can we use the endpoint survey data to avoid needing to predict screening rates within cohorts altogether?
3. Are there alternative tools that can be used in place of the existing tools that would see improved predictive power?
4. What is the best measure of non-compliance? For example, do we get different performance if we use the approach rate compared with the hit rate? And, which is preferable operationally?

4.1. Measuring Model Performance

The use of a training dataset and a testing dataset is important when building statistical models to be used for prediction. This is where the overall data are split either by a specific rule, such as by date, or by random sampling. The data that are allocated to the training set is used for training the model and tuning the model parameters and then the data in the testing set are used to assess model performance. It is imperative that the test data are not used in any part of the creation of the model so that we can show how generalisable the model is to new data. In our experiments, we used 3 years of data for our training set (July 2015 – June 2018) and 1 year of data for our testing set (July 2018 – June 2019). This was appropriate for our problem because our aim was to produce a model that generalises well into the future, and the one-year time step helped avoid effects of seasonality.

In supervised learning problems, all the data is labelled (i.e. the response variable is explicitly defined in both the training data and the testing data) and so model performance is easy to measure simply by looking at the prediction errors (i.e. the difference between the predicted value and the true value), generally in some sort of aggregated way such as the root mean squared error. This case, however, is a semi-supervised learning problem since we don't have the true population non-compliance in either the training data or the testing data. In such cases, we can use a heuristic approach whereby we use the labelled data to create proxy labels for the unlabelled data (Triguero *et al.*, 2015).

The techniques outlined in this section should be followed if the Department wishes to consider alternative models or alternative variables to use for profiling in the future. For example, the Department might get access to additional passenger information such as visa subclass or occupation. By following the techniques in this section, the Department can easily assess whether such additional profiling variables would include the predictive performance of the model. It is important in such a scenario to use the same training data and testing data for the models we are comparing.

4.1.1. Incomplete Test Data

This, however, does not solve the problem that the test data is also mostly unlabelled. To get an estimate of the non-compliance in cohorts for testing purposes, we used each of two different proxy non-compliance rates, as shown in Equation 4.1 and Equation 4.2. By using two measures of model performance throughout our experiments, we ensured that the method performs well across all channels. This also prioritised the non-compliance that can actually be found through screening.

$$\hat{\text{NCR}}_l = \frac{x_l}{v_l} \tag{4.1}$$

and

$$\hat{\text{NCR}}_l = \frac{x_{sl} + x_{el}}{n_{sl} + n_{el}} \tag{4.2}$$

Here, we define for cohort l,

v_l as the number of passengers;

x_l passengers are found to be non-compliant in screening;

n_{sl} screened passengers are reinspected in the endpoint survey; and

x_{sl} of them still have ABM;

n_{el} exiting passengers are reinspected in the endpoint survey; and

x_{el} of them still have ABM.

In other words, we used the passengers that we *found* to be non-compliant among the screened passengers (Equation 4.1) and the passengers for which we know we *missed* non-compliance using the previous profiling method (Equation 4.2) as our proxies for non-compliance. It is important that we considered both because Equation 4.1 does not consider passengers in the exit channel whilst Equation 4.2 includes non-compliance that may not necessarily be found in practice when screening passengers.

In practice, Equation 4.1 is where we use the interception data and the volume data in the test set to represent the non-compliance that the model aims find. Likewise, Equation 4.2 is where we use the non-compliance that was found in the endpoint survey to represent the non-compliance that the model aims to find. For both of these, we can create a plot similar to the traditional ROC to visualise model performance where we see the non-compliance that would be found for all screening cutoffs. These measures are used for the remainder of the report to assess model performance.

4.1.2. Non-Compliance Metrics

In order to measure model performance, we must first define what we consider to be non-compliant and therefore what we are aiming to find. The Department currently calculates three metrics that are used to report aspects of non-compliance. These are:[1]

- **Non-Compliance Rate (NCR)**: the proportion of screened passengers that were found to be non-compliant
- **Approach Rate (AR)**: the proportion of total travellers that are non-compliant
- **Hit Rate (HR)**: the proportion of passengers that we would find to be non-compliant if we screened them

A measure called Before-Intervention Compliance (BIC) is sometimes used in the place of Approach Rate - they are directly related since BIC $= 1 - \text{AR}$. The Department also uses a number of other measures such as Post-Intervention Compliance (PIC) and Non-Compliance Effectiveness (NCE) as Key Performance Indicators. Although these are not suitable for this stage of the profiling, they are excellent at reporting the real performance of the screening after it has taken place in practice. Mathematically, the NCR is as previously defined in Equation 2.1 and the AR and HR are defined as Equations 4.3 and 4.4.

$$\text{AR} = \frac{\text{leakage in screen} + \text{leakage in exit} + \text{non-compliance found}}{\text{total number of passengers}} \tag{4.3}$$

$$\text{HR} = \frac{\text{non-compliance in exit} \times \text{screening effectiveness} + \text{non-compliance found}}{\text{total number of passengers}} \tag{4.4}$$

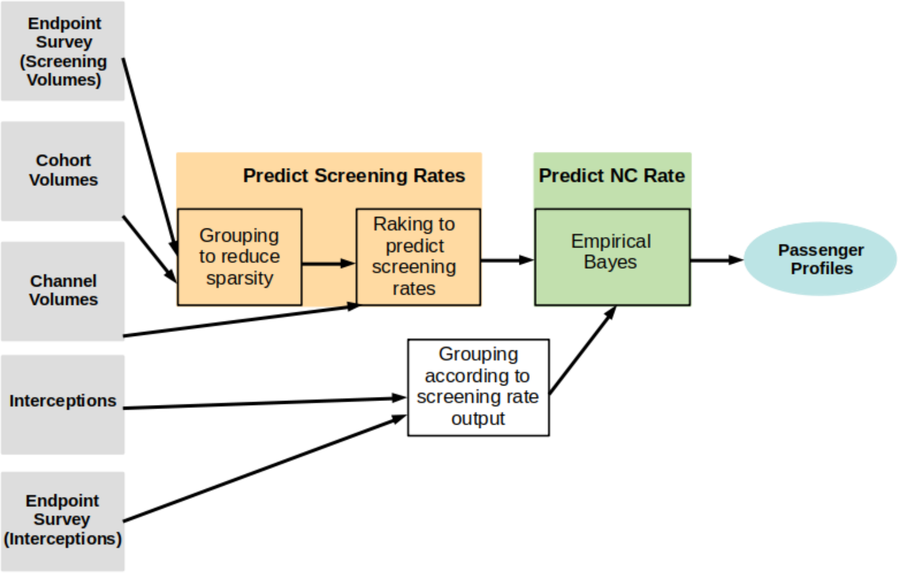
The above equations are defined algebraically in Appendix A.

Both AR and HR rely heavily upon the endpoint survey to calculate leakage and screening effectiveness, which means that they are also affected by cohort bias. However, unless we know the true screening rate, this will also impact the NCR calculation as the endpoint survey is used to determine the proportion of passengers screened in a cohort. If we have few samples for a particular cohort in the endpoint survey, we will have a lower confidence in our estimated proportion of passengers screened. By definition, the passengers screened will be biased towards particular cohorts as well, though much more data is collected overall compared to the endpoint survey.

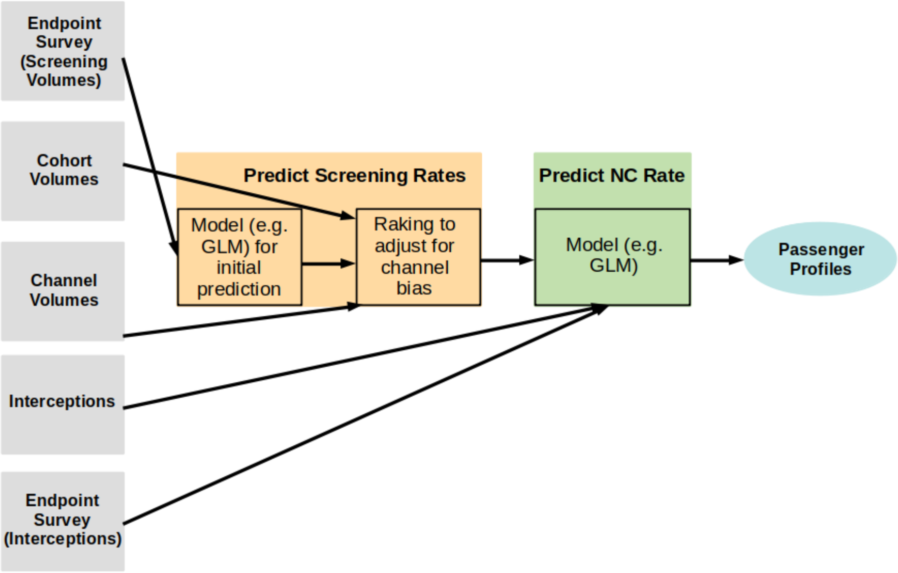
4.2. Solution Structure

In this section, we outline the principles of the models that we are considering in three forms: graphically, algorithmically and mathematically. Figure 4.1 provides an overview.

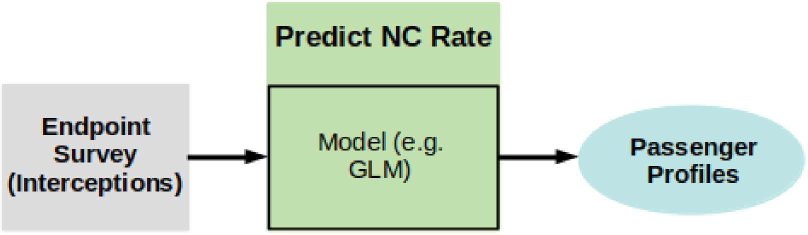
[1] The equations and definitions in this section are inferred from current Department profiling code as well as descriptions in Robinson et al (2013) *Adoption of meaningful performance indicators for quarantine inspection performance*, ACERA Final Report for 1101D


Endpoint Survey (Screening Volumes)
Cohort Volumes
Channel Volumes
Interceptions
Endpoint Survey (Interceptions)
Predict Screening Rates
Grouping to reduce sparsity
Raking to predict screening rates
Predict NC Rate
Empirical Bayes
Passenger Profiles
Grouping according to screening rate output

(a) Historical ACERA profiling method; comprising grouping and raking to augment the screening data, and group-based smoothing to estimate hit rates.


Endpoint Survey (Screening Volumes)
Cohort Volumes
Channel Volumes
Interceptions
Endpoint Survey (Interceptions)
Predict Screening Rates
Model (e.g. GLM) for initial prediction
Raking to adjust for channel bias
Predict NC Rate
Model (e.g. GLM)
Passenger Profiles

(b) First proposed method; comprising modeling and raking to augment the screening data, and model-based smoothing to estimate hit rates.



(c) Second proposed method; comprising model-based smoothing of just endpoint data to estimate hit rates.

Figure 4.1.: Graphical comparison of three algorithms: (a) the current method (group/ rake/ group smooth), (b) the first proposition (smooth/ rake/ model smooth), and (c) the second proposition (model smooth).

4.2.1. Graphically

The method that is currently used is reported in Figure 4.1a. In particular, we group cohorts to reduce the sparsity in the data before using a tool called raking to predict the screening rates for each cohort and then apply a smoothing technique called empirical Bayes to predict the non-compliance. The grouping of the cohorts results in large profiles that are not as precise as they could be.

To improve the process, we introduced a statistical model to the existing structure in place of the grouping to act as a seed for raking. In addition to this we introduced a statistical model in place of the empirical Bayes. This proposed solution is outlined in Figure 4.1b and described in Algorithm 1.

4.2.2. Algorithmically

Algorithm 1 Algorithm for proposed profiling method (smooth/rake/model smooth).

```
1: Aggregate data within each cohort $l$ to compute $v_l$, $x_l$, $n_{sl}$, $x_{sl}$, $n_{el}$ and $x_{el}$ as defined in Section 4.1. Here, a cohort is defined by its unique combination of profiling characteristics: age, gender, citizenship country and flight number.
2: Scale the channel volumes so that the total number of passengers across all channels is equal to the total number of passengers in the cohort volumes data. We have that $v_s$ and $v_e$ are the scaled volumes of passengers going through the screening channels and the exit channel respectively.
3: Train a screening model with $\frac{n_{sl}}{n_{sl}+n_{el}}$ as the response variable and the profiling characteristics as the predictor variables. This gives us a prediction of the number of passengers that were screened within cohort $l$ which we define as $\hat{s}_l$. Then the predicted number of passengers within cohort $l$ that went through the exit channel is $\hat{e}_l = v_l - \hat{s}_l$.
4: Correct for underestimations of number of passengers screened: if $\hat{s}_l < x_l$, then set $\hat{s}_l = x_l$
5: Carry out raking with $\hat{s}_l$ and $\hat{e}_l$ as our initial predictions and $v_s$, $v_e$ and $v_l$ as the marginal totals. We then have transformed predictions of screened and unscreened passengers which we denote $\hat{s}_l''$ and $\hat{e}_l''$.
6: Train a non-compliance model with $\frac{x_l}{\hat{s}_l''}$ as the response variable and the profiling characteristics as the predictor variables.
7: Train leakage models for both the screening channels and the exit channel with $\frac{x_{sl}}{n_{sl}}$ and $\frac{x_{el}}{n_{el}}$ as the response variables and the profiling characteristics as the predictor variables once again.
8: Calculate the metric we wish to measure non-compliance by such as approach rate or hit rate. In our base case, we use the approach rate $\mathrm{AR}_l$.
9: Train a final model with $\mathrm{AR}_l$ as the response variable and the profiling characteristics are the predictor variables. Embedded in this model is the information from all the previous steps so this is the only model that we need to apply to the testing data.
10: Apply to either the actual cohorts that we expect to receive or to all unique combinations of characteristics so that we get risk scores for all possible future cohorts. Order cohorts by descending risk score and screen the top $k$%.
```

This method has the issue that any errors in the screening rate prediction will be carried through to the non-compliance rate prediction. We chose to also consider a method that simply uses the endpoint survey data to make a prediction, as shown in Figure 4.1c. While side stepping the issue of approximating screening rates, this method has other drawbacks. Namely, just using the endpoint survey means we are working with a much smaller dataset, and we are throwing away the information available in the interception dataset. The non-compliance in the endpoint survey may also not have been found, had the passenger been screened, because the endpoint survey involves a full inspection of passengers which is more thorough than the screening.

