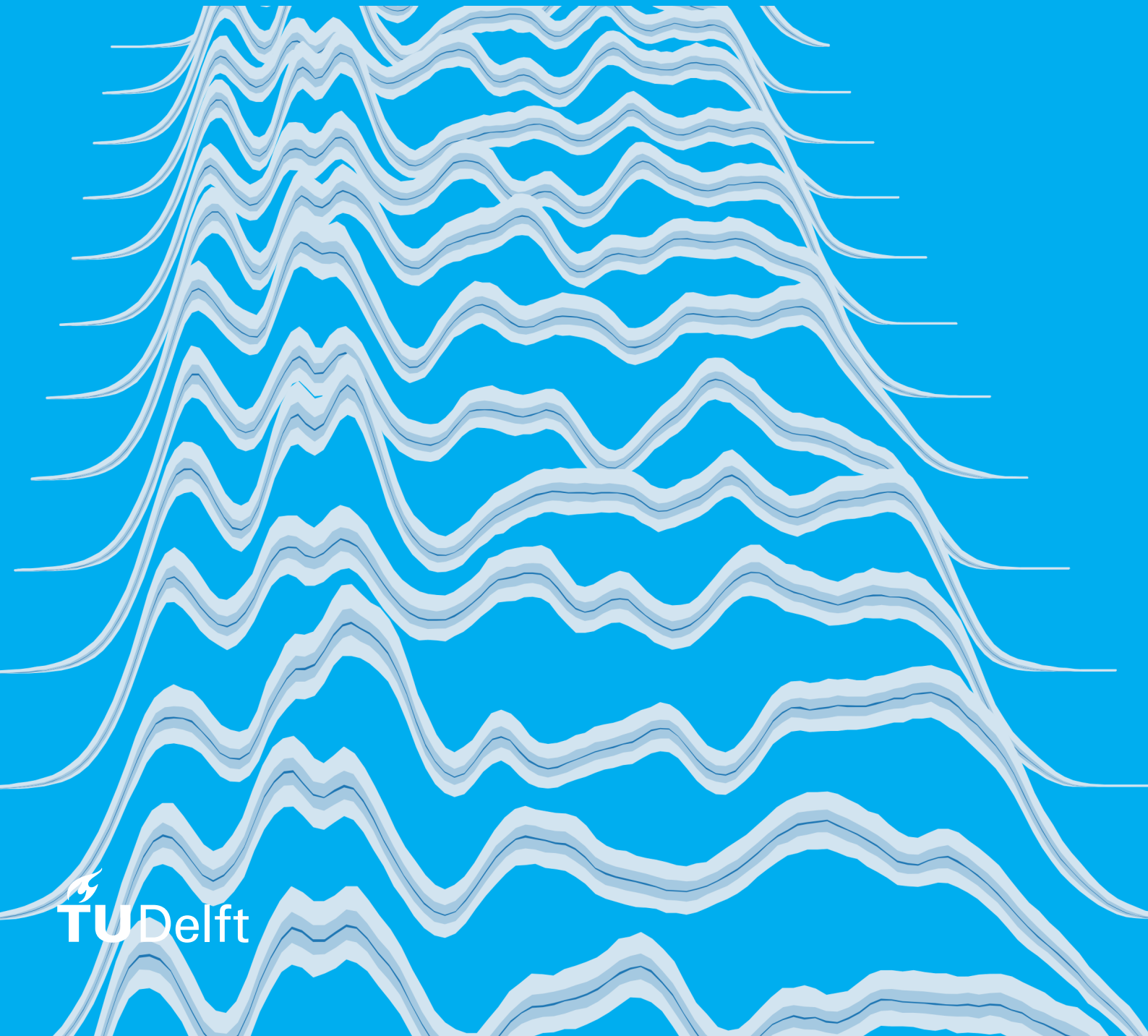
Master of Science Thesis

Real-time Probabilistic Passenger Arrival Forecasting

Mihaly Katona



TU Delft

Real-time Probabilistic Passenger Arrival Forecasting

by

Mihaly Katona

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Thursday January 18th, 2024.

Student number:	4537688
Project duration:	Jan 2023 – Jan 2024
Thesis committee:	Dr. O.A Sharpans'kykh
	Dr. B.F Lopes Dos Santos
	Dr. E. van Kampen

TU Delft

Acknowledgements

This thesis marks the culmination of over six years of study at Delft University of Technology and a year of dedicated work towards my Master's degree in Aerospace Engineering. It has been a long and tough journey, but one that has allowed me to grow into the person I am today. And for that, I'm incredibly grateful. Not to mention all the incredible people who have helped, guided and inspired me through all of it.

Firstly, I would like to thank Dr. Alexei Sharpans'kykh who has supervised this project. I'm particularly thankful for his support and encouragement with my slightly unconventional subject and approach. Providing insights, suggestions and, ensuring academic rigour, while allowing me to think outside the box.

This thesis would not have been possible without the continued and enthusiastic support from GRASP Innovations. I'm especially grateful for the trust and support that Robert and Michael have provided throughout this work.

I would also like to express my gratitude to Siert and Koen who have not only kept me company during the long days of work but also provided invaluable suggestions. Additionally, my thanks go to the rest of the people who joined our informal "Thesis Town". You made the time fly, making our sessions perhaps a bit too much fun at times.

Many thanks to my family for enabling this opportunity; none of this would have been possible without them. Finally, I would like to dedicate this thesis to my mother who passed away before I was able to obtain my Master's degree.

Mihaly Katona
Delft, January 2024

Contents

List of Abbreviations

API	Application Programming Interface
ARIMA	Autoregression Integrated Moving Average
ARMA	AutoRegression Moving Average
CDF	Cumulative Distribution Function
CI	Confidence Interval
CRPS	Continuous Ranked Probability Score
LSTM	Long Short-Term Memory
MCMC	Markov Chain Monte Carlo
ML	Machine Learning
MLE	Maximum Likelihood Estimator
PDF	Probability Density Function
PIT	Probability Integral Transform
PPL	Probabilistic Programming Language
STD	Standard Deviation
SVI	Stochastic Variational Inference
TTD	Time To Departure

Introduction

Airports form the foundational infrastructure enabling the aviation industry, facilitating all necessary processes. A relatively recent pain point for both passengers and airports has been the security checkpoints. These checkpoints control access to secure areas by filtering out contraband items and restricting access to malicious actors. However, this process forms a significant bottleneck. Given that security accounts for up to a quarter of airport operational expenses and is widely disliked by passengers, improving it presents a significant incentive [33].

One of the simplest and most cost-efficient ways to improve this is by operational optimisation, typically in the form of resource allocation. This entails estimating the required number of open security lanes throughout the day. The optimal performance of these algorithms requires high-quality forecasts of the expected arrival rates at the checkpoint [63]. However, it has been found that current forecasting approaches are lacking.

This research has been done in collaboration with GRASP Innovations whose goal is to employ technology-driven data collection and aggregation to offer clear insights for optimising resource and infrastructure utilisation. A large amount of high-quality and resolution data was made available for this project. This enabled the development of a novel, bottom-up approach that predicts and combines individual flight's arrival rates. The main goal is developing and evaluating a real-time probabilistic passenger arrival forecasting model. This forecasting model allows for more powerful operational optimisation algorithms that can quantify risk and uncertainty. Providing decision-makers with a significantly more informative decision-support system.

This thesis report is organised into three main parts: In Part I presents the scientific paper that develops and evaluates the forecasting model. Part II contains the relevant Literature Study that supports the research.

I

Scientific Paper

Real-time Probabilistic Passenger Arrival Forecasting

Mihaly Katona *

Delft University of Technology, Delft, The Netherlands

Abstract

Through several contractions, stiff competition, and increasing passenger expectations, airports must evolve continually. One of the main avenues for this has been improving the efficiency of the security checkpoints, which are airports' primary bottlenecks. Operational optimisation methods, such as resource and task scheduling are relatively mature fields of research, however, they require accurate forecasts. Current forecasting approaches seldom use useful information such as the flight schedule, nor are they able to represent uncertainty or integrate real-time information. Therefore this paper aims to develop and evaluate a real-time probabilistic security checkpoint arrival rate forecasting model by utilising a Bayesian framework. This is achieved using a probabilistic programming language to create a bottom-up model, where per-flight arrivals are predicted. The passenger arrival rate for each flight is determined by estimating the total number of passengers and their temporal arrival distributions probabilistically. The combination of arrival rates from all flights in the flight schedule then provides the full checkpoint forecast. Furthermore, an updating scheme is proposed, that updates the expected number of passengers for each flight through Bayesian inference. Results show that the static forecasting model has promising performance, while successfully capturing uncertainty. However, the proposed real-time updating approach does not function as intended, due to a consistent negative bias. This has been attributed to a fundamental asymmetry present in the problem. Finally, this study includes an application of lane requirement estimation, which yielded highly favourable results. Allowing decision-makers to minimise costs while keeping the probability of poor checkpoint performance to acceptable levels.