4.2.3. Mathematically

Mathematically speaking, all the models that we consider throughout this report use the same profiling characteristics and the same model form unless otherwise specified (e.g. when we consider interactions). The passenger characteristics that we use to fit our models are:

- citizenship country (this refers to the passenger's citizenship country and not the flight origin),
- gender,
- age, and
- flight (this refers to the flight number and route rather than a particular instance of a flight on a certain day).

We use these characteristics to produce a model of the form shown in Equation 4.5 where β_i are the various coefficients for each variable.

$$\text{NC} \sim \beta_1\text{Citz} + \beta_2\text{Gender} + \beta_3\text{Age} + \beta_4\text{Flight} \tag{4.5}$$

We considered alternative model structures to that shown in Equation 4.5 such as a B-spline (Unser *et al.*, 1993) on age which we found did not improve the overall performance of the model. Additionally, we considered including interactions between variables, the results of which are presented in Section 4.5.1.

4.2.4. Port-based Profiling

Profiling is currently carried out at a flight level which means flights are individually assessed and there is no comparison of passenger risk from one flight to another. This can cause passengers on low-risk flights to appear more risky overall and passengers on high-risk flights to appear less risky overall. This is less than ideal for highly targeted screening. The reason for working at a flight level was due to when the profiling work was initially rolled out it allowed for a phased approach. Since that is no longer an issue, we recommend moving to a method that carries out analysis at the port level, that is, profiles are constructed for passengers within each port.

4.2.5. Ungrouped Cohorts

Under the previous methodology, the cohorts would be grouped until there was sufficient endpoint survey data for each cohort, which would be used as the seed. As such, the cohorts would remain grouped in this way throughout the rest of the analysis and we would end up with rather large cohorts once the analysis was complete. This meant that there was opportunity for high risk cohorts to be combined with low risk cohorts and so they would both drag the other toward a central mean and both cohorts would not have the correct risk score. The new method avoids this by using passenger characteristics to inform predictions rather than the cohort as a whole. This means that for a given passenger, passengers with one or more same characteristic(s) can also be used for prediction rather than just the passengers in the cohort. This is extremely beneficial for a dataset with high sparsity such as is common in these cases.

4.3. Results

In this section, we present the effectiveness of the methods outlined in the previous two sections.

4.3.1. Choosing a Non-Compliance Metric

Firstly, we must decide what our *goal* should be: our definition of non-compliance. Figure 4.2 shows how the NCR compares to the HR and AR. The x-axis is the proportion of passengers screened, and the y-axis is the proportion of interceptions found or the true positive rate. Therefore, the method with the steepest curve towards the top left hand corner of the plot is the best. Since we can't know the true proportion of non-compliance that would be captured, we estimate it using the methods outlined in Section 4.1.1. We see that both the HR and AR outperform the NCR substantially but there is not a big difference between HR and AR.

As such, we recommend following Occam's Razor (Blumer *et al.*, 1987) and choosing the simplest model. This would be the approach rate since the hit rate has to estimate screening effectiveness which would likely vary by channel. Alternatively, more simply, looking at Equations 4.3 and 4.4 in Appendix A, we see that the hit rate equation is more complicated with many interaction terms.

4.3.2. Evaluating the performance of each potential solution

Figure 4.3 compares how each possible solution structure predicts onto new data for Sydney. We see that both methods saw a noticeably improved performance from the current method and that the first proposed method saw a slight improvement over the second proposed method that just used the endpoint survey data for training. This is particularly true when screening a smaller proportion of passengers, which is the most relevant section of the plot when considering actual screening proportions (which are 10–20%, see Table 3.7. Therefore, we concluded that (a) we can certainly improve model performance from the current method by applying different tools, and (b) the exercise of estimating screening rates in order to calculate non-compliance using the interception data is indeed worthwhile.

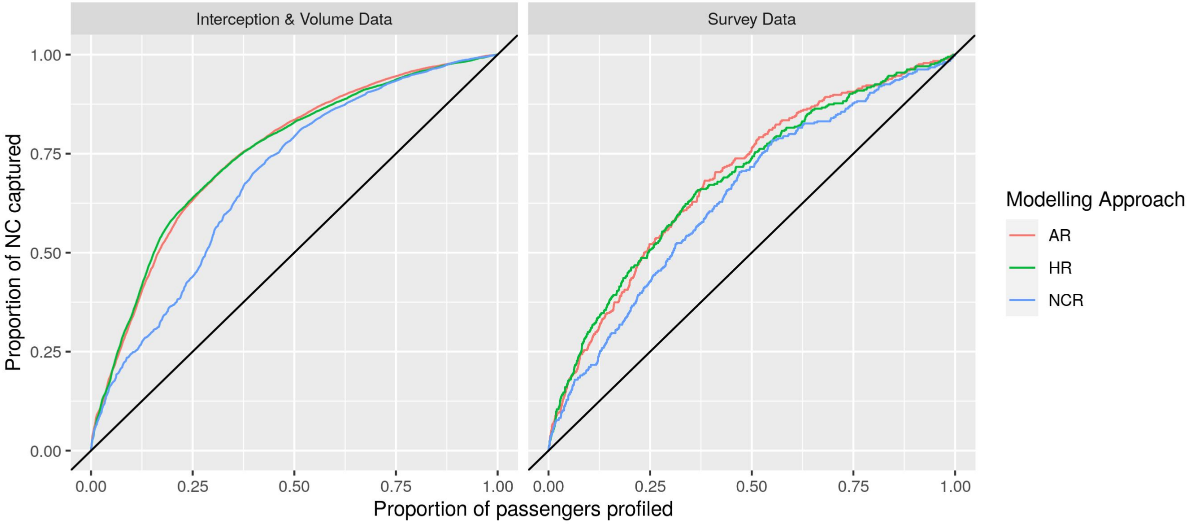


Figure 4.2.: The effectiveness of the various metrics (namely, approach rate, hit rate, and non-compliance rate) at describing non-compliance.

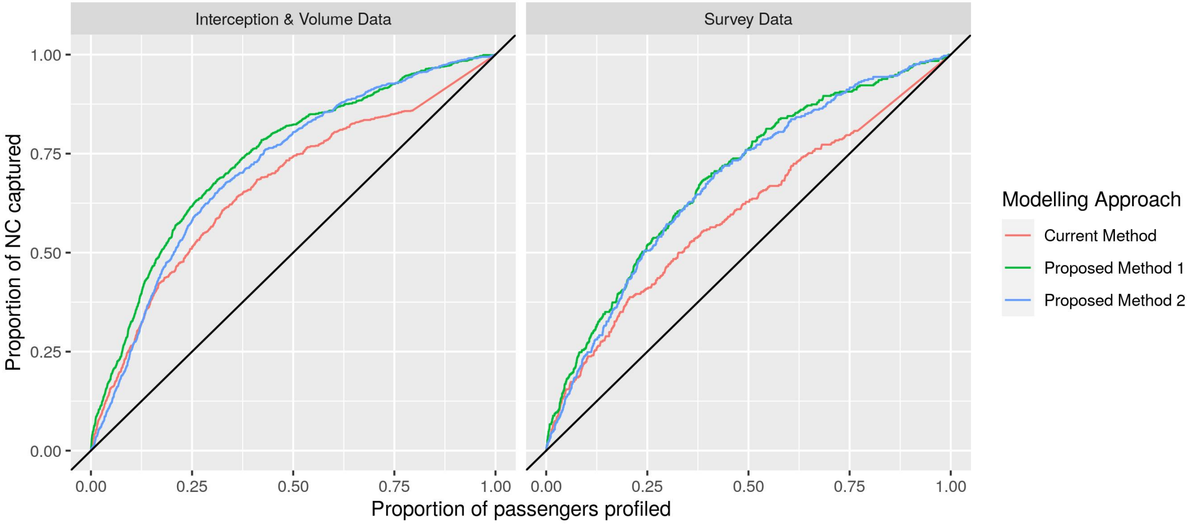


Figure 4.3.: Comparison of the performance of two proposed methods to the current method. The left panel uses interceptions as a measure of performance and the right panel uses the endpoint survey data to measure performance.

4.4. Tool Selection

In this section we looked at a number of machine learning methods that may be used as the statistical model in Algorithm 1. These methods still used the structure shown in Equation 4.5, it was just the method used to fit the data to that structure that differed. There are many supervised learning tools that could be applicable to this problem, we chose to test four techniques that are known to perform well on similar problems:

- Generalised Linear Mixed Effects Model or *GLMER* (Wood, 2017) is an extension of the Generalised Linear Model or *GLM* (Hastie, 1992) which is itself an extension of the Linear Model or *LM* (Woodward *et al.*, 1990). The GLM allows response variables to have arbitrary distributions which is particularly helpful for problems such as ours whereby the response variable is a probability and so is bounded between 0 and 1. The GLMER also has this quality but with the additional benefit of allowing us to group variables as random effects and measure trends within groups as well as in the full data.
- Random Forest (Pal, 2005) is a collection of decision trees, each built from a random sample of the overall data. It is known to be effective at solving supervised classification and regression problems.
- Gradient Boosting Machine or *GBM* (Friedman, 2001) is a machine learning method that uses a collection of weak learners (in this case, decision trees) in order to make some prediction. By combining the decision trees iteratively, we create a single strong learner.
- Naive Bayes (Murphy *et al.*, 2006) uses Bayes theorem to calculate the probability of the response variable taking a particular value conditional on the data. Naive Bayes is known to be effective at solving supervised classification problems.

In our experiments, we used the data analysis software R (R Core Team, 2020) with the corresponding statistical packages for each method (Greenwell *et al.*, 2020; Bates *et al.*, 2015; Meyer *et al.*, 2020; Wright & Ziegler, 2017). The results are shown in Figure 4.4. We see that GLMER performs the best out of these methods. Random Forest is the weakest of the methods; a possible explanation for this is the vastly different number of categories in each characteristic (e.g. 2 possible genders compared with 240 possible flight numbers) which Random Forest has been known to struggle with (Deng *et al.*, 2011).

All of these techniques followed Algorithm 1 in structure, it was only the choice of model at the five key points that changed. We chose to keep raking, sometimes called iterative proportional fitting (Deming & Stephan, 1940), as a part of the analysis because it does an excellent job at rescaling values to match marginal totals. Appendix B provides a brief explanation as to the mechanics of raking.

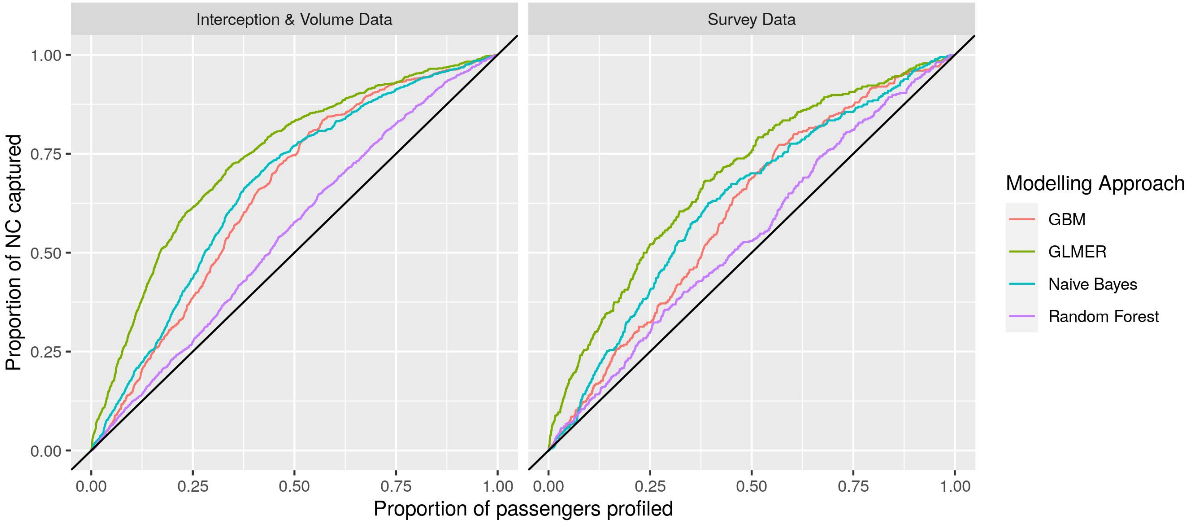


Figure 4.4.: Comparison of the predictive performance of various statistical and machine learning methods (see text for description). The left panel uses interceptions as a measure of performance and the right panel uses the endpoint survey data to measure performance.

4.5. Improving the Model

Once we had settled on the basics of the model, we began to consider how the model might be improved.

4.5.1. Interactions

The models in Chapter 4 all considered the characteristics as independent variables and did not consider interactions between them. We investigated whether this was indeed the best model.

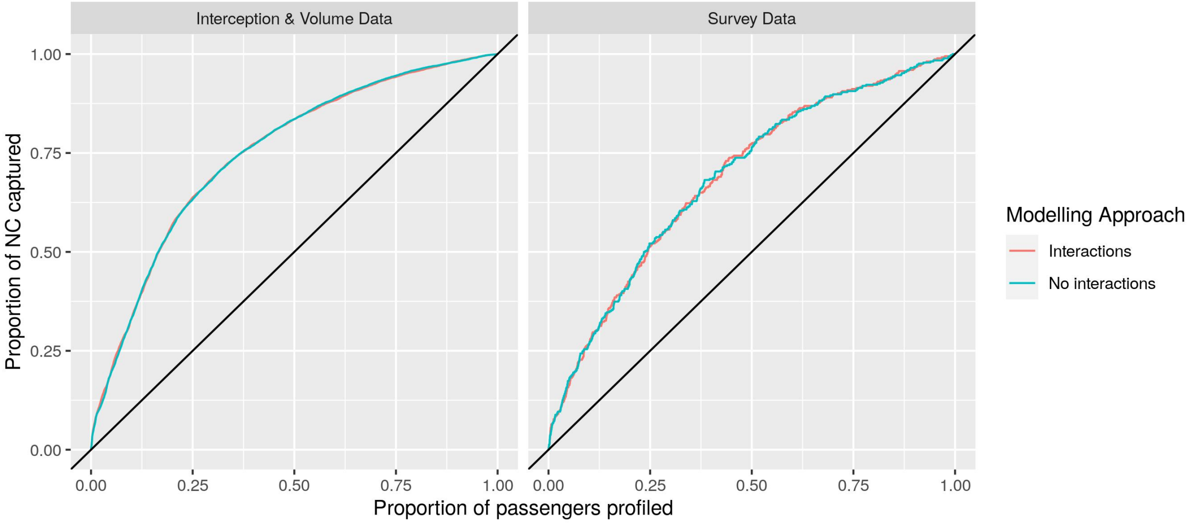


Figure 4.5.: ROCs showing model with interactions compared to model with no interactions. The lines are essentially indistinguishable.

Figure 4.5 shows the performance of both the original model and the model with interaction terms. We see no difference in the performance of the models and so, as per Occam's razor, we must conclude that the simpler model without the interaction terms is preferred.

4.5.2. National Profiles

We know that moving from profiling at a flight level to profiling at a port level is more effective so we considered taking it a step further and profiling at a national level. In particular, we trained the model using data from all airports and then tested it against the data for each individual airport.

Figure 4.6 shows the comparison between national profiles and port level profiles for Sydney. There is no noticeable improvement seen by moving to a national profiling method. This may be unsurprising for Sydney because it is a very large port and a large proportion of the data came from there so the national model might be biased towards a model that fits Sydney data well. In order to test this, we compared the national profiles to the other airport that we had complete data for: Melbourne, Brisbane, Perth, Coolangatta and Adelaide - these can be seen in Figures 4.7 to 4.11. Generally, the other airports behave similarly to Sydney, seeing no improvement to model performance by using national profiles.

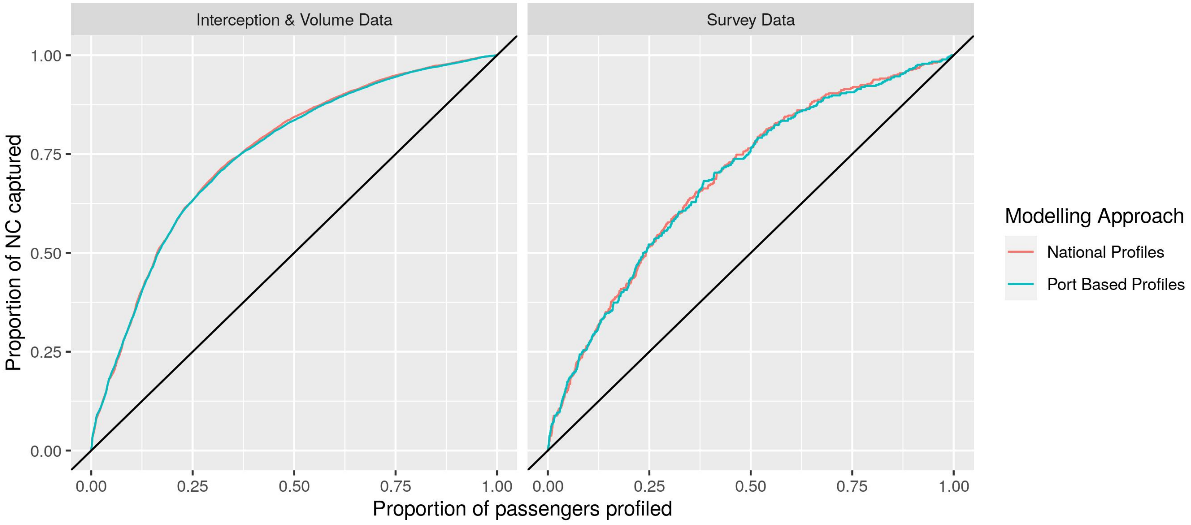


Figure 4.6.: Comparison of the predictive performance of national profiles against port-level profiles for Sydney using Sydney data.

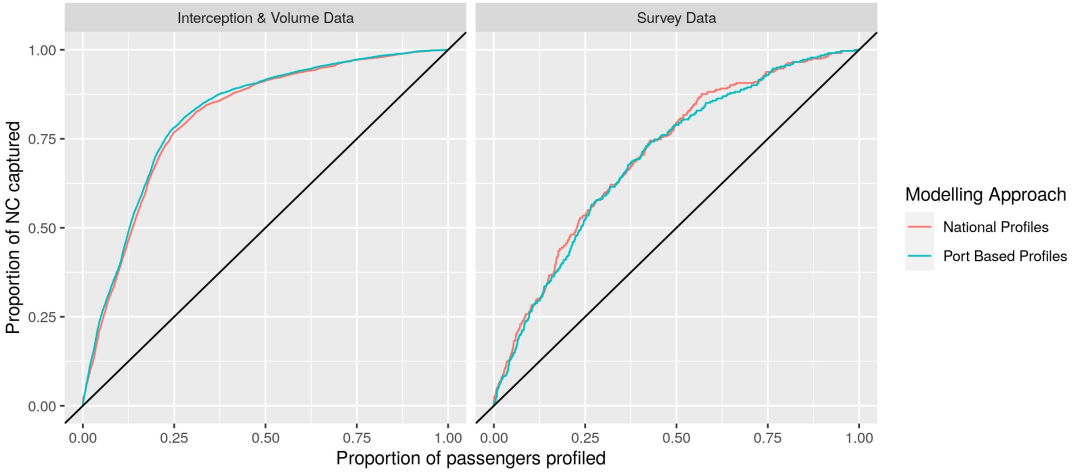


Figure 4.7.: Comparison of the predictive performance of national profiles against port-level profiles for Melbourne using Melbourne data.

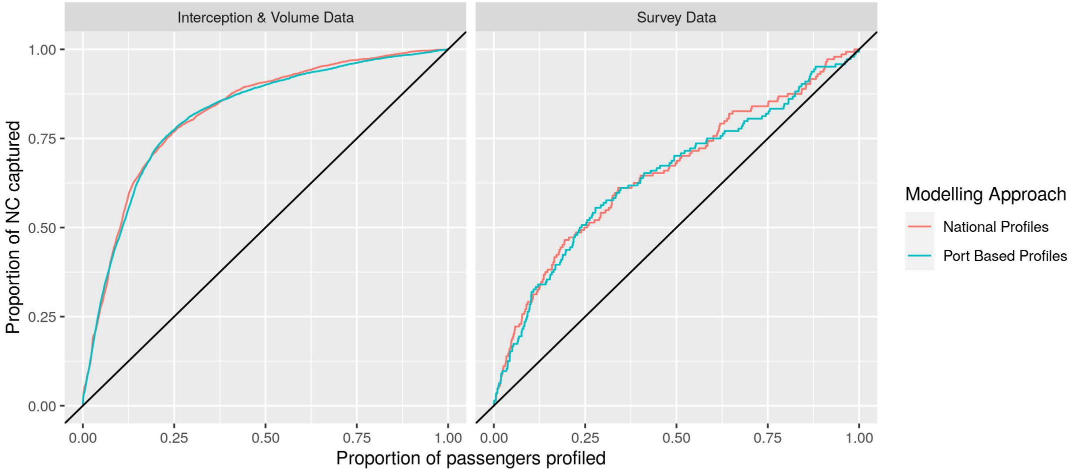


Figure 4.8.: Comparison of the predictive performance of national profiles against port-level profiles for Brisbane using Brisbane data.

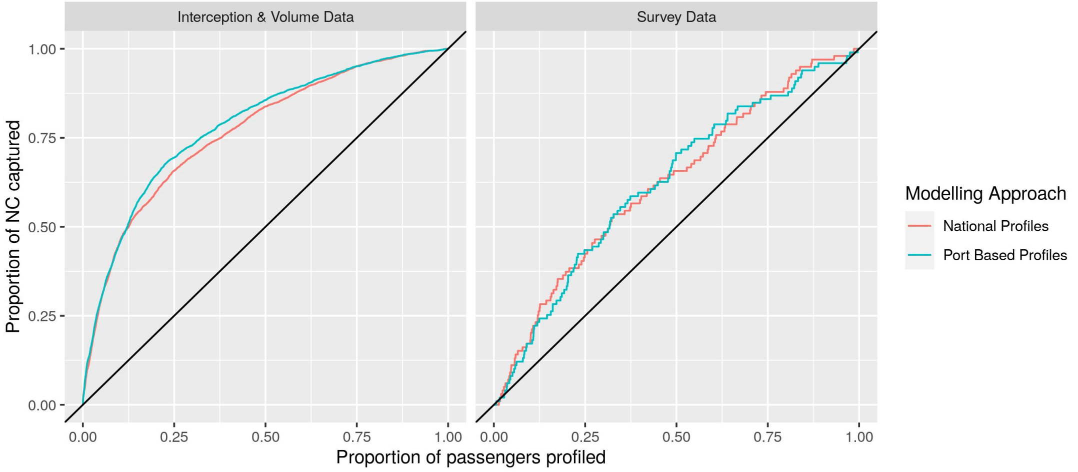


Figure 4.9.: Comparison of the predictive performance of national profiles against port-level profiles for Perth using Perth data.

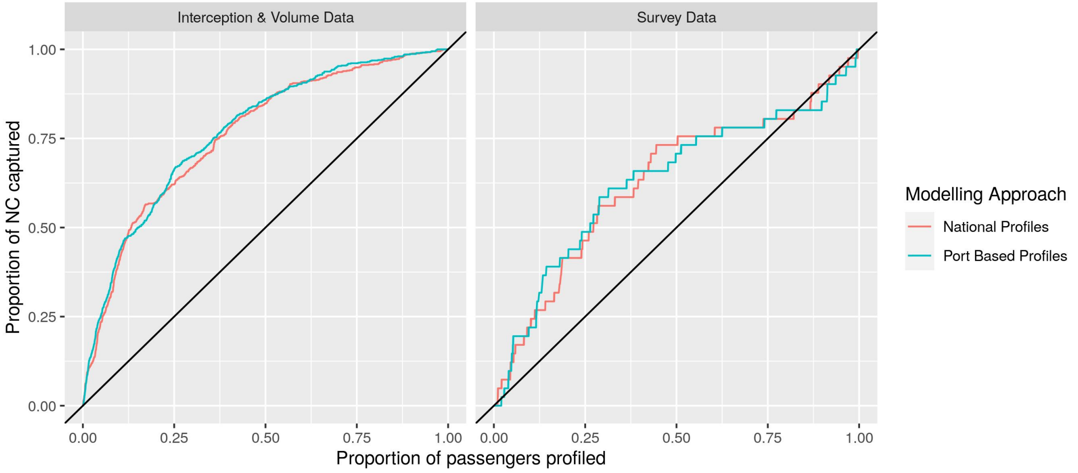


Figure 4.10.: Comparison of the predictive performance of national profiles against port-level profiles for Coolangatta using Coolangatta data.

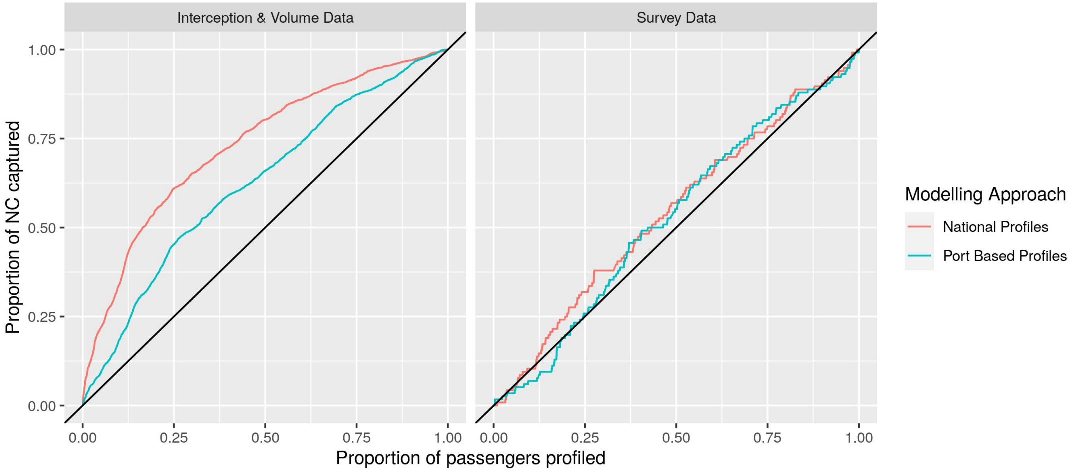


Figure 4.11.: Comparison of the predictive performance of national profiles against port-level profiles for Adelaide using Adelaide data.