Keywords: Airport security checkpoint, Forecasting, Passenger arrival rate, Bayesian framework, Probabilistic programming, Real-time updating

1 Introduction

The aviation industry as a whole has been a, if not the, defining industry of the modern era, facilitating globalisation and enabling massive economic growth. Just the airport service markets have seen a consistent yearly growth rate of 4%, and a total valuation of USD 159 billion in 2022, revealing the significance of the industry [6]. However, over its relatively short lifetime, there have been significant contractions caused by events such as the September 11th attacks, the 2008 financial crisis, and most recently the COVID-19 pandemic [5]. These have increased pressure on airports to improve their operational efficiency and reduce costs. As a result, significant focus has been centred on security checkpoints in academia and the industry. This is because they are the primary bottlenecks in the flow of passengers, having a high impact on passenger satisfaction and accounting for up to a quarter of all operational costs of airports [8].

There are two broad approaches to improving security checkpoint efficiency: infrastructure upgrades, and operational optimisation. Infrastructure upgrades require significant capital investments and downtime during implementation and therefore carry a high associated risk. But can result in significant increases in throughput at the same operational costs. Alternatively, operational optimisation offers a lower-risk approach that aims to maximise the utilisation of existing infrastructure, through better scheduling and resource allocation. In absolute terms, it provides for comparatively smaller efficiency gains. However, implementation costs are orders of magnitude less and therefore potentially offer superior cost-to-benefit ratios. A cornerstone of these techniques is knowledge of the future, for which forecasting is used [16].

Specifically, it has been found that forecasting passenger arrivals at the security checkpoint allows for optimal resource and task allocation to be carried out. There are 4 primary airport passenger throughput models; time series, causal, artificial intelligence, and hybrid models [4]. Time series models, such as the ARMA family of models, were found to be lacking due to them not being able to utilise flight schedules [9]. On the other hand, causal models offer a novel way of integrating flight schedule information by fitting a Time To Departure (TTD) distribution, which uses a parametric distribution for the probability of passengers arriving x minutes before departure of their flight [13] [14]. ML models such as LSTM and deep neural networks with autoencoders

*Msc Student, Air Transport and Operations, Faculty of Aerospace Engineering, Delft University of Technology

had negligible performance improvements over other approaches. While requiring orders of magnitude more data, computational resources, and model complexity [12] [10]. Finally, hybrid models typically use a time series model, such as ARIMA, with a subsequent ML model to capture non-linearities, which does improve performance but does not fix underlying issues of not being able to utilise flight schedules [15]. The existing literature has two significant gaps, uncertainty is not quantified even in a highly stochastic environment, and no approach investigated integrating real-time data. Given these gaps, Bayesian approaches potentially allow for uncertainty quantification and updating of the model using real-time data.

Therefore the following research aims *to develop and evaluate a real-time probabilistic security checkpoint arrival rate forecasting model by utilising a Bayesian framework.* To achieve this, the following steps have been taken. First comprehensive data filtering and analysis steps were performed, the goal of which was to ensure consistent data quality and identification of features of interest. Following this a static forecast is made by fitting two models, one model estimates the expected number of passengers for a given flight, while the other model represents the temporal distribution of the arrivals with respect to the departure time of the flight. Both of these are fitted using Pyro which is a probabilistic programming language [2]. The sampling of the combination of the aforementioned two models provides the expected arrival rate as a function of time to departure (TTD) for a single flight. And given the flight schedule, the individual arrival rates of all flights can be combined to get the checkpoint arrival rate. Then a real-time updating strategy is proposed to update the number of expected passengers for each flight. However, it has been found that fundamental asymmetry in the problem causes biased updates to take place.

The paper is structured as follows. Section 2 provides background on the tools necessary to apply a Bayesian framework. The methodology is then outlined in section 3 followed by a brief overview of the data in section 4. Then a detailed description of the static forecasting approach is presented in section 5, followed by an evaluation of this approach in section 6. Section 7 elaborates and discusses the proposed real-time updating approach. The model then is evaluated with a case study on estimating the required number of lanes to open in section 8. Limitations and implications are then discussed in section 9 and then a conclusion in section 10.

2 Background

In this work, forecasting will be carried out using Bayesian framework, which requires the utilisation of specific tools and approaches. First, a brief overview of why and how probabilistic programming can be used as a practical implementation of Bayesian approaches is presented in section 2.1. This is followed by an exploration of appropriate metrics for evaluating probabilistic density forecasts in section 2.2.

2.1 Bayesian Inference & Probabilistic Programming

Bayes theorem is one of the most fundamental equations in probability theory and forms the foundation for the Bayesian framework. It describes the probability of an event based on prior beliefs and the likelihood of observing new evidence given that the event occurs. This allows it to represent uncertainty, and to update the initial forecast with new information, providing for an elegant way to address the identified research gap. Which mathematically is given by Equation 1

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)} \tag{1}$$

Application of this equation to toy examples such as estimating the fairness of a coin, given some observations, is relatively trivial. However, as the complexity of the system increases, finding an analytical solution becomes impractical due to multiple and often non-linearly interacting variables. Not to mention the fact that the marginal distribution $P(B)$ is often intractable in practical settings. Probabilistic programming languages (PPL) provide a solution to these problems, by allowing for complex model definitions. Where variables are represented by arbitrary distributions. They use observations to update these distributions and automatically infer posterior distributions, that is perform Bayesian inference. When using a PPL to model a system three main steps are performed:

- **Model specification** - the relationships between model parameters and variables are declared. This takes the form of a bottom-up approach, where the mathematical relationships between variables are represented with respect to observations from the data. Furthermore, variables can be used for priors of other variables, which allows for Hierarchical structures.
- **Inference** - given some observations the parameters and variables of the model are iteratively adjusted such that the output of the model matches the observations. In general terms, the value of each observation is propagated up the model to inform the possible values of each variable, resulting in a distribution of possible values.

- **Parameter extraction and use** - once a probabilistic model is trained, relevant variables can be extracted with either point estimates or (non-)parametric distributions. Given the initial model structure, the variables then can be sampled from their respective distributions to produce new samples for the output of the model. This is useful, especially for contexts where decision-making under uncertainty is performed, since uncertainty is naturally captured in the output samples.

While a PPL is required for all three of the above-mentioned steps, PPL's main contribution is the implementation of a streamlined and efficient inference engine. There are two main approaches, Markov Chain Monte Carlo (MCMC) and Stochastic Variational Inference (SVI). In short MCMC approaches use a Markov Chain of sampled values for the model variables, where new elements are accepted based on a biased criteria that prefer more optimal choices. The values in the chain then converge to the posterior distribution of the variables over time. Although MCMC requires fewer assumptions and can represent more complex models, it is also computationally expensive and scales poorly with data and model size [7]. On the other hand, SVI utilises a simplified approximation of the original model. Turning the inference problem into an optimisation problem using gradients, which can be solved a lot more efficiently. This is done by introducing approximating distributions which are tractable, to represent the true distributions in the model [3]. This approach assumes that the approximating distributions sufficiently resemble the true underlying distributions. Since quite a large amount of data will be used, and its assumptions can be sufficiently satisfied, SVI will be used in this thesis. Particularly a version that requires that underlying distributions can roughly be represented by a Normal distribution, the validity of which will be discussed in later sections.

For the following research, Pyro, a python PPL library, has been utilised for two significant reasons [2]. Firstly and foremost Pyro was built from the ground up to process large amounts of data efficiently by focusing its development on robust SVI capabilities. Secondly being a python package with a focus on providing a "pythonic" interface, facilitates streamlined model definition and integration. Thereby reducing development overhead and potential technical complications.

2.2 Density Forecast Evaluation

Finally, the utilisation of the Bayesian framework necessitates evaluation metrics specifically designed to assess the quality of density forecasts. These metrics should capture both the calibration and sharpness of the distribution, given observations. Calibration refers to the distribution of the observations, that is, if the proportions of the frequencies of events are correctly distributed in the prediction. A well-calibrated model will predict an event to occur 50% of the time if the chance of occurring is 50%. On the other hand, the sharpness gives the "resolution" of the model, describing how narrow the predicted densities are [11]. Consider a well-calibrated weather model, predicting a 30% chance of rain accurately for every single day; but this is not very informative. The Continuous Ranked Probability Score (CRPS) captures both of these into a single value. This is done by integrating the squared difference between the cumulative distribution function (CDF) and the observed values. Lower scores indicate better performance, with the minimum score occurring when the observation aligns with the mean of the predicted distribution. The CRPS scoring function is given by Equation 2

$$CRPS(F, y) = \int_{-\infty}^{\infty} (F(x) - H(y - x))^2 dx \tag{2}$$

Where y is the observation, F is the CDF of the forecast, and H is the Heaviside step function (that evaluates to 0 if the value is negative and 1 otherwise). Although CRPS effectively captures both calibration and sharpness, it is primarily used for comparative model evaluation, since the absolute value isn't very informative as it depends on the scale of the data. To overcome this several additional common or ad hoc metrics will be discussed and used where necessary throughout this paper.