The sole exception to this statement is Adelaide (Figure 4.11). Using interceptions as our measure of performance, national profiles appear to noticeably outperform the Adelaide model. When we use the endpoint survey as our measure of performance, not only do we see little to no difference between the two models but both appear to have poor predictive power, barely better than a random sample. This indicates that there may be something different in how the data is collected in Adelaide. It was essential that we investigate this before applying this profiling method to Adelaide passengers to ensure that the model is effective.

We investigated a number of hypotheses outlined in Table 4.1, however none brought us closer to understanding the reason for Adelaide's poor performance.

Table 4.1.: Investigation of candidate explanations for the comparatively poor performance of profiles for Adelaide. Four hypotheses (left column) are tested (middle column) with results reported in right-hand column.

Hypothesis	Test	Outcome
Something happened in the 2019 data in Adelaide that was different to 2015 – 2018.	Split the data differently so that there are 2 years in the training set and 2 years in the test set: July 2015 – June 2017 is training, July 2017 – June 2019 is test.	Performance is exactly the same as in 4.11.
Other temporal effect.	Split training and test set by random sampling across all 4 years.	Performance is the same as in Figure 4.11.
The model is overfitting to the training data.	Predict model back onto training data to measure effectiveness.	On interceptions plot (left plot in Figure 4.11) national profiles perform similarly but port based sees an improvement. Both do better when tested on the survey data but not up to the same standard as the other ports
The complexity of our model is doing something strange to the data.	Train a simple model like the one shown in Figure 4.1c.	Predictive performance was worse than in Figure 4.11.
The risk within the Adelaide data is more homogeneous between cohorts than in the other ports, perhaps due to fewer flights from high-risk countries.	Compare the top citizenship countries for Adelaide to those of the other ports	There were fewer high-risk citizenships arriving at Adelaide.

Our investigation indicated that part of the problem for Adelaide might be that the model is overfitting however this is not the whole explanation. We also saw that there appear to be fewer high-risk flights coming in to Adelaide, indicating that the risk might be more homogeneous between cohorts.

4.6. Unseen Cohorts

The new method can predict the risk for previously unseen cohorts by using the passengers' other characteristics, however the profiles will not be as accurate as if we were to retrain when data on the cohort become available. The best way to handle the unseen cohort is a policy decision, but we would advocate a risk posture somewhere between risk neutrality (namely, set unknown risk factors to the average) and risk aversion (increase efforts on unknown cohorts in case of heightened risk).

5. Cutoffs: How Many Cohorts to Inspect?

The work thus far looked exclusively at how to rank passengers according to their likelihood of being non-compliant. This section looks at how we can use that list of all cohorts to determine who to actually screen in practice.

Under the current methodology, profiles are generated at regular intervals in a static manner — the training data is used to create a list of cohorts that, in the event that a passenger from one of the cohorts arrives at the airport, an alert will be generated indicating that this passenger should be screened. These alerts are currently generated at a flight level so each flight uses the same cutoff method, resulting in a potential over-screening of low risk flights and under-screening of high risk flights. As discussed in Section 4.2.4, we recommend that the Department moves to a port level screening methodology which should amend those issues.

When profiles are generated, there is currently no mechanism to determine the implications this may have for current screening capacities. Anecdotally, more alerts are being generated than there is capacity to screen, and as such passengers that have an alert raised against them may be allowed to exit the airport without further intervention. As a result, we are effectively randomly selecting passengers from the list of alerts, and these are the passengers that are actually screened.

This is not ideal because the random sampling means that we are not necessarily prioritising the highest risk passengers. Additionally, it are further contributes to the issue that we don't know who was screened and therefore makes future analyses harder. We propose choosing a cutoff to match capacity as closely as possible. This will also make performance measures more accurate to real life since we will be assessing our performance based on passengers that were actually screened rather than passengers that we aimed to screen.

To inform this decision for a single port (using Sydney as an example), we can use a combination of Figures 5.1 and 5.2, which are interpreted as follows. Figure 5.1 shows us the percentage of passengers that we would have to screen in order to capture a given amount of non-compliance (for example, say we wanted to capture 62% of non-compliance then Figure 5.1 tells us that in order to achieve this interception rate, we would need to screen 25% of all passengers coming through the airport). Figure 5.2 shows the average effort required to screen passengers at different rates, for different times of the day on each day of the week, as the number of inspections to be done, with colour to help show patterns. Comparing the three heat maps, it is clear that the demand on the screening staff depends on the choice of screening rate cut off.

The heat maps can also be useful for management scheduling tasks that are not time-dependent. Many of the the peak fluctuations will be very familiar to managers who know the flight schedules well; however, peak total passenger arrival times are not necessarily at the same times as peak risky passenger arrival times. Indeed, comparing the heat maps in Figure 5.2 with the heat map of total passenger arrivals in Figure 5.3, we see a number of key differences.

Specifically looking at the heat map that uses a 10% screening rate since these are the highest risk passengers, we notice that there is a stripe of passenger arrivals in Figure 5.3 at 6pm which is not present in the Figure 5.2. Unsurprisingly, there are a lot of flights from New Zealand that arrive at this time and New Zealand also has

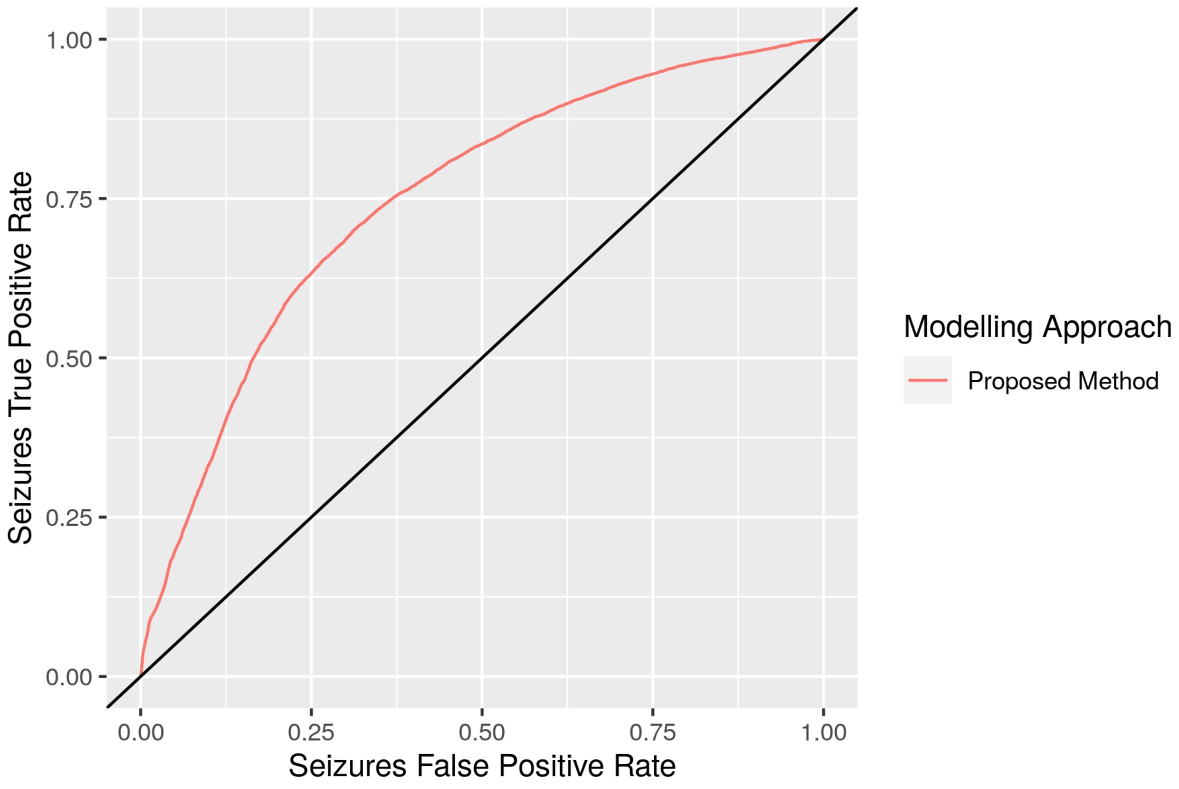
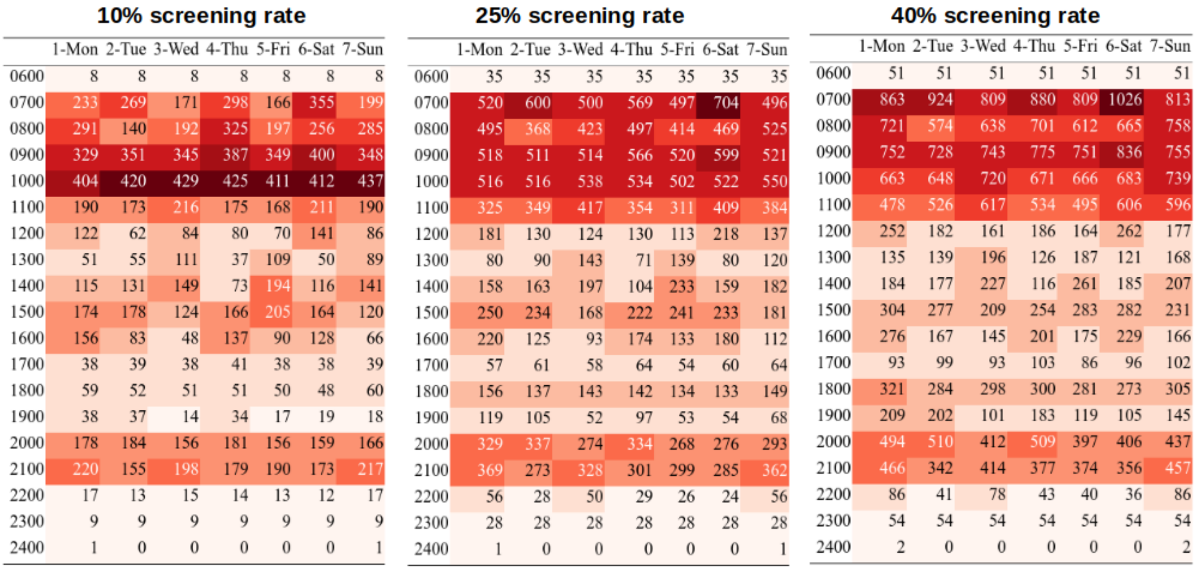


Figure 5.1.: Proportion of passengers to screen in order to capture a given proportion of non-compliance (ROC curve).



10% screening rate

	1-Mon	2-Tue	3-Wed	4-Thu	5-Fri	6-Sat	7-Sun
0600	8	8	8	8	8	8	8
0700	233	269	171	298	166	355	199
0800	291	140	192	325	197	256	285
0900	329	351	345	387	349	400	348
1000	404	420	429	425	411	412	437
1100	190	173	216	175	168	211	190
1200	122	62	84	80	70	141	86
1300	51	55	111	37	109	50	89
1400	115	131	149	73	194	116	141
1500	174	178	124	166	205	164	120
1600	156	83	48	137	90	128	66
1700	38	39	38	41	38	38	39
1800	59	52	51	51	50	48	60
1900	38	37	14	34	17	19	18
2000	178	184	156	181	156	159	166
2100	220	155	198	179	190	173	217
2200	17	13	15	14	13	12	17
2300	9	9	9	9	9	9	9
2400	1	0	0	0	0	0	1

25% screening rate

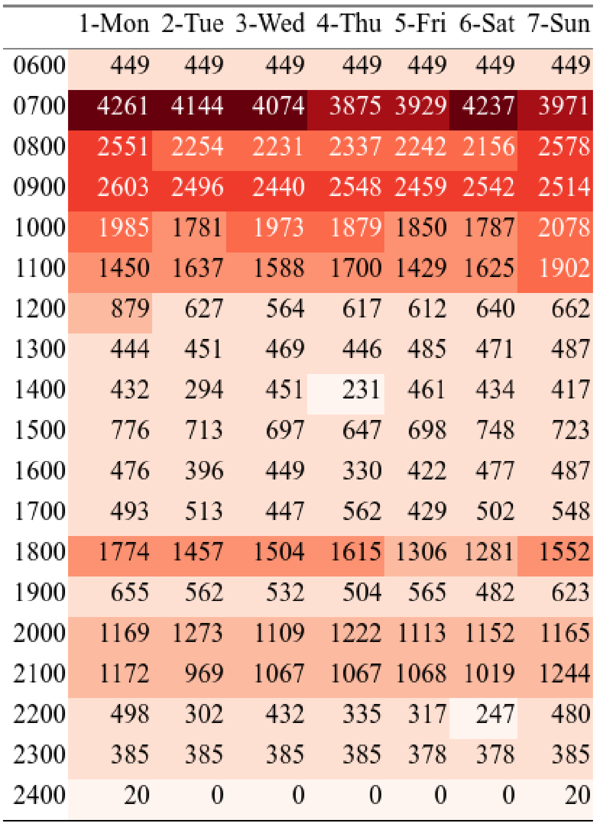
	1-Mon	2-Tue	3-Wed	4-Thu	5-Fri	6-Sat	7-Sun
0600	35	35	35	35	35	35	35
0700	520	600	500	569	497	704	496
0800	495	368	423	497	414	469	525
0900	518	511	514	566	520	599	521
1000	516	516	538	534	502	522	550
1100	325	349	417	354	311	409	384
1200	181	130	124	130	113	218	137
1300	80	90	143	71	139	80	120
1400	158	163	197	104	233	159	182
1500	250	234	168	222	241	233	181
1600	220	125	93	174	133	180	112
1700	57	61	58	64	54	60	64
1800	156	137	143	142	134	133	149
1900	119	105	52	97	53	54	68
2000	329	337	274	334	268	276	293
2100	369	273	328	301	299	285	362
2200	56	28	50	29	26	24	56
2300	28	28	28	28	28	28	28
2400	1	0	0	0	0	0	1

40% screening rate

	1-Mon	2-Tue	3-Wed	4-Thu	5-Fri	6-Sat	7-Sun
0600	51	51	51	51	51	51	51
0700	863	924	809	880	809	1026	813
0800	721	574	638	701	612	665	758
0900	752	728	743	775	751	836	755
1000	663	648	720	671	666	683	739
1100	478	526	617	534	495	606	596
1200	252	182	161	186	164	262	177
1300	135	139	196	126	187	121	168
1400	184	177	227	116	261	185	207
1500	304	277	209	254	283	282	231
1600	276	167	145	201	175	229	166
1700	93	99	93	103	86	96	102
1800	321	284	298	300	281	273	305
1900	209	202	101	183	119	105	145
2000	494	510	412	509	397	406	437
2100	466	342	414	377	374	356	457
2200	86	41	78	43	40	36	86
2300	54	54	54	54	54	54	54
2400	2	0	0	0	0	0	2

Figure 5.2.: Heat and number maps of peak screening times by screening rate. Cell colour intensity reflects the size of the number in each cell.

very strict biosecurity protocols, thus making these flights rather low risk and very few passengers on these flights generate alerts.



	1-Mon	2-Tue	3-Wed	4-Thu	5-Fri	6-Sat	7-Sun
0600	449	449	449	449	449	449	449
0700	4261	4144	4074	3875	3929	4237	3971
0800	2551	2254	2231	2337	2242	2156	2578
0900	2603	2496	2440	2548	2459	2542	2514
1000	1985	1781	1973	1879	1850	1787	2078
1100	1450	1637	1588	1700	1429	1625	1902
1200	879	627	564	617	612	640	662
1300	444	451	469	446	485	471	487
1400	432	294	451	231	461	434	417
1500	776	713	697	647	698	748	723
1600	476	396	449	330	422	477	487
1700	493	513	447	562	429	502	548
1800	1774	1457	1504	1615	1306	1281	1552
1900	655	562	532	504	565	482	623
2000	1169	1273	1109	1222	1113	1152	1165
2100	1172	969	1067	1067	1068	1019	1244
2200	498	302	432	335	317	247	480
2300	385	385	385	385	378	378	385
2400	20	0	0	0	0	0	20

Figure 5.3.: Heat and number map of total passenger movements. Cell colour intensity reflects the size of the number in each cell.

Additionally, the times of day that would see the most demand also depends on the choice of cutoff. We see that the busy time slots come from flights where the majority of the flight should be directed to the screening channels. Table 5.1 shows the flights with the highest number of passengers being directed to screening with a cutoff of 25% and we see that these are consistent with the darker periods in the heat map (Figure 5.3). Within these flights, we see up to 86% of passengers being sent for screening (Table 5.1, computed as cutoff25 divided by Total).

We also see that the peak time for screening varies depending on the choice of cutoff. For example, when we use the 10% screening rate, we see a clear stripe where the highest risk passengers arrive around 10:00 whereas for 25% and 40% screening rate, the high risk passengers are distributed fairly evenly between 7:00 and 11:00. This is due, in part, to a large number of flights arriving from China at 10:00. The flights are found to be high risk by the profiling algorithm and a large proportion of the passengers on these flights are selected for screening. In contrast, when we increase the screening rate to 25%, the majority of the high risk flights at 10:00 were already being screened so the additional screening capacity goes towards flights at different times such as at 7:00, when flights from United States and United Arab Emirates arrive. It is this sort of behaviour that clearly demonstrates the impact of the choice of cutoff.

Table 5.1.: Examples of flights with a high proportion of passengers being directed to the screening channels for Sydney with a screening rate of 25%.

weekday	ETAround	flt	origincountry	cutoff25	Total
2-Tue	700	HU7997	China	128	149
6-Sat	700	HU7997	China	128	149
1-Mon	1000	CZ325	China	102	164
2-Tue	1000	CZ325	China	102	164
3-Wed	1000	CZ325	China	102	164
4-Thu	1000	CZ325	China	102	164
5-Fri	1000	CZ325	China	102	164
6-Sat	1000	CZ325	China	102	164
7-Sun	1000	CZ325	China	102	164
1-Mon	1600	CA173	China	98	126

6. Survey Sample Sizes

The larger a port is, the more flights it will be receiving as well as likely receiving passengers from a broader selection of citizenship countries, thus increasing the dimensionality of the problem and creating more sparsity. This is demonstrated in Table 6.1 which shows that the larger airports have more sparsity with Adelaide having twice the number of observations per cohort as Brisbane.

Table 6.1.: Endpoint observations from Jul 2015 – June 2018 per cohort per port to show the inevitability of sparsity. The first row is the number of unique cohorts for each port, the second row is the size of the endpoint survey and the third row is the number of endpoint survey observations per cohort (i.e. row 1 / row 2).

	ADL	**BNE**	**MEL**	**OOL**	**PER**	**SYD**
Unique cohorts	46,642	261,057	409,616	59,042	204,362	644,698
Endpoint survey size	7,659	21,496	28,207	7,372	18,386	55,908
Observations per cohort	0.164	0.082	0.069	0.125	0.090	0.087

Clearly, then, every observation that we can get from the endpoint survey is valuable and so we would not want to lose any of that information.

We examined how changing the sample size of the endpoint survey affects the model prediction. We sampled with replacement from the endpoint survey at 1%, 10% and 100% of its original size and used those new, smaller datasets to train models. We repeated this 50 times for each sample size and looked at how the model's ability to predict the non-compliance changes. Figure 6.1 shows this for Sydney and we see that although the mean model performance didn't change as we reduce the endpoint survey size, the variance increased as we reduced the endpoint survey size, and the loss is asymmetric, that is, the envelope below the curve contains more area than the envelope above the curve.

In addition to the increased variance in performance, we found that as the endpoint survey size reduced, the model would sometimes fail to fit. With the 1% endpoint survey this happened to approximately 25% of instances. Within the experiment we were able to simply resample when this happened however the same would not be true when profiling in real life.

It must be noted that the endpoint survey is useful for more than just training models. Common practice in biosecurity is to use an endpoint survey to monitor the leakage within a pathway and therefore the effectiveness of the border measures that are in place.

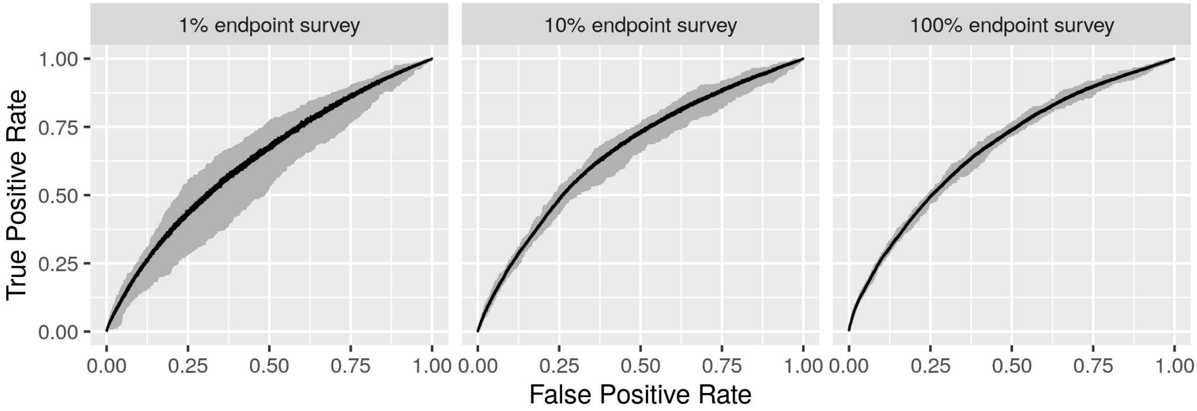


Figure 6.1.: Predictive model quality (using ROC curve) results for simulation experiment comparing three endpoint survey sizes. The black lines are means and the light-grey polygons are interval clouds of the simulation results.

7. Update Triggers

One vital part of the profiling activity is that of knowing that the profiles are working. The Department generates profiles at a regular time period such as once per year or once every 6 months and these are applied to all passengers until the profiles are regenerated. Generating profiles can be costly in terms of time due to the manual work involved in pulling together the necessary datasets so we do not want to be profiling unnecessarily frequently, however we do not want to be using profiles that are out of date and no longer fit the data.

Figure 7.1 shows how the performance of a model changes over time. The model was trained using data from July 2015 – June 2016 and then tested across the following three years. The ROC plot indicates that there was no substantial change in profile setup over this time as there was no noticeable decrease in profile quality. That being said, a refresh of profiles once per year is not an unreasonable task and that would allow for changes in flight numbers to be taken into account.

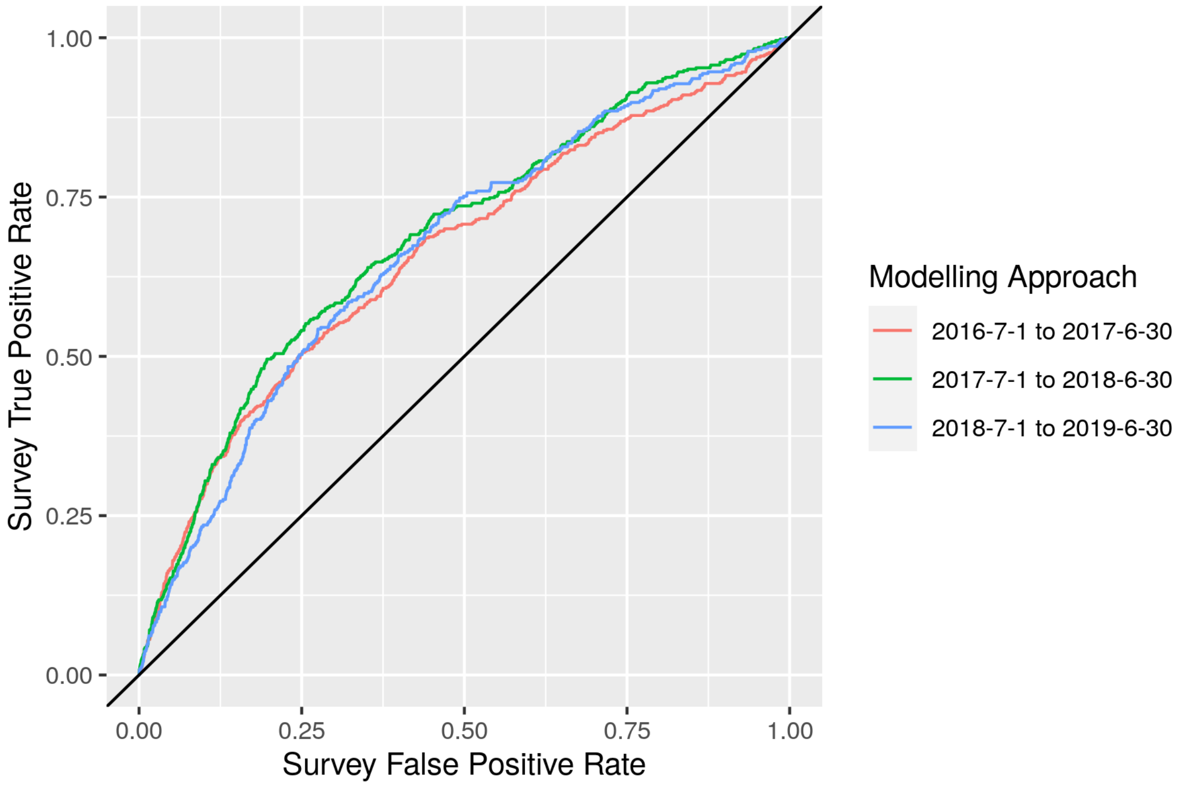


Figure 7.1.: Predictive model quality (using ROC curve) results comparing a model fit using 2015/16 FY data and tested over three different time periods (2016/17 FY, 2017/18 FY, and 2018/19 FY data.

If we are going to retain the profiles for such a long time, however, we should have some measure that we are able to easily check more frequently to provide some indication that the profiles are working. The endpoint survey data are more readily accessible than some of the other datasets so we recommend to use this as a way of monitoring our profiles. In particular, we used CUSUM control charts (Woodall & Ncube, 1985) on the citizenship country and flight number variables of all of the endpoint data. These are the most likely to change suddenly for reasons such as policy change, new flights being introduced or airlines swapping flight numbers between routes. A more sophisticated version would be to weight the endpoint data by the population weight corresponding to the channel from which it was taken, but we did not do so for this pilot case study.