3 Methodology

The following research aims to answer the question of "How can uncertainty in passenger arrival rate forecast be captured and quantified, and then updated in the presence of real-time information?". A Bayesian static forecasting tool, followed by a real-time adjustment component has been developed. The outline of the steps taken is given in section 3.1, with subsequent sections providing more detailed insights into each component. Verification and validation steps are briefly outlined in section 3.2.

3.1 Methodology Overview

Firstly, due to the data-driven nature of this work, initial data formatting, cleaning, and prepossessing has been carried out before model development. The primary data is timestamped boarding card reader data, which

registers the time of each passenger entering the security checkpoint and the flight they will be boarding. This has been interpreted from a time-to-event perspective, measuring the time to departure (TTD) (in minutes) of each passenger's arrival. Additional features for each flight, such as; airline, aircraft capacity, destination, etc., have been collected through the Aerodatabox API [1]. Finally, the original boarding card data contains periods of missing data. To overcome this a simple algorithm has been developed to identify affected flights, reducing the amount of data that had to be discarded. Which then was used to train the forecasting models.

The static forecasting model breaks down the forecasting problem into two prediction problems. For each flight in the flight schedule, the number of passengers is estimated, along with the time to departure (TTD) distribution. The TTD distribution represents the temporal distribution of the passengers arriving to a flight. The combination of this provides the absolute arrival rate as a function of time for each flight. This then is combined according to the temporal information from the flight schedule to get the arrival rate for the whole checkpoint. First, the TTD distribution model has been developed. Here a parametric distribution is fitted using a Probabilistic Programming Language (PPL), this allows for the parameters of the distributions to be represented by distributions themselves. This results in a distribution over distributions, that is probabilistic output. For each time period before departure, instead of a single value, a distribution of values is returned for the expected fraction of arrivals. The principal concern in this model's analysis was the identification of the parametric distribution that most accurately characterised the TTD data.

The second model estimates the number of passengers of each flight. Firstly the number of people arriving for each flight is converted into a load factor, which is the number of passengers divided by the capacity of the aircraft. This simplifies the prediction problem to be in the range of $[0, 1]$, enabling several advantageous simplifications that will be discussed later. The model is then trained to return a distribution to predict the likely number of passengers. This model is primarily evaluated on capturing long-term seasonal trends. Combining this model and the TTD distribution, a distribution of arrival rates is produced for each flight. This then allows for estimating the full checkpoint arrival rate using the flight schedule, resulting in a density forecast. Validation of this final static forecast has been evaluated against the available time series representation of the boarding card data.

Finally given the validated static forecasting model, a real-time updating algorithm is proposed. This is based on the observation that the correct prediction of the passenger count has significantly more impact on the forecast result than the TTD distribution. Furthermore, the count distribution is represented by a single distribution, as opposed to distribution over distributions, and therefore easier to update. The updating schema proposed uses the difference between the observed arrivals, and the expected distribution of arrivals for each flight. This is used to perform Bayesian inference updates on the count distribution. However, it has been found that informational asymmetry between observing a large number of passengers and a small number of passengers causes a bias in the proposed method. Resulting in a negatively biased adjustment of the static forecast. This finding is analysed in detail, and modelling decisions and assumptions are discussed.

Lastly, a brief example application of the forecasting algorithm is implemented. Specifically, a capacity planning problem where the number of required lanes is probabilistically evaluated. In this case study, lanes are assumed to have uncertain throughput. For each time bucket, the distribution of the possible number of arrivals is compared to this throughput. Returning the percentage chance that a given number of lanes will be sufficient to meet demand. This is then validated against the same process with the actual arrival rates.

3.2 Verification and Validation

In the context of forecasting, verification ensures that the model was built correctly and operates as intended, while validation focuses on ensuring that the model accurately represents the forecasted phenomena. Verification was performed following software engineering standards, such as unit and integration testing. Additional verification was carried out by extensive visualisation of intermediate and final values produced by the model. Validation has been primarily been carried out through the use of the CRPS score to evaluate the performance of the final model, which is further explored in detail in section 6 for the static forecasting model, and section 7 for the real-time component. Additional validation, in the form of the case study, has been performed, evaluating the forecasts' ability to be used for lane requirement estimation. Before delving into the models and their evaluations, the required data must be explored and formatted.

4 Data Overview and Processing

Data had been made available through collaboration with GRASP Innovations and the large European airport. Primarily boarding card data, collected at the entrance to the security checkpoint queue, which contains a timestamp of the scanning, as well as the flight number associated with the boarding card. The following sections will first provide a detailed overview of the available data in section 4.1, and then the formatting and handling of missing data is discussed in section 4.2.

4.1 Data overview

The available, and useful data spans from Jun 1 to Nov 7, containing **4,272,695** passengers over **39,837** flights. This boarding card reader data has been supplemented by the aerodatabox API, which provides additional information such as the destination, the aircraft and the capacity [1]. With this additional data Table 1 provides the main features available. It is worth noting that additional features can be extracted from the ones presented below, such as the day of the week from the flight departure time, or distance to the destination from the destination.

Feature	Data type	Comments
Scanning time	Timestamp	-
Flight departure	Timestamp	-
Flight ID	Categorical	808 flight ID's and 105 airlines
Aircraft capacity	Numerical	-
Destination	Categorical	200 destinations in 67 countries

Table 1: Available raw data description

The TTD (time to departure) data is measured in minutes and has been assigned negative values for passengers arriving before departure, with departure being at 0 minutes. This was done to allow for more intuitive visualisation since both time and the numerical values of the x-axis are expected to progress from left to right as demonstrated in Figure 1. The data contains a wide range of arrival times, both significantly before the flight's departure and after. For the TTD model, parametric distribution are used to represent each flight's TTD pattern. Since the fitting is sensitive to outliers, the range of TTD values needs to be determined, balancing between the removal of outliers, without discarding too much data. Table 2 show the percentage of data that is excluded for each cutoff time, additionally **0.18%** of passengers arrived after departure.

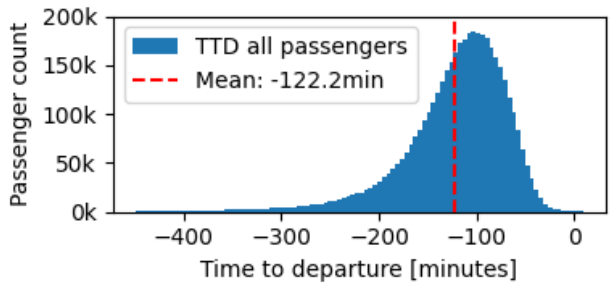


Figure 1: Combined TTD distribution of all passengers

TTD cutoff [min]	Count	Percentage [%]
-450	8411	0.20%
-400	17087	0.40%
-350	33945	0.79%
-300	66148	1.55%
-250	139694	3.27%

Table 2: Data influenced by TTD cutoff choices

Given some initial fitting tests, and discussions with industry experts a suitable TTD range of $[-300, 0]$ minutes was chosen, which misses **1.73%** of passengers. This was deemed to be acceptable, both because of the relatively low % of passengers excluded, as well as the fact that passengers typically are requested to arrive 2-3 hours before departure. Finally, flights with less than 10 passengers have also been discarded, as they constitute 9.84% of flights but only 0.29% of total passengers. This can potentially significantly skew model behaviour while representing a minuscule proportion of arrivals. Therefore overall **2.02%** of passengers has been excluded.

4.2 Data Processing

The quality of the available data is relatively high overall, featuring standardised fields and consistent formatting. However occasional technical issues related to the saving of the data have left time windows with no available data. Since it is unknown how many passengers, and to which flights they belong, flights that are likely to be affected by these data "anomalies" cannot be used for training of the models. This would both negatively influence the TTD model, as well as bias the count distribution to predict a lower number of arrivals. A naive approach could discard all data for days affected by anomalies, however, this would lead to removing 87 days out of 158 or roughly 55% of the data. To recover more data a nuanced approach was required, necessitating first the detection of these anomalies, and then identification of affected flights. Fortunately, it has been observed that this issue occurs consistently, with data both stopping and starting again at exact 5-minute intervals, simplifying detection. Additionally as established, effectively all passengers arrive within a TTD range of $[-300, 0]$ minutes, meaning that an anomaly will only affect flights departing from the time of the anomaly to 300 minutes in the future. A time series representation of the data bucketed at 5 minutes is shown in Figure 2, where the red squares show identified anomalies, and the grey area shows the time in which affected flights are located.