The CUSUM control chart is a way of measuring when a variable of interest is consistently above or below the mean. Say the variable of interest (in this case, the proportion of passengers in the endpoint survey that were non-compliant) is X_i at the i^{th} month, then we define the high side cumulative sum is defined as

$$\mathrm{SH}(i) = \max[0, \mathrm{SH}(i-1) + X_i - \mu - k] \tag{7.1}$$

and the low side cumulative sum is

$$\mathrm{SL}(i) = \min[0, \mathrm{SL}(i-1) + X_i - \mu + k] \tag{7.2}$$

where μ is the mean of the X_is and k is a multiple of the standard deviation of the X_is, in this case 0.25sd. We also define the action limits to be 3sd. This part would all be calculated on the time period that the model was trained on, in our case using July 2015 to June 2018 and then the upper and lower cumulative sums would be calculated for new data instances.

Using this method, we found a number of types of control chart emerging. Figure 7.2 shows examples of some of the interesting control chart types. The top row is the control charts and the bottom row is the corresponding actuals. The vertical blue line is a date divider so the means and standard deviations were calculated using the data points to the left of the blue line and we are interested in instances to the right of the blue line that go outside our control limit. The control limit is shaded grey and the points are red on both plots when the cumulative sum goes outside of the control limits. Ordinarily, when the cumulative sum goes outside the control limit, that would be an indicator to retrain; and indeed, CUSUM is well placed to solve problems that have non-normality, are highly skewed or are heavy tailed (Stoumbos & Reynolds, 2004) as well as problems with binomial variables (Gan, 1993). However, the extreme sparsity of data for many of the characteristics can lead to *false alarm* warnings from the control charts which is why we only used them in the context of the actual data. Indeed, Gan (1993) showed how CUSUM control charts with small sample sizes pick up on changes in the decision variable much more quickly than those with large sample sizes.

Looking at the first column in Figure 7.2, we see that the majority of months have had $X_i = 0$ and in fact, when we examine the data more closely, we find that there have only been 82 passengers selected for the endpoint survey across the whole 3 years of training data, making for a sparse dataset. Similarly, the second column is an even more extreme case where there are no cases of non-compliance in the training data at all. This gives us a standard deviation of 0 so when a single case is discovered it goes way out of the control limit immediately. In neither case would it be appropriate to retrain the profiles.

The third and fourth columns however, are far more interesting. The third column shows a clear trend to passenger risk decreasing and the scatter plot of actuals shows that all the points in the test set are below the mean of the training set, meaning that there has been a sustained reduction in risk. Similarly for the final column, there appears to be a sustained reduction in risk once again, apparent in both the scatter plot and the control chart.

We further considered an alternative type of control chart: namely, the p-control chart (Duclos & Voirin, 2010). This is specifically designed for binomial data and so might be a good fit to our problem. Similar to the classic control chart, the p-control chart acts as an indicator of any single measurement being outside what we would expect by defining control limits. These are simply defined as

$$\mathrm{CL}_p = \bar{p} \pm 3\sqrt{\frac{\bar{p}(1-\bar{p})}{\bar{n}}} \tag{7.3}$$

where $\bar{p}$ is the mean probability of failure across all the data and $\bar{n}$ is the mean number of observations taken per time period (in our case, month). Our problem has much more variation in normal times than the problems that p-control charts would usually be used for so we increase the control limits so that it is 5 times the square root rather than 3.

The p-control chart explicitly defines a criterion for when there is sufficient data for the method to be used and this may be of particular use in our case. The criterion is that $\bar{n}\bar{p}$ must be sufficiently large, in particular

$$\bar{n}\bar{p} > 5 \tag{7.4}$$

Indeed, from the first set of control charts in Figure 7.2, the first two countries are deemed not to have sufficient data, thus eliminating the two weak examples of the four and the resulting control charts for the remaining two are meaningful, see Figure 7.3.

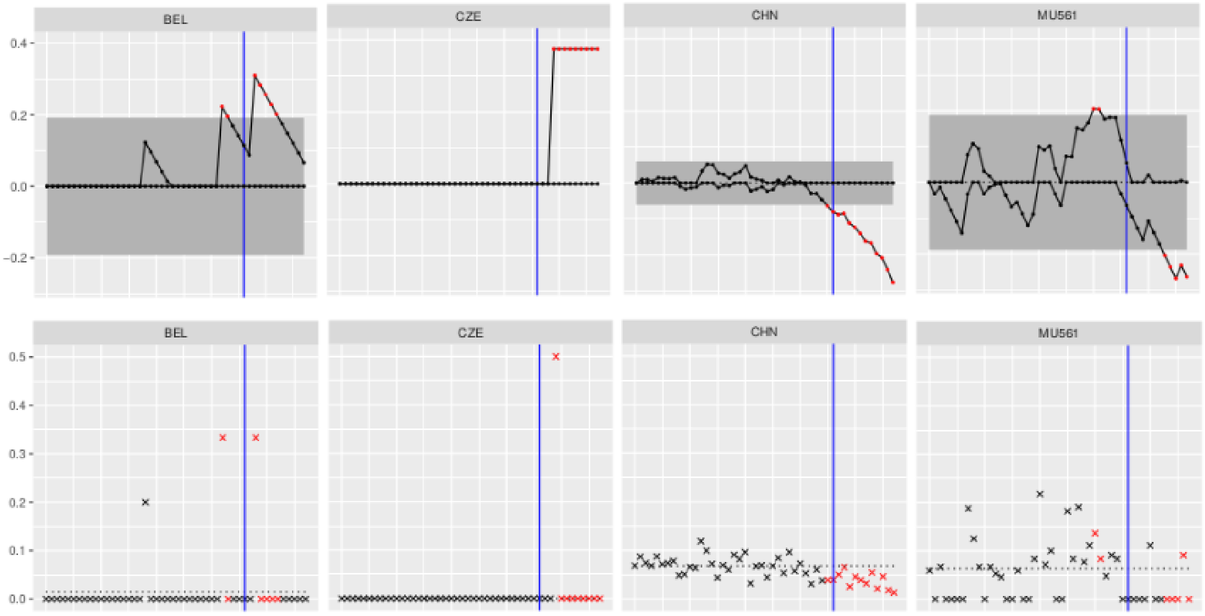


Figure 7.2.: Example CUSUM control charts and scatter plots of actuals. The x-axis is months, and the y-axis is detection rate across all of the endpoint data.

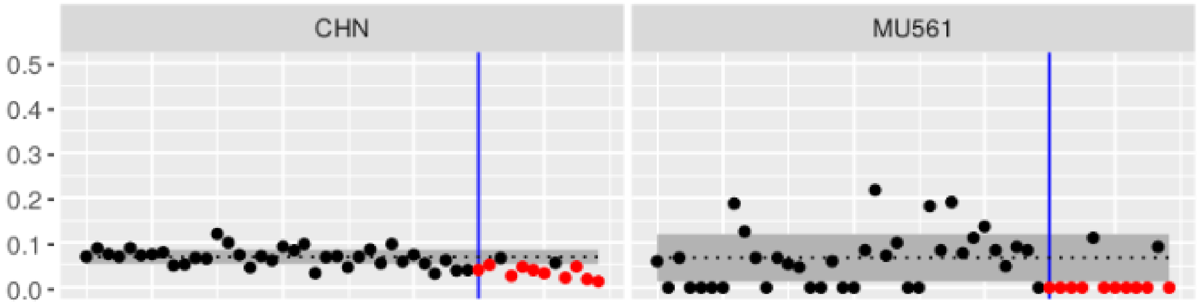


Figure 7.3.: Example p-control charts and scatter plots of actuals for a country (left panel) and a flight (right panel). The x-axis is months, and the y-axis is detection rate across all of the endpoint data.

The p-control chart is not without fault however, because since it is not a cumulative method, it can be very sensitive picking up small and unsustained changes in the endpoint survey data. When considering the cusum charts, we only considered the 50 largest citizenship countries and 50 largest flights and the combination of both criteria appears to catch all of the obvious *false alarm* charts so that we can focus on those cohorts that might really have an impact on the profiling.

The interpretation of control charts is key to their success. They will only work for the larger flights and countries and so should be used in combination with a system that alerts us to flights that have not previously been seen to account for some of the new risk that might be introduced by smaller flights. It is worth noting, however, that it is generally the larger cohorts that have a bigger impact on the profiling models so in fact retraining the model when a small cohort has a change in risk may not be appropriate.

8. Profile Automation (Technical)

This chapter is more technical than the balance of the report, and may safely be omitted by a reader with non-technical agenda. A key deliverable of this project was that of profile automation. We have provided a number of scripts along with this report to assist in the adoption of the method. The scripts generate profiles from data, run simple experiments to determine whether to include additional variables and generate CUSUM control charts for the largest citizenship countries and flights.

8.1. Generate Profiles

The profile generation scripts (`profile_generation.R`) take as input two data sets; `MAPS_totals.csv` contains the channel volumes and `train_data.csv` is an aggregation from the cohort volumes, endpoint survey and interceptions for a given port. There are a number of necessary columns:

- flt (flight number)
- origin (citizenship country)
- gender
- age
- ***any other profiling characteristics that might be introduced at a later date should be here too***
- total (v_l: total volume of passengers)
- svyexit (n_{el}: number of passengers in the endpoint survey who came from the exit channel)
- svyscrn (n_{sl}: number of passengers in the endpoint survey who came from one of the screening channels)
- lkgexit (x_{el}: number of passengers in svyexit who were found to be non-compliant)
- lkgscrn (x_{sl}: number of passengers in svyscrn who were found to be non-compliant)
- szrscrn (x_l: number of passengers who were found to be non-compliant at the screening stage)
- svy ($n_{el} + n_{sl}$: number of passengers in the endpoint survey)
- svylkg ($x_{el} + x_{sl}$: number of non-compliant passengers in the endpoint survey)

When generating profiles to be used in practice, use the most recent data.

A number of key outputs are generated along with the profiles. The first are the random effects plots for the screening model and the final non-compliance model. These show the contribution to a passenger's risk score made by the various flights and citizenship countries. These are saved at `plots/randomeffects_screeningmod.pdf` and `randomeffects_apprmod.pdf` respectively. A sample of the random effects plot for approach rate for citizenship country can be found in Figure 8.1. Here the dots show the relative risk[1] of the passengers corresponding to the profile variable on the y-axis. The bars are intervals that report how precisely the risk is estimated; points that correspond to wide bars are uncertain. In this example, the relative risk of IND is well estimated, and the relative risk of PRY is poorly estimated.

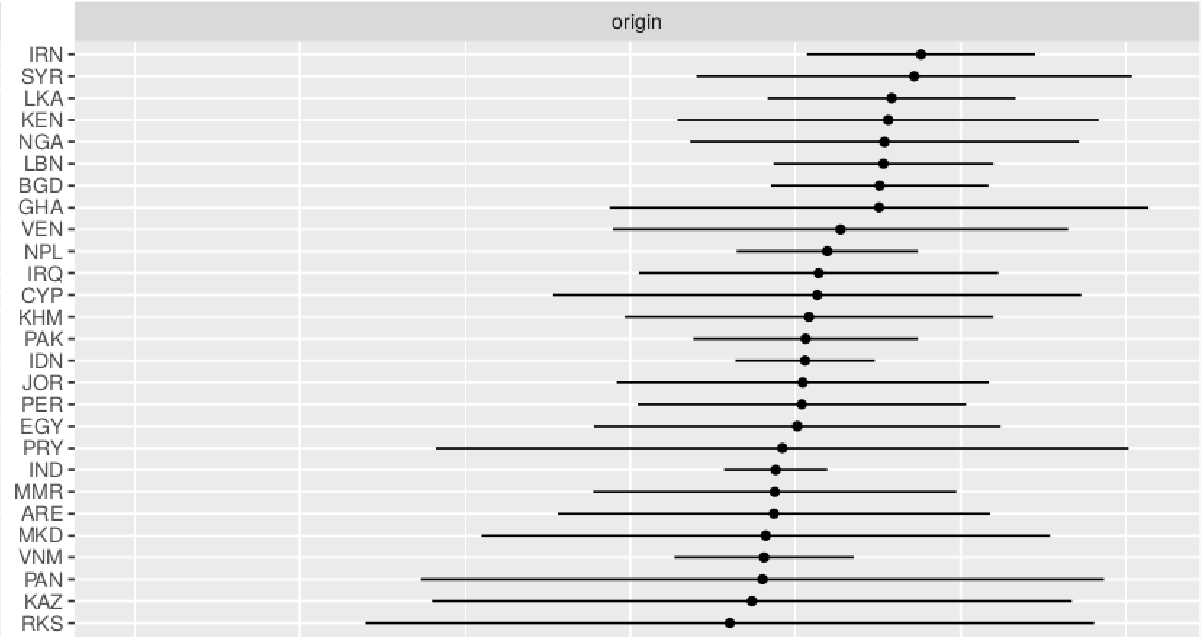


Figure 8.1.: Sample of random effects plots, shown here for the approach rate model on citizenship country. The black dot is the relative risk of the passengers corresponding to the profile variable on the y-axis. The bars are intervals that report how precisely the risk is estimated; points that correspond to wide bars are uncertain..