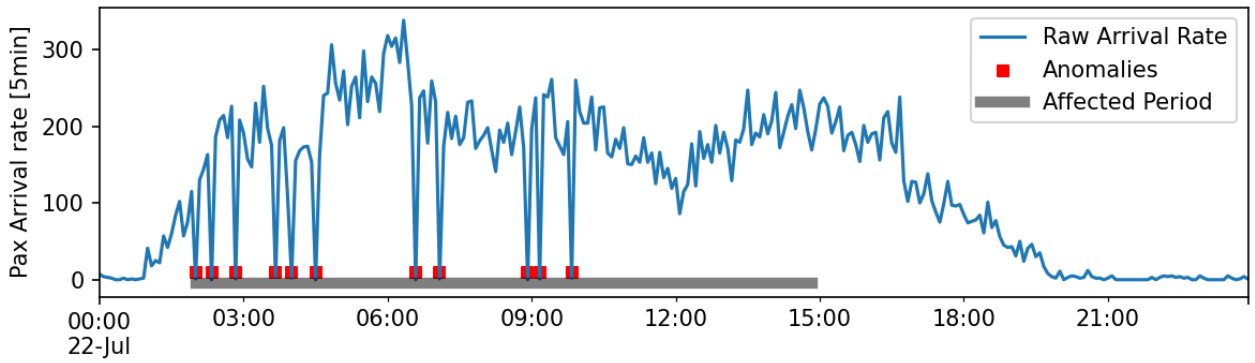


Figure 2: 5 minutes bucketed time series boarding card data, showing the identified data anomalies (red rectangles), and affected time range (grey area)

Given the range of the affected time periods, it is possible to exclude all flights that fall in this range only keeping flights that were not affected by the anomalies. This can be seen in Figure 3, where the included flights are green diamonds, and the resulting filtered arrival rate for these flights is indicated by the dashed line.

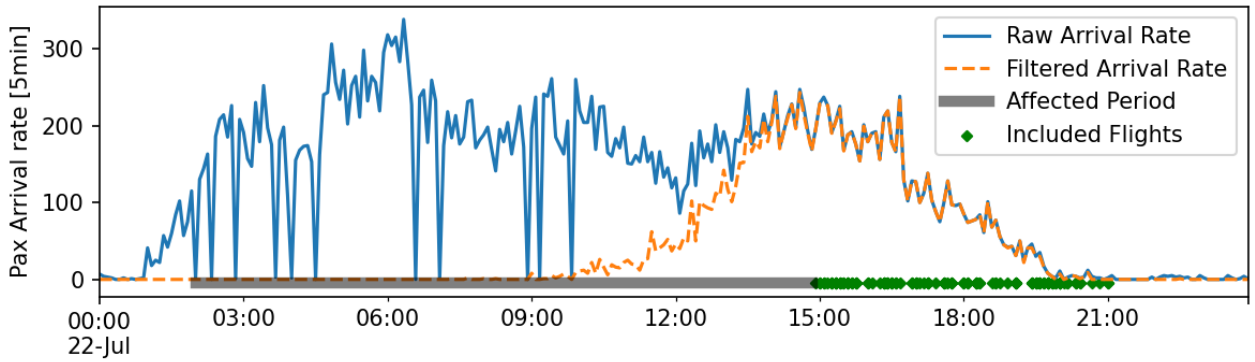


Figure 3: 5 minutes bucketed time series boarding card data, showing the flights and time series data kept

Finally, it is possible to slightly relax the criteria with which flights are excluded. For example, the number of arrivals in the TTD range of $[-300, -250]$ is usually very low, making the impact of missing 5 minutes of data negligible. Additionally even for the most populous flights missing a single 5-minute window of arrivals should on average only miss a couple of arrivals. Therefore for the final filtering, anomalies are considered to only affect flights in the range $[-250, -10]$, and a single anomaly is allowed within this range. Further detail and the trade-off performed can be found in Appendix A. The result of this is that only 23% of the data needs to be discarded, meaning that there is an increase of 71% in data available for training compared to a naive discarding approach. This data will be used by both the static forecasting model, discussed next, and the real-time updating model.

5 Static Forecasting Model

The following section elaborates on the models used to solve the static component of the forecasting problem. Focusing on predicting the arrival rate for individual flights. First modelling the temporal distribution of arrivals relative to departure time, the TTD model in section 5.1 and passenger count predictions, the count model in section 5.2. The section concludes their combination into the overall arrival forecast at security checkpoints and additional considerations (section 5.3)

5.1 TTD Distribution Model

The first step in building up the checkpoint forecast is the estimation of the TTD distribution for each flight. This is done by taking a flight's features as inputs and returning the temporal distribution of passenger arrivals. Several features have been identified that impact passenger arrival times; time of day, airline, destination, or the distance to it. Conveniently these are all captured in the flight ID, since a flight ID represents a given airline

flying to a destination at a consistent time of day. This observation allows for a significantly simplified model, that requires fewer input features, while still implicitly representing a large number of useful characteristics. With this simplification, a hierarchical Bayesian regression model was chosen, using the airline and then the flight ID. Here airline-level information informs priors for individual flight IDs. This is especially beneficial for flight IDs that have a low number of flights, and would otherwise have large uncertainties. To implement this, a parametric distribution has been fitted to each flight's TTD rate using a MLE (maximum likelihood estimator) with the python package scipy. For purposes of explanation, the normal distribution will be used as an illustrative example. However, any parametric distribution can be used, which is explored in section 6.1.

Since large amounts of data will be used, SVI is preferred over the less efficient MCMC algorithm [2]. This means that model parameters need to be reasonably well represented by the approximating distributions used by the SVI algorithm. Normal distributions not only allow for the most optimal computational efficiency and convergence in SVI but they have been found to sufficiently approximate the available parameters. The Shapiro-Wilk test has been used for its reliable performance with small sample sizes, with a standard 0.05 significance. From this, it was found that **71%** of means and **74%** of standard deviations can be assumed to come from a normally distributed population. Since a significant proportion meets this requirement normal distributions are deemed to be representative enough. Additional assumptions are discussed in Appendix B. Given the above finding, the individual parameters for each flight are normalised to have a global mean of 0 and variance of 1, which is necessary for improved model convergence.

The hierarchical model is trained with the parameters from the fitted parametric distributions of flight's TTD patterns. In this case, the mean and variance, represented by the left side of Equation 3a. As discussed above it is assumed that the flight ID captures the behaviour of each flight. Therefore the observations are modelled by a Normal distribution, parameterised by μ_i and σ_i, which represent the mean and variance of the associated flight ID i. These flight ID parameters are themselves modelled by Normal distributions as given in Equation 3b and Equation 3c. Which have priors from airline a, that map onto all associated flight IDs i given by the function FlightID. Finally, the hyperpriors from the airlines are given by Equation 3d - 3g. The superscripts of these variables are used to help differentiate them and are used as an extra "level" of indexing. Where the means are sampled from $\mathcal{N}(0,1)$ and the standard deviations from HalfNormal(1) for all airlines a. It is worth highlighting that Equation 3c, which is the distribution of standard deviations for each flight ID i, can be represented by a Gaussian (which can be negative) because of the normalisation.

$$X_{flight} \sim \mathcal{N}(\mu_i, \sigma_i) \tag{3a}$$

$$\mu_i \sim \mathcal{N}(\mu_a^{(\mu)}, \sigma_a^{(\mu)}) \quad \forall \text{i} \in \text{FlightID}(a) \quad \forall \text{a} \in \{1, ..., n_{\text{airline}}\} \tag{3b}$$

$$\sigma_i \sim \mathcal{N}(\mu_a^{(\sigma)}, \sigma_a^{(\sigma)}) \quad \forall \text{i} \in \text{FlightID}(a) \quad \forall \text{a} \in \{1, ..., n_{\text{airline}}\} \tag{3c}$$

$$\mu_a^{(\mu)} \sim \mathcal{N}(0,1) \quad \forall \text{a} \in \{1, ..., n_{\text{airline}}\} \tag{3d}$$

$$\mu_a^{(\sigma)} \sim \mathcal{N}(0,1) \quad \forall \text{a} \in \{1, ..., n_{\text{airline}}\} \tag{3e}$$

$$\sigma_a^{(\mu)} \sim \text{HalfNormal}(1) \quad \forall \text{a} \in \{1, ..., n_{\text{airline}}\} \tag{3f}$$

$$\sigma_a^{(\sigma)} \sim \text{HalfNormal}(1) \quad \forall \text{a} \in \{1, ..., n_{\text{airline}}\} \tag{3g}$$

After training the above model, all of the above variables will have an associated parameterised Normal distribution. However, only μ_i and σ_i are saved resulting in a total of 4 parameters for each flight ID, since these are Normal distributions themselves with 2 parameters each. These are denormalized before saving to revert them to their original scale. By sampling from the distributions of means and variances, we generate multiple normal distributions. Each represents the potential range of arrival times for a flight, effectively capturing the variability in arrival rates. Producing distribution over distributions which can be seen in Figure 4.