Additionally, the output in the R console provides a summary of each models. Of particular interest is the fixed effects table which looks something like Table 8.1. This table provides technical statistical feedback on the quality of the models; detailed explanation of which is beyond the remit of this report. Briefly, the estimate column shows the coefficient for that variable in the model (in the case of binary variables such as gender, it simply chooses a value to be the baseline so genderF would just be the inverse of genderM). The Pr(>|z|) shows the significance or the degree of certainty we hold that the variable should be included in the model.

We also save the training data which includes the various calculation steps as the process is worked through as well as some raw figures for context. This is found at `output_data/full_train_data.csv`.

The script then goes into the profile generation stage. At the very top of the script there is a variable called `n.pax.toscreen` which is the number of passengers that

[1] Technically, the x-axis reports the log-odds.

Table 8.1.: Fixed effects summary for fitting models to cohort data. This is what will be seen in the output of the R console when running `profile_generation.R`.

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-2.15	0.147	-14.6	< 2e-16 ***
genderM	-0.23	0.0454	-5.0	4.68e-07 ***
age	0.01	0.001	6.3	1.88e-10 ***

we would like to be sending to be screened per day (purely from the profiling activity). There is also another variable called `n.years.in.training.data` which is necessary to be correct in order to calculate the correct cutoff. These are used to calculate which cohorts we can afford to screen if the total number of passengers in the cohorts is similar to that in the training data. This allows us to produce static profiles which is what is required in practice although with this method we will never be able to truly predict how many passengers will be flagged for screening. We suggest a trial-and-error approach to the choice of cutoff — to start with a number that seems sensible and then if it is regularly producing more passengers for screening than we are able to handle in practice then to reduce it. Likewise, if it is regularly producing far less passengers to be screened than our capacity allows then we can increase it. The goal is to choose a cutoff that sits just within the capacity allowance without exceeding it.

8.2. Determine a cutoff

The script `determine_cutoff.R` produces heat maps similar to those found in Section 5. By changing the variable screenrate, we can see what the screening load would look like for different cutoffs. These can be compared with the model effectiveness plot from `compare_models.R` to inform a decision of cutoff for `profile_generation.R`. This script requires a number of input files:

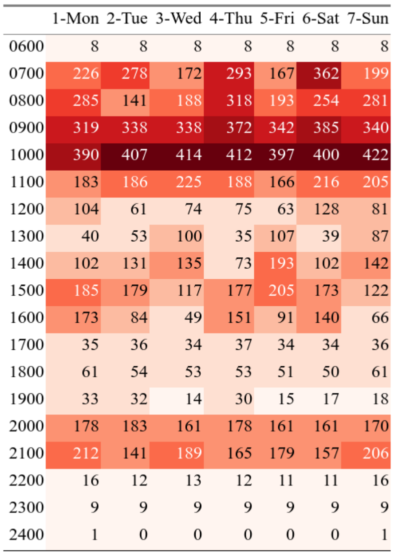
1. `input_data/airports_countries.csv`: this is a publicly available database showing the country that each airport belongs to. This data should remain reasonably static so using the file that comes with the package shouldn't be a problem.

2. `input_data/sad_all.csv`: this is the flight schedules data.

3. `input_data/vol_data.csv`: this is the cohort volume data (ideally as up-to-date as possible).

4. `models/glm_approach_rate.Rdata`: this is a pre-trained model that we will use to predict. Run `profile_generation.R` if such a file does not exist, or to obtain a more up-to-date model.

Once `determine_cutoff.R` has finished running, a plot similar to Figure 8.2 will appear in the RStudio Viewer. An iterative process of changing the variable `screenrate` and rerunning the script should help the user to decide an appropriate cutoff.



	1-Mon	2-Tue	3-Wed	4-Thu	5-Fri	6-Sat	7-Sun
0600	8	8	8	8	8	8	8
0700	226	278	172	293	167	362	199
0800	285	141	188	318	193	254	281
0900	319	338	338	372	342	385	340
1000	390	407	414	412	397	400	422
1100	183	186	225	188	166	216	205
1200	104	61	74	75	63	128	81
1300	40	53	100	35	107	39	87
1400	102	131	135	73	193	102	142
1500	185	179	117	177	205	173	122
1600	173	84	49	151	91	140	66
1700	35	36	34	37	34	34	36
1800	61	54	53	53	51	50	61
1900	33	32	14	30	15	17	18
2000	178	183	161	178	161	161	170
2100	212	141	189	165	179	157	206
2200	16	12	13	12	11	11	16
2300	9	9	9	9	9	9	9
2400	1	0	0	0	0	0	1

Figure 8.2.: Example heat and number map of required screening count for passengers based on a recommended set of profiles, output from `determine_cutoff.R`.

8.3. Compare models

When comparing models, it is essential to always use a training dataset and a testing dataset and keep them completely separate. Both sets should have the same columns as outlined in Section 8.1 for generating profiles. We recommend dividing the data by date so that it is immediately obvious if something has gone wrong. It is also important to note that we are using a generalised linear mixed effects model with a binomial family, meaning that it expects the response variables to be integer. All of the underlying information that we are modelling is binomial however because we needed to use our own predictions at various points, these outputs were not necessarily integer. As such, although the model works perfectly well, we cannot use some of the traditional indicators of model fit such as AIC. Instead, we recommend the use of ROC curves and other metrics used in machine learning.

The file `compare_models.R` uses a separated training and test set to properly assess the effectiveness of the model. If the department wanted to do consider alternative models in the future, they could follow the same structure found in this file. The model effectiveness plot (the plot that we have used throughout this report to assess model performance) saves to the file `plots/model_effectiveness.jpg`. In order to incorporate a different model, we just need to join its predictions into the dataframe test.data before it is expanded and then add it into the plot.raw.AUCs function as demonstrated in the comments of the file.

Models must be trained and tested on the same data to be appropriately compared and there must be no overlap between the training data and test data.

8.4. Update Alerts

The update alerts should be interpreted with care in line with the recommendations in Section 7. The `update_alerts.R` file requires separate training and test data with the same columns as in Section 8.1. It outputs two files: `plots/pcontrolcharts.pdf` and `plots/cusumcontrolcharts.pdf`. These are control charts for the largest citizenship countries and flights (there are many plots on each page). The CUSUM chart takes the top 50 most sampled countries and flights and plots those whereas the p-chart uses the criteria outlined in Equation 7.4. For a risk to have sufficiently changed that we might consider retraining the model, we require:

- the country or flight to be present on both sheets,
- there to be several red dots amongst the test data (to the right of the vertical line),
- there to be relatively fewer red dots amongst the training data (to the left of the vertical line), and
- there to be no obvious seasonal explanation for the increase in red dots.

In particular, in order to justify retraining the models, we need a possible change to be sustained and different enough to the training data that we would obtain a different model by retraining.

9. Version Control

The automated profiling system may have multiple people doing analyses on it at the same time making small changes that it is important to track. Effective version control is essential in code sharing or when working with scripts of high value. This section outlines some of the best practices in version control and introduces some useful tools for version control in programming.

9.1. What is Version Control?

Software development involves revision, repair, and evolution of code, and often requires parallel effort by collaborating individuals. A common consequence of this is that multiple, potentially conflicting versions of code are produced, leading to confusion about the exact nature of changes, as well as their authorship, order, and motivation. Further, in the absence of an effective backup protocol, files being overwritten during experimentation can lead to the loss of previous versions of the work. A common response to these concerns is to use descriptive filenames or code comments that provide relevant context. However, this *ad hoc* form of local version control is error prone and unwieldy[1].

Formal version control systems (VCSs) provide an effective solution to the problem. These systems comprise one or more databases (*repositories*) that record the current state of the codebase, as well as a complete history of changes made throughout development. Modern collaborative web apps have begun to integrate version control into their software. For example, Google Docs and Microsoft Word Online both feature a revision history that permits users to compare (and restore, if desired) historical versions of a document. Dedicated version control systems exist as desktop clients; these systems facilitate structured, consistent development workflows, parallel development, painless merging of edits (including conflict resolution), and version comparison/restoration. Combined with cloud syncing (e.g. GitHub), VCSs provide security and convenience, and are considered indispensable by many in professional software development.

The rich 50-year history of version control systems saw them evolve through three broad generations with increasing levels of flexibility and sophistication (Raymond, n.d.). The first VCS gaining popular adoption was *Source Code Control System* (SCCS; Rochkind, 1975), released in 1972. Born in the pre-internet era, SCCS lacked network support—users were forced to share a single local machine. Importantly, users were limited to editing one file at a time, and file locks prevented multiple users editing a given file simultaneously. Despite falling out of favour as the next generation of VCSs appeared (and arguably reaching obsolescence since), SCCS pioneered concepts and practices that remain in use today, including branching, version numbering style, efficient storage of deltas (namely, instructions describing sequential file changes relative

[1] e.g., `corefunctions-final-Jan2020-v2-REVIEWED-jb-CORRECTED-final.R`

to the initial version) rather than complete copies of code, checksumming changes to verify source integrity, and the requirement for descriptive log messages summarising the nature of the changes. In 1982, a decade or so after the initial release of SCCS, *Revision Control System* (RCS) arose to address some of its shortcomings (Tichy, 1982). In particular, RCS implemented a more efficient method of storing version changes (allowing faster recreation of specific versions), and introduced functionality that allowed multiple, potentially conflicting, versions of a file to be merged. However, RCS remained a local, file-locking system best suited to single-developer projects.

Effective programmer collaboration was enabled by the second generation of VCSs, which permitted concurrent editing of files stored within a single network-connected file repository. With these centralised systems, users retrieve (check out) the latest revision of one or more files, edit them on their own computers, and send (commit) modified versions back to the server. Files can be checked out and edited by multiple users simultaneously; this proves highly practical when working in a team environment, but requires that any changes committed to the repository are incorporated (merged) into a user's modified version before that version is accepted by the repository. Prominent centralised VCSs include *Concurrent Versions System* (CVS, the first centralised VCS; Grune, 1986) and *Subversion* (SVN; Collins-Sussman *et al.*, 2004). While CVS was very popular, dominating the open-source version control marketplace for around a decade, SVN introduced key improvements: it was faster, supported versioning of binary files, recognised file renaming (i.e., linking revision history with the new name), and tracked revision of an entire commit (i.e., changes to a batch of files) rather than of individual files. In general, this paradigm of centralised, network-enabled VCSs represented a leap forwards in version control, greatly simplifying revision tracking for team-based projects. Their key drawback, relative to the local, file-locking, first generation systems, was their requirement for full time connectivity to the server hosting the repository—changes could not be committed otherwise.

The third and current generation of VCSs combined the benefits of local and centralised systems. These *distributed* systems are characterised by each user maintaining a complete copy (clone) of the source code, which they modify locally and sync with a remote repository if and when desired. Clearly, this is far more convenient than the earlier systems. Local changes are fast and obviate the need for an uninterrupted connection to a remote server, yet collaborative work is facilitated by periodically updating the remote, shared repository. In addition, the system does not hinge on the integrity of a single instance of the repository; rather, each developer has a complete copy that can serve as a backup. Modern distributed VCSs provide powerful branching and merging functionality, further supporting efficient parallel development. Notable distributed systems include *Git* and *Mercurial* (both open source and initially released in 2005), with Git being by far the most popular system today[2].

While master repositories (remotes) can be stored on private, network-connected drives, there are several cloud platforms available that provide free, dedicated hosting for this purpose. These are mostly geared towards Git repositories; the most popular include GitHub (over 40 million users and 100 million repositories), Bitbucket (over 10 million users and 28 million repositories), and GitLab. Comparisons of these competing platforms are available elsewhere. Users sync their repositories to the cloud, with the option of flagging them as private such that they are only visible to autho-

[2] As reported by Rhodecode in the results of their 2016 survey of VCS popularity (`https://rhodecode.com/insights/version-control-systems-2016`).

rised viewers. The services offer continuous-integration tools that test code whenever it is sent to the cloud, alerting the user to any issues. Key features common to most repository hosting platforms include issue tracking, whereby bugs and potential enhancements can be recorded, discussed, and tracked; and a code review process, which allows users to request that their proposed changes are reviewed and incorporated by the repository maintainer. For public repositories, this fosters engagement with the wider developer and user communities.

In summary, VCSs simplify software version maintenance and debugging, and facilitate efficient solo and collaborative development. Version control systems (particularly when used in conjunction with a repository hosting platform):

- track a project's history, including who made a change, and when and why the change was made;
- compare with or revert to older versions; manage version releases;
- share, contribute, and discuss code;
- isolate feature development in parallel branches;
- merge those features back into the original branch;
- maintain backups; and
- identify exactly how/when a bug was introduced, and track its status and resolution.

Software development, whether independently or as part of a team, greatly benefits from incorporating a VCS into the workflow.

9.2. A Typical Git + GitHub Workflow

A typical Git version control workflow involves the following steps. Note that this assumes interaction with Git via a command line interface (e.g. Bash, Git Bash for Windows), although graphical user interfaces are available. It is also assumed that Git is installed (verify with `git -version` at the command line); pushing to GitHub requires registration at the GitHub website.

1. Initialise a repository. From a command line interface this would be done with `git init`. This marks the current folder as a repository.
2. Create/delete/edit files. For example, we might create two files: myfile1.txt and myfile2.txt.
3. Stage changed files to be committed to the local repository. Committing a set of changes records that set of changes as a revision. This is done with `git add myfile1.txt`. Multiple files can be staged with e.g., `git add myfile1.txt myfile2.txt`, and all files can be staged at once with `git add -all`.