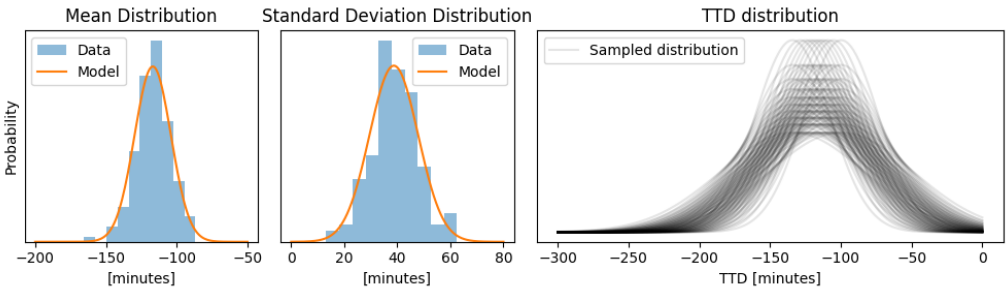


Figure 4: An example flight's TTD distribution modelled by a Normal distribution, with the fitted distribution and actual data of the mean [left], and standard deviation [centre], and the combined TTD distribution [right]

This method captures two types of uncertainty: the natural variation in arrival rates between flights and the model's predictive confidence, which improves with more data per flight ID. With a robust temporal distribution model established, the next step involves forecasting the number of arrivals per flight.

5.2 Count Distribution Model

In this next model, the likely number of passengers arriving to each flight is estimated through a probability distribution. First, each flight's number of arrivals is converted into a load factor, representing the proportion of capacity used, constrained between $[0, 1]$. With 0 representing no passengers arriving, and 1 indicating that the maximum number of passengers has shown up to the flight. Literature reviews and data analysis indicate three primary factors influencing the number of arrivals: flight characteristics, and short-term and long-term seasonal effects. As already discussed in section 5.1 when grouping by flight ID, it implicitly contains a large number of flight characteristics in a single variable. As for the short-term seasonal effects, it has been found that there are significant, unique, and repeating patterns present depending on the day of the week. Finally, since there is not enough data to represent long-term seasonal effects, a reasonable alternative has been found. Using the running average of the load factor as a lagging indicator for long-term effects. This results in a model that has three input features, the flight ID, the day of the week, and the rolling average of the load factor.

Given these features, a desired model output distribution needs to be selected. Since the load factor of a flight is partially influenced by temporal components, data cannot be simply aggregated per flight ID, and therefore it's difficult to statistically prove that any one distribution represents the data. Therefore a hypothesis has been made that the beta distribution will be a good fit for this problem, primarily driven by the following beneficial properties:

- **Probability density constrained on 0-1** - This allows a direct representation of the load factor and ensures that the two most significant constraints are respected. Namely, the number of arrivals must be greater or equal to zero, and at most the capacity of the aircraft.
- **High representational range** - The beta distribution can take on a uniform distribution, close to a normal distribution, and heavily skewed distributions near the boundary values of $[0, 1]$. This should allow it to capture most unimodal underlying data distributions.
- **It is a conjugate prior** - Meaning that a Beta distribution can trivially be updated by another Beta distribution, resulting in a posterior that is also a Beta distribution. This is useful for both modelling and updating the distribution in real-time which is expanded on in section 7.2.

Each flight's load factor observations are modelled as a distribution given by Equation 4a. This beta distribution is parameterised by α_c and β_c which are defined by equations Equation 4b and Equation 4c respectively. These equations represent the combined influence of all three input features. The first component, parameters $\alpha_i^{(\text{flight ID})}$ and $\beta_i^{(\text{flight ID})}$, represents the characteristics of each flight ID i and their relation to the load factor's lagging indicator lf_i, with priors detailed in Equation 4d and Equation 4e. This captures the variance of each flight around its moving average. The second component of the equations quantifies the short-term seasonal effects for each day of the week, given by $\alpha_{i,d}^{(weekday)}$ and $\beta_{i,d}^{(weekday)}$. The variables are sampled for each flight ID i, and each weekday d with Equation 4f - 4g and represent the offset from the load factor.

$$X_{\text{lf}} \sim \text{Beta}(\alpha_c, \beta_c) \tag{4a}$$

$$\alpha_c = \alpha_i^{(\text{flight ID})} \times lf_i + \alpha_{i,d}^{(weekday)} \tag{4b}$$

$$\beta_c = \beta_i^{(\text{flight ID})} \times (1 - lf_i) + \beta_{i,d}^{(weekday)} \tag{4c}$$

$$\alpha_i^{(\text{flight ID})} \sim \text{HalfNormal}(1) \quad \forall \text{i} \in \{1, ..., n_{\text{flight ID}}\} \tag{4d}$$

$$\beta_i^{(\text{flight ID})} \sim \text{HalfNormal}(1) \quad \forall \text{i} \in \{1, ..., n_{\text{flight ID}}\} \tag{4e}$$

$$\alpha_{i,d}^{(weekday)} \sim \text{HalfNormal}(1) \quad \forall \text{i} \in \{1, ..., n_{\text{flight ID}}\} \quad \forall \text{d} \in \{1, 2, ..., 7\} \tag{4f}$$

$$\beta_{i,d}^{(weekday)} \sim \text{HalfNormal}(1) \quad \forall \text{i} \in \{1, ..., n_{\text{flight ID}}\} \quad \forall \text{d} \in \{1, 2, ..., 7\} \tag{4g}$$

$$\tag{4h}$$

Post-training, the model saves the α and β parameters corresponding to each flight ID and weekday. This allows for the the final α and β to be calculated for a flight given the moving average value of the load factor, and the day of the week. This is illustrated in Figure 5, where on the right plot the predicted distribution is given by the box plot, and the influence of the day of the week and running average can be seen. This load factor distribution is multiplied by the aircraft's capacity to yield the expected passenger count distribution.

Combined with the TTD model, this approach returns individual flight arrival rates, paving the way for the next section on combining these into a full forecast.

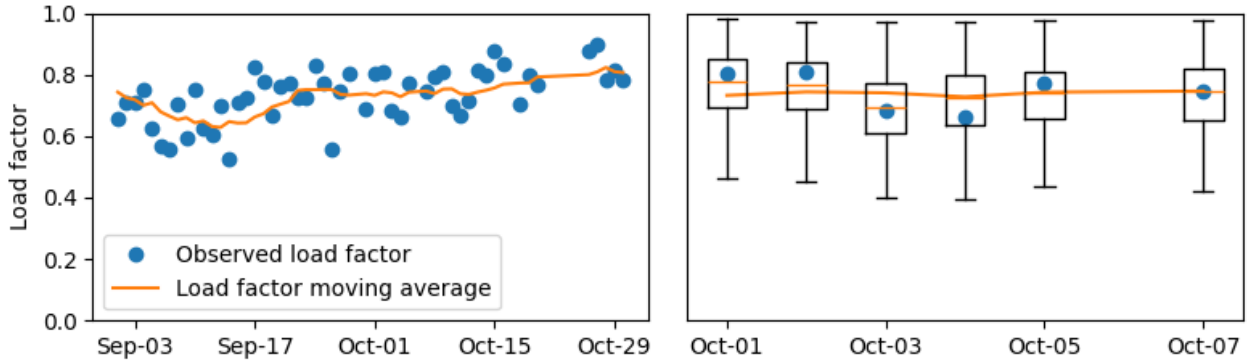


Figure 5: An example flight's load factor evolution with time [left], and a subset of times for which the load factor has been predicted [right]

5.3 Combined Forecast & Considerations

Once parameters for both the TTD distribution and the count distribution are fitted, a full forecast can be evaluated given the flight schedule. This process is sequentially applied to every global time t, each associated with a constant bucket length b. For each t, samples are drawn from both of the fitted distributions of each flight and multiplied together, and then combined. More specifically there are four distinct steps required to generate the forecast for the expected number of arrivals at time t with bucket length b:

- **TTD distribution is sampled for each flight** - For global time t the time to departure t_{TTD} is calculated using the flight's departure time, and n pairs of means and std's are drawn from the TTD distribution represented by Equation 3b and Equation 3c. For each pair of parameters in n, the probability distribution function (PDF) is numerically evaluated from t_{TTD} to $t_{TTD}+b$, yielding n samples per flight, representing a distribution of arrival fractions.
- **Count distribution is sampled for each flight** - For each flight in the flight schedule the additional inputs of the day of the week, and the running average of recent load factors for a flight ID are collected. Using these the alpha and beta parameters are calculated using Equation 4b and Equation 4c. The resulting beta distribution is subsequently sampled n times randomly. Then multiplied by the aircraft capacity, resulting in samples for the number of arrivals to a given flight.
- **Samples from TTD and count distributions are combined for each flight** - The mathematical combination of these two distributions can be seen as a product of distributions. Where practically the n samples from the TTD distribution, representing the fraction of arrivals, are multiplied element-wise by the n samples from the count distribution, representing the total arrival count. The resulting n samples give a distribution for the number of passengers expected to arrive at time t, over a bucket of length of b for a given flight.
- **Combination of arrival count from all flights** - The final step involves aggregating the distributions from each flight through convolution. Here the n samples are element-wise summed together from all flights, resulting in a final distribution of samples that describes the expected number of passengers arriving at the checkpoint at time t over a bucket length of b.

An example full forecast is shown in Figure 6. Since many samples are used for this, confidence intervals (CI) can be used to simplify visualisation and provide uncertainty quantification. Where, for example, the 50% confidence interval contains 50% of the samples centred around the mean.

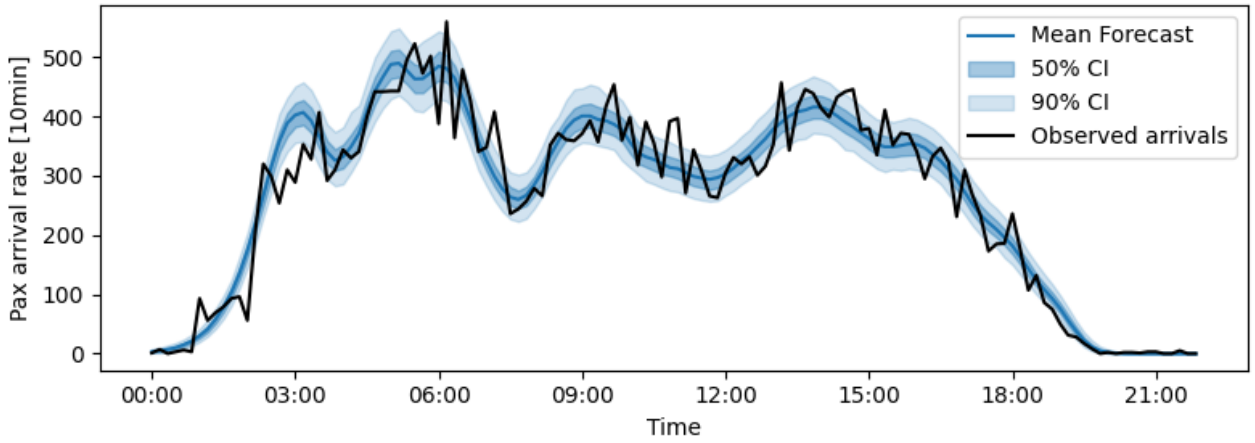


Figure 6: Forecasted and actual arrivals for the full day of October 10th

Finally, one additional step has a significant impact on the confidence intervals of the final forecast and therefore needs to be elaborated. The output of the above process may be better characterised as a distribution of the average arrival rates for a given time bucket. To refine the forecast, the nature of passenger arrivals must be addressed, which are discrete events occurring in continuous time. Thus, the next step is to convert the continuous distribution of average arrival rates into a discrete distribution. This has been achieved by using the Poisson process. Which takes the average rate of passenger arrivals and transforms it into a probability distribution of discrete numbers of arriving passengers. This approach not only transforms the continuous output into discrete values but also captures uncertainty originating from variable bucket lengths. Mathematically, the Poisson distribution's variance equals its mean, making its standard deviation the square root of the mean. As a result, larger average arrival rates yield proportionally narrower confidence intervals. An intuitive explanation for this is that for smaller buckets the inherent variability of individual arrivals is relatively large compared to the absolute value of the number of arrivals. Conversely, everything being equal, larger buckets typically capture more arrivals, reducing relative variability. Further considerations are discussed in Appendix C in addition to further elaboration of the use of the Poisson process.

6 Static Forecasting Model Evaluation

This section outlines the validation experiments and results for the static forecasting model, performed in a step-by-step manner. The evaluation consists of two main phases: first, a detailed examination will be carried out on the two "sub-models" of the TTD and count distribution models in section 6.1 and section 6.2 respectively. Then the section concludes by evaluating the full checkpoint forecast in section 6.3.

6.1 TTD Distribution Model Evaluation

As expanded in section 5.1 a distribution is fitted to the TTD data of each unique flight. Through the hierarchical Bayesian model a distribution over distributions is fitted to the temporal arrival pattern of passengers for each flight ID. The methodology was explained through the use of a Gaussian distribution, which has 2 parameters; mean and variance. However, any parametric distribution can be used, as the hierarchical model can fit an arbitrary number of parameters. Therefore the goal of this evaluation is to identify the parametric distribution that is best able to capture the distribution of the TTD arrival data.

The primary difficulty for this evaluation is the sparsity of the TTD data. With most flights having ~ 100 passengers arriving over a range of 300 minutes, even a relatively low number of passengers can significantly change the parameters of the fitted TTD distribution. This means that directly evaluating the goodness of fit per flight won't result in conclusive results. Therefore evaluations will be performed by fitting each of the distributions using the TTD model. This returns a distribution over distributions, which will be evaluated by the CRPS metric. Two experiments will be used for this:

- **Evaluating individual flight TTD pattern** - For each flight the observed TTD data will be bucketed and then normalised into a probability distribution. The TTD distribution will then be sampled with the same bucket size for each flight. For each time point this results in a distribution of expected outcomes from the TTD model, and observations for the fraction of arrivals for the corresponding bucket. These pairs will be evaluated using CRPS and averaged over all buckets for each flight. The left plot in Figure 7

shows the individual flight data for all flights belonging to a flight ID. This experiment will evaluate if the distribution used can capture the TTD arrival rate for each unique flight, testing how well the sparsity and randomness of individual flights are captured.

- **Evaluating aggregated flight ID TTD pattern** - The procedure for this test is the same, however, the TTD data is aggregated per flight ID, thereby combining multiple flights. This is visualised by the aggregation of the data in the left plot in Figure 7, resulting in the right plot. The goal of this experiment is to evaluate the goodness of fit of each distribution to the aggregated and smoothed-out TTD data. This can be seen as evaluating if the distribution can capture the underlying distribution.

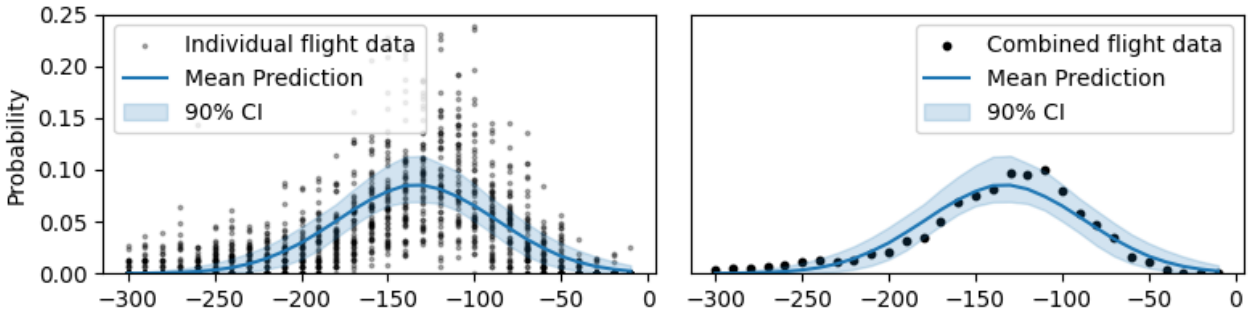


Figure 7: Example TTD evaluation (using a normal distribution) for a flight ID bucketed at 10-minute intervals, with data for the 54 individual flights [left], and the combined data [right]

6.1.1 TTD Distribution Model Results

The following experiments were carried out using TTD models that have been trained on all the available data. A total of 4 parametric distributions have been evaluated, 2 of which have been already used in literature, the Normal and Weibull distributions. Additionally, the Skew Normal distribution has been added since it is a more expressive Normal distribution. From preliminary testing and visualisations, an ad hoc adjustment of the Skew Normal is also evaluated. The Fixed Skew Normal, where the skew parameter is fitted on the flight ID level aggregated data. Finally, the evaluation was carried out at a 10-minute bucket size.