4. Commit the staged changes to the repository, with a commit message describing the change: `git commit -m "initial commit"`[3].

5. We can make additional changes to the same files, or remove them, and commit those additional changes. This usually involves repeating the process of adding them and committing the changes (steps 3–4 above). To commit changes to files that have been previously tracked (staged), the `-a` flag can be added to the commit command to add the changed files and commit the changes simultaneously: `git commit -a -m "remove file"`.

   The complete log of changes is given by `git log`. Each commit is identified by a unique hash, which can be used to view (and revert to) the state of the repository at that time, and to compare files as they were at different commits (e.g. `git diff hash1 hash2`, where `hash1` and `hash2` are the first few characters—sufficient to uniquely identify commits in the repository—of the respective commit hashes).

6. Assuming the repository is associated with a remote repository, e.g., on GitHub, the command `git push` would push all committed changes to that remote. If the remote has commits that are not yet incorporated into the local repository, they can be retrieved with `git pull`.

An abundance of tutorials and documentation can be found online to supplement this brief introduction. A useful starting point is the book *Pro Git* (Chacon & Straub, 2014), available free of charge at `https://git-scm.com/book`.

[3] Commit messages should begin with a short line (less than around 50 characters) summarising the changes, and usually begin with a present-tense verb (e.g. Implement, Fix, Add). Further detail can be added by leaving a blank line and then providing that detail as free text. See introductory Git resources for additional guidance (e.g. *A Note About Git Commit Messages*, by Tim Pope)

10. Discussion and Conclusions

This report demonstrates that fitting statistical models can offer improved passenger profiling performance over the department's current methodology, which calculates a risk score for each unique combination of age group, gender, flight number and passport country. The survey and interception data used to identify risk and construct these profiles are very sparse, which causes difficulties as there are many combinations which have a small amount of information, or none at all. Models help solve this problem by using the data to identify the key attributes that contribute to non-compliance, and are able extrapolate these to less well represented combinations of attributes. Given a cohort of male passengers from country X on flight Y and aged between 20-30, even if we had zero interception records or endpoint survey results a model allows us to estimate the risk of these passengers being non-compliant based on how male passengers, or passengers from country X have behaved in other cohorts in the dataset.

The key challenge with profiling passengers is missing data, and our proposed method recovers the screening rate for each cohort by fitting a GLMER to the endpoint survey data, which provides information on how passengers were screened. As we note, the endpoint survey contains screening channel bias, so raking is also done after fitting the model to correct for this. With these more accurate screening rates, we can then fit a model to identify non-compliance. This allows us to use the much larger interception dataset in building the non-compliance model. We test this against a number of alternative methods, including one which ignores this screening rate step and directly builds a non-compliance model based on the endpoint survey and a number of alternatives to GLMER in the screening and non-compliance models.

While the results from this project are satisfying, it has opened up more questions to be answered as us so often the case. We used high risk ABM as our definition of non-compliance in both the training and testing phases. While it is high-risk material that we aim to find, it may be that non-compliant behaviour acts as an effective predictor of risk and using all passengers carrying ABM to train the model might improve the prediction of high risk non-compliance in the test set.

We also have not put thought towards calculating the uncertainty of the profiles. With cohorts of such varying size this would certainly be an interesting metric to examine. In particular, a potential perceived issue with having smaller cohorts is that if non-compliance is found in these cohorts, they can end up with high risk scores due to their low sample size and so the highest risk cohorts often end up being very small cohorts that are not frequently coming through the airport. This may seem to be a waste of an alert if such passengers rarely pass through however the value of the new data obtained by screening is greater within the small cohorts. Additionally, models using the characteristics of the cohort rather than the cohort itself should reduce this effect of small cohorts appearing to be the most risky because of a single interception. An interesting piece of future work would be to look at the confidence intervals of our predictions. This is not trivial due to the fact that we are using consecutive models and so there are a number of options for how we carry the confidence intervals through the stages of the model.

There is also ample opportunity to also consider declarants - to investigate if screening declarants is a better use of screening resource or prioritising non-declarants is more efficient. Finally, there are a great number of similarities between the problem of profiling air passengers and the problem of profiling mail — there is certainly opportunity to bring some of these techniques into the mail space.

Bibliography

Barocas S, Hardt M, Narayanan A (2017) Fairness in machine learning. *NIPS Tutorial*, **1**.

Bates D, Mächler M, Bolker B, Walker S (2015) Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*, **67**, 1–48. doi:10.18637/jss.v067.i01.

Bishop YM, Fienberg SE, Holland PW (2007) *Discrete multivariate analysis: theory and practice*. Springer Science & Business Media.

Blumer A, Ehrenfeucht A, Haussler D, Warmuth MK (1987) Occam's razor. *Information processing letters*, **24**, 377–380.

Carlin BP, Louis TA (2008) *Bayesian methods for data analysis*. CRC Press.

Chacon S, Straub B (2014) *Pro Git*. Apress.

Chouldechova A, Roth A (2018) The frontiers of fairness in machine learning. *arXiv preprint arXiv:1810.08810*.

Collins-Sussman B, Fitzpatrick BW, Pilato CM (2004) *Version Control with Subversion*. O'Reilly.

Deming WE, Stephan FF (1940) On a least squares adjustment of a sampled frequency table when the expected marginal totals are known. *The Annals of Mathematical Statistics*, **11**, 427–444.

Deng H, Runger G, Tuv E (2011) Bias of importance measures for multi-valued attributes and solutions. In: *International conference on artificial neural networks*, pp. 293–300. Springer.

Derous E, Ryan AM (2019) When your resume is (not) turning you down: Modelling ethnic bias in resume screening. *Human Resource Management Journal*, **29**, 113–130.

Duclos A, Voirin N (2010) The p-control chart: a tool for care improvement. *International Journal for Quality in Health Care*, **22**, 402–407.

Friedler SA, Scheidegger C, Venkatasubramanian S, Choudhary S, Hamilton EP, Roth D (2019) A comparative study of fairness-enhancing interventions in machine learning. In: *Proceedings of the conference on fairness, accountability, and transparency*, pp. 329–338.

Friedman JH (2001) Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pp. 1189–1232.

Gan F (1993) An optimal design of CUSUM control charts for binomial counts. *Journal of Applied Statistics*, **20**, 445–460.

Garcia M (2016) Racist in the machine: The disturbing implications of algorithmic bias. *World Policy Journal*, **33**, 111–117.

Gianfrancesco MA, Tamang S, Yazdany J, Schmajuk G (2018) Potential Biases in Machine Learning Algorithms Using Electronic Health Record Data. *JAMA Internal Medicine*, **178**, 1544–1547. doi:10.1001/jamainternmed.2018.3763. URL https://doi.org/10.1001/jamainternmed.2018.3763.

Greenwell B, Boehmke B, Cunningham J, Developers G (2020) *gbm: Generalized Boosted Regression Models*. URL https://CRAN.R-project.org/package=gbm. R package version 2.1.8.

Grune D (1986) *Concurrent versions systems, a method for independent cooperation*. VU Amsterdam. Subfaculteit Wiskunde en Informatica.

Hastie TJ (1992) Generalized linear models. *Statistical models in S*.

Holstein K, Wortman Vaughan J, Daumé III H, Dudik M, Wallach H (2019) Improving fairness in machine learning systems: What do industry practitioners need? In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pp. 1–16.

Joh EE (2017) Feeding the machine: Policing, crime data, & algorithms. *Wm. & Mary Bill Rts. J.*, **26**, 287.

Lane SE, Gao R, Chisholm M, Robinson AP (2017) Statistical profiling to predict the biosecurity risk presented by non-compliant international passengers. *arXiv preprint arXiv:1702.04044*.

Meyer D, Dimitriadou E, Hornik K, Weingessel A, Leisch F (2020) *e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien*. URL https://CRAN.R-project.org/package=e1071. R package version 1.7-4.

Murphy KP, *et al.* (2006) Naive bayes classifiers. *University of British Columbia*, **18**.

Pal M (2005) Random forest classifier for remote sensing classification. *International journal of remote sensing*, **26**, 217–222.

R Core Team (2020) *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Rajkomar A, Hardt M, Howell MD, Corrado G, Chin MH (2018) Ensuring fairness in machine learning to advance health equity. *Annals of internal medicine*, **169**, 866–872.

Raymond ES (n.d.) Understanding version-control systems. URL http://www.catb.org/~esr/writings/version-control/version-control.html.

Rich ML (2016) Machine learning, automated suspicion algorithms, and the fourth amendment. *University of Pennsylvania Law Review*, pp. 871–929.

Robinson A, Chisholm M, Mudford R, Maillardet R (2015) *Biosecurity Surveillance: Quantitative Approaches*, chap. Ad-hoc solutions to estimating pathway non compliance rates using imperfect and incomplete data., pp. 167–180. CABI.

Rochkind MJ (1975) The source code control system. *IEEE transactions on Software Engineering*, pp. 364–370.

Stoumbos ZG, Reynolds MR (2004) The robustness and performance of cusum control charts based on the double-exponential and normal distributions. In: *Frontiers in Statistical Quality Control 7* (eds. Lenz HJ, Wilrich PT), pp. 79–100. Physica-Verlag HD, Heidelberg.

Suresh H, Guttag JV (2019) A framework for understanding unintended consequences of machine learning. *arXiv preprint arXiv:1901.10002*.

Sutton RS, Barto AG (2011) *Reinforcement learning: An introduction*. Cambridge, MA: MIT Press.

Tichy WF (1982) Design, implementation, and evaluation of a revision control system. In: *Proceedings of the 6th International Conference on Software Engineering*, ICSE '82, p. 58–67. IEEE Computer Society Press, Washington, DC, USA.

Triguero I, García S, Herrera F (2015) Self-labeled techniques for semi-supervised learning: taxonomy, software and empirical study. *Knowledge and Information systems*, **42**, 245–284.

Unser M, Aldroubi A, Eden M (1993) B-spline signal processing. i. theory. *IEEE trans-*

actions on signal processing, **41**, 821–833.

Williamson R, Menon A (2019) Fairness risk measures. In: *Proceedings of the 36th International Conference on Machine Learning*, vol. 97 of *Proceedings of Machine Learning Research* (eds. Chaudhuri K, Salakhutdinov R), pp. 6786–6797. PMLR, Long Beach, California, USA. URL http://proceedings.mlr.press/v97/williamson19a.html.

Wood SN (2017) *Generalized additive models: an introduction with R*. CRC press.

Woodall WH, Ncube MM (1985) Multivariate cusum quality-control procedures. *Technometrics*, **27**, 285–292.

Woodward JA, Bonett DG, Brecht ML (1990) *Introduction to linear models and experimental design.* Harcourt Brace Jovanovich.

Wright MN, Ziegler A (2017) ranger: A fast implementation of random forests for high dimensional data in C++ and R. *Journal of Statistical Software*, **77**, 1–17. doi:10.18637/jss.v077.i01.

A. Non-Compliance Metrics Algebraically

Algebraically, assume that we have L cohorts. Here, as for Section 4.1 and Algorithm 1, we define for cohort l,

v_l as the number of passengers;
e_l as the number of passengers that were exited (estimated);
s_l as the number of passengers that were screened (estimated);
x_l passengers are found to be non-compliant in screening;
n_{sl} screened passengers are reinspected in the endpoint survey; and
x_{sl} of them still have ABM;
n_{el} exiting passengers are reinspected in the endpoint survey; and
x_{el} of them still have ABM.

Then, in its simplest form, assuming that the endpoint survey is random and representative within the screened and exiting passengers,

$$\hat{e}_l = \frac{n_{el}}{\sum_{i=1}^{L} n_{ei}}, \text{ and } \hat{s}_l = \frac{n_{sl}}{\sum_{i=1}^{L} n_{si}}. \tag{A.1}$$

Then,

$$\text{NCR}_l = \frac{x_l}{\hat{s}_l} \tag{A.2}$$

$$\text{AR}_l = \frac{1}{v_l}\left(x_l + \frac{x_{el}}{n_{el}}\hat{e}_l + \frac{x_{sl}}{n_{sl}}\hat{s}_l\right) \tag{A.3}$$

$$\text{HR}_l = \frac{1}{v_l}\left(x_l + \hat{e}_l \frac{x_l}{x_l + \frac{x_{sl}}{n_{sl}}\left(\hat{s}_l - x_l\right)}\frac{x_{el}}{n_{el}}\right) \tag{A.4}$$

B. Raking

Raking takes data in the form of Table B.1 where there is some initial seed data on volumes by cohort and screening channel (the main body of the table) along with marginal totals for each cohort and screening channel. As in Table B.1, there is a discrepancy between the seed data and the marginal totals; raking iteratively rescales the seed data so that it matches each of the marginals, alternating between cohort volumes and screening volumes. This method has been shown to converge to the least squares estimates (Deming & Stephan, 1940) and maximum likelihood estimates (Bishop *et al.*, 2007) for the cohort/screening cell counts.

Table B.1.: Example of data used for raking. The values in the table cells come from the endpoint survey and the marginal totals from channel volumes (rows) and cohort volumes (columns).

	Cohort 1	Cohort 2	Cohort 3	Total
Screened	1	3	5	12
Not Screened	5	5	10	25
Total	9	10	18	

Of particular importance is that we know both sets of volumes and that they are equal to one another as in the example. This is our convergence criteria to meet the guarantees outlined above.