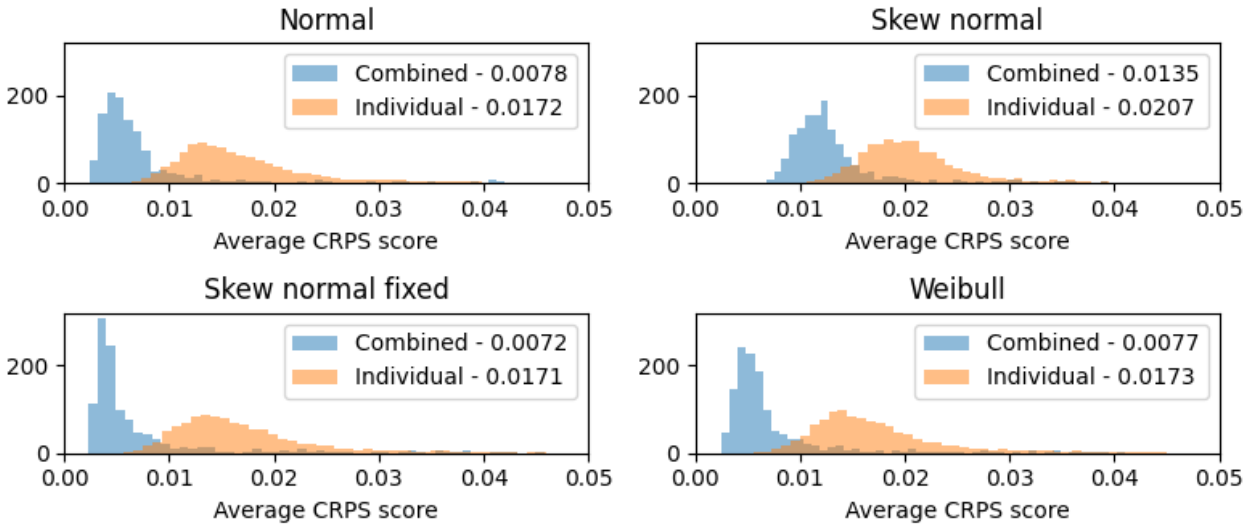


Figure 8: CRPS distribution for different TTD distributions, evaluating goodness of fit for flights combined per flight ID, and per individual flight

Figure 8 shows the distribution of CRPS values for the combined flight ID level, and the individual flight level for each of the four evaluated distributions. Firstly a clear difference can be seen between the combined and individual scores for all four. This is due to the "smoothing" effect of the data aggregation, which allows for better representation by a parametric distribution. As for differences between the distributions, the main

observation is that the normal, Weibull, and Fixed Skew Normal perform nearly identically. With the skew-normal distribution being the only outlier. In short, this is caused by the fact that the actual mean and variance of the distribution non-linearly depends on the skew parameter, this is expanded upon in section D.1. None of the remaining three distributions have statistically different scores and therefore will be further evaluated through the combined forecast in section 6.3.

6.2 Count Distribution Model Evaluation

The count distribution model uses high-frequency and long-term seasonal components to forecast a beta distribution for the load factor, as discussed in section 5.2. The day of the week provides for the high-frequency component, capturing the fluctuations due to differing travel patterns on specific days. However, due to having only having 6 months of data, yearly seasonal effects cannot be directly similarly predicted. Therefore a lagging indicator has been used for the long-term seasonal components. To effectively utilise the lagging indicator, a thorough analysis is needed to determine the most effective configuration. Two different smoothing methods have been considered: moving average, and exponential smoothing. Which were chosen for their proven efficacy and simplicity. These methods are compared via a sensitivity analysis to identify the optimal parameter values for each. The best-performing method and parameter value combination were chosen for use in the final forecasting evaluation. To achieve this, these experiments also use the CRPS to compare the difference between the predicted distribution of load factors, and the associated observation. For this evaluation, the data has been split up into a training set, that uses data till October 31 and contains 34122 flights. And a test set that uses the last 6 days of data available in November and contains 1680 flights. An example of the test set, for a single flight ID can be seen in Figure 9.

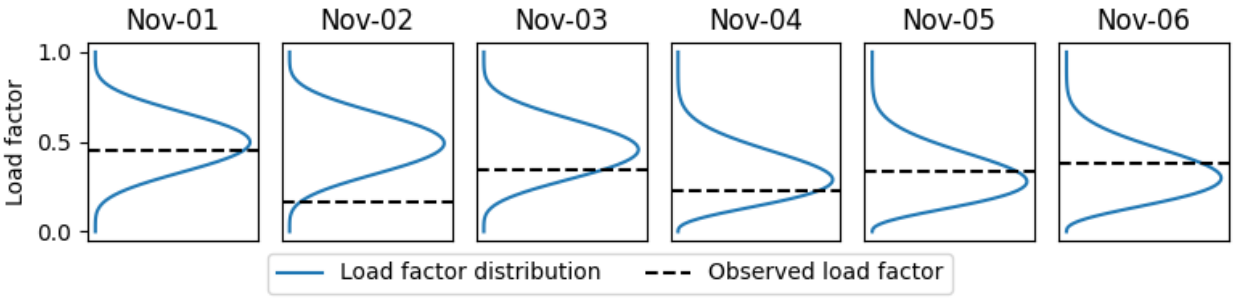


Figure 9: Example count distribution evaluation, showing the predicted load factor distribution vs the observed load factor for a flight over 6 days

Additionally, the aggregated performance of the count distribution model was evaluated, by comparing the total number of predicted passengers for a day, against the actual number of arrivals. This evaluation requires a broader time frame, that captures a large number of days. However, a challenge encountered was data quality, particularly the missing data detailed in section 4.2. The majority of days from the start of the data set till around the middle of September contain at least a few periods of missing data, with the remaining days having sporadic issues. Moreover, during the last 10 days of the dataset, an unusually high number of new and previously unobserved flights were introduced. Visualisations and further discussion of this issue are presented in section D.2. Due to these factors, the decision was made not to split the data into test and validation sets and to evaluate the model's performance using October's data. The consequences of these decisions are elaborated on in section 9.2.

6.2.1 Count Distribution Model Results

First, the smoothing methods have been evaluated with the results shown in Figure 10. The performance of each model and its parameters were evaluated 10 times to account for the stochastic fitting of parameters. The figure shows that the best option is the moving average with a 7-day window size. It is hypothesised that the optimal window size of 7 days is not by coincidence. Flights that fly more frequently have a higher representation in the data, with flights that fly daily having the most flights. A window size of a week likely captures the behaviour of these frequent flights well, and as they are highly represented, influencing the scores significantly. Additionally, exponential smoothing likely underperformed due to interference with the model's day-of-the-week component, as it disproportionately weights more recent days.

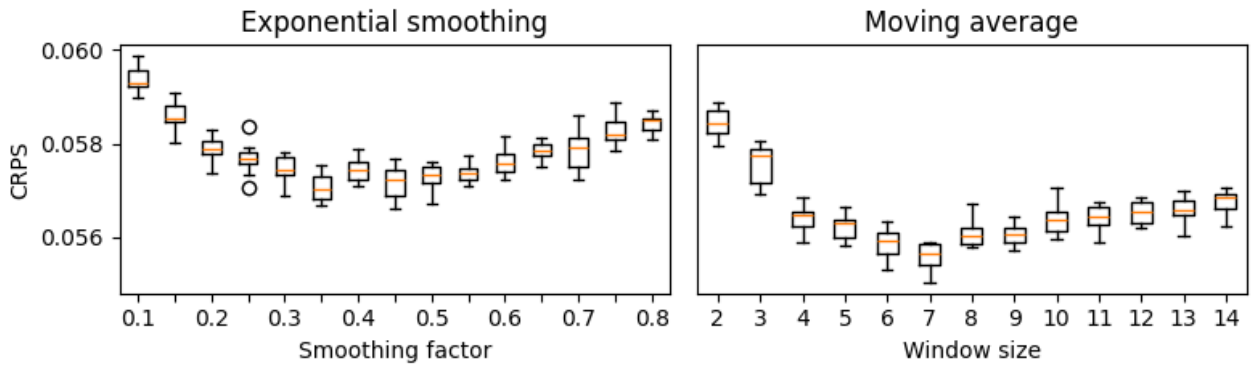


Figure 10: CRPS sensitivity analysis on time series smoothing techniques: exponential smoothing [left] and moving average [right] for estimating the load factor of flights

Having established that a 7-day moving average optimally captures load factors, the next experiment compares the predicted number of passengers per day with the actual arrivals. Specifically, only passengers that arrived to predicted flights are used, since flights with < 10 passengers are excluded, as discussed in section 4.2. The outcome of this evaluation is presented in Figure 11. The performance of the model is good, having on average an error of -1.3%, corresponding to estimating 552 passengers less per day with a standard deviation of 1132. While the mean performance is good, it has quite a large variance, indicating inconsistent performance over different days. The exact reason for this has not been identified, but it offers an interesting avenue for future research. Given this and the TTD model analysis, the final evaluation of the full checkpoint forecast can be carried out in the next section.

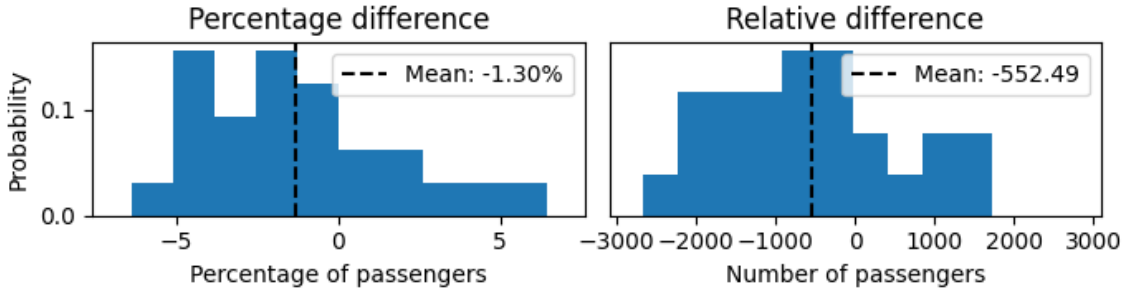


Figure 11: Daily passenger count prediction performance for October, showing the percentage difference between the mean of the prediction and the arrivals [left], and the relative difference [right]

6.3 Static Forecast Model Evaluation

As previously noted in section 6.1, there was no statistically significant difference among the three well-performing distributions. On the other hand for the count distribution, a moving average with a window size of 7 has been found to perform best. Similarly to section 6.2 due to issues with the data these evaluations were confined to the month of October. Additionally each day only the period between 2:00 - 18:00 will be analysed. Since this period has the highest activity, and outside this range, the arrival rate is often near or to 0 and would skew the results too much. First, the different distributions were analysed and the best performing one was selected. The CRPS metric will primarily serve to compare the distributions against each other. For each time bucket, the distribution of forecasted arrival rates and the observed arrivals are evaluated and then averaged. Additionally, more traditional metrics used such as MAE and RSME will be used to provide better known error metrics. For these metrics the observation is compared against the mean of the density forecast. Lastly R^2 have also been calculated to gain insight on the goodness of fit of the model. This will be followed by an evaluation of the total daily forecast, vs the actual arrival count. Which can be seen as an extension of the same analysis performed on the count distribution model. And will be used to assess the impact of various assumptions made for the full forecast. With all subsequent tests using a 10-minute bucket size.

6.3.1 Static Forecast Model Results

First, the evaluation of the three candidate distributions; Normal, Fixed Skew Normal, and Weibull are evaluated for all days of October, with an example day being displayed in Figure 12. The most interesting observation

from this is the ability of the Fixed Skew Normal distribution to capture high-frequency patterns. This is most evident in the peak occurring around 5:00, where the Fixed Skew Normal can capture the two peaks in the data connected by a plateau. Furthermore as reflected in the CRPS score in the figure, it overall better represents the observed data. And since the count distribution used is the same, the improvement is purely caused by a more representative TTD distribution.

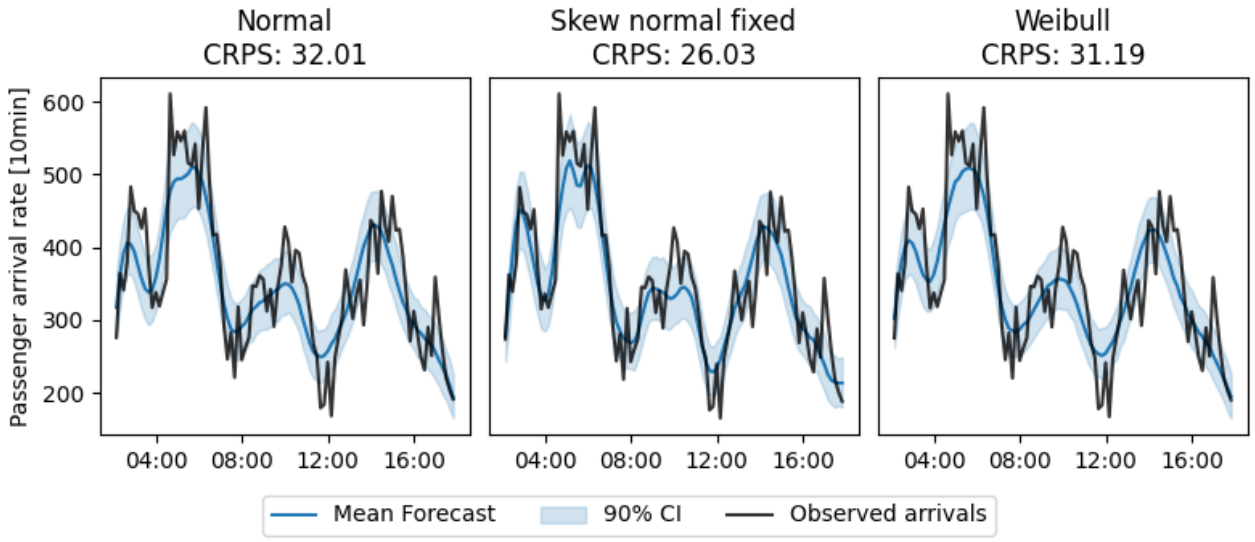


Figure 12: Forecast performance comparison for a single day between the three candidate distributions, showing the forecast vs arrival rate

The Fixed Skew Normal distribution provides the best scores for all metrics when evaluating the days of October, as can be seen in Table 3. Given this result, it must be determined whether the Fixed Skew Normal distribution has statistically significantly different scores. For this, the Kruskal–Wallis H test has been chosen, which is a non-parametric test for assessing whether samples originate from the same distribution. Here the distribution of the individual scores of the 10 minutes buckets are compared. With a significance of $\alpha = 0.05$, a statistically significant $P = .026$ difference was found between the distributions, rejecting the null hypothesis. Since there is a significant difference, the pairwise Mann–Whitney U test was used to identify specifically which distributions differ, with the null hypothesis being that there is no difference. Only the Fixed Skew Normal and the Normal distribution had a statistically significant difference with $P = .007$. With the difference between the Fixed Skew Normal and Weibull being $P = .107$. While this cannot be strictly considered to be statistically different, in combination with the other error metrics, it is enough to confidently choose the Fixed Skew Normal as the final TTD distribution model. Additional forecasting figures are presented in section D.3.

Distribution	CRPS	RMSE	MAE	R^2
Normal	31.83	53.29	42.11	0.61
Fixed Skew Normal	**29.01**	**49.81**	**38.86**	**0.66**
Weibull	30.86	51.81	40.76	0.63

Table 3: Comparison of distribution metrics, with the best performing score highlighted in bold

Finally, the difference between the total daily forecasted passengers is compared to the mean forecasted number of passengers. Here all arrivals are taken into account. Figure 13 provides the same plots as Figure 11 from the count distribution. With the mean percentage difference going from -1.3% to -1.91%, and the relative difference going from -552 to -784 passengers. However, the standard deviation of the relative difference is effectively unchanged going from 1132 to 1152. Indicating that the forecasting step only adds systematic errors. There are two sources for this -0.61% difference. As discussed in section 4.1 0.29% of passengers are excluded from training since they belong to flights with less than 10 passengers. The remaining missing 0.32% of passengers likely comes from the fact that TTD distributions have tails that contain some probability mass outside the $[-300, 0]$ range. Leading to the area under the distribution being less than 1 for the evaluated range. Which then will result in fewer passengers than the count distribution, when multiplied together.

In conclusion an overall mean difference of -1.91% corresponding to -784 passengers can be easily adjusted for. By adding a correcting factor with a similar magnitude to subsequent forecasts. The larger issues of the at most $\sim \pm 2000$ passengers from the mean on the other hand cannot be easily compensated for. However,

over the roughly 18 hours of active checkpoint time, this uncertainty corresponds to about $\sim \pm 100$ passengers per hour which is < 1 lane worth of throughput. This concept will be further explored in section 8 where the required number of lanes per 30-minute bucket will be evaluated.

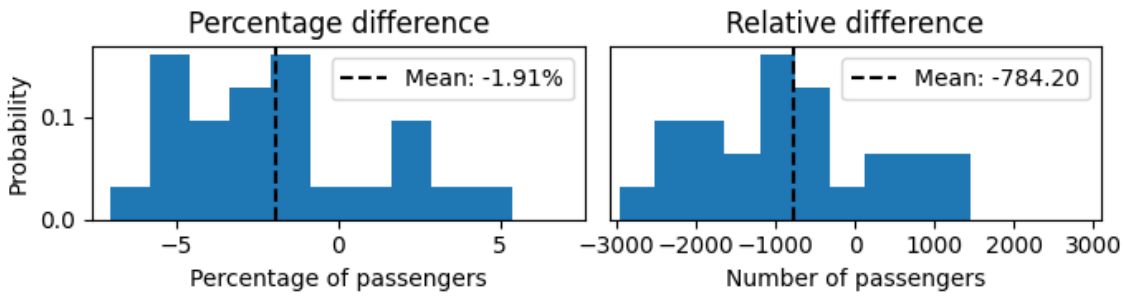


Figure 13: Daily passenger count forecasting performance for October, showing the percentage difference between the mean of the prediction and the arrivals [left], and the relative difference [right]

7 Real-Time Updating Model and Analysis

This section proposes a real-time update mechanism for the forecasting model using boarding card reader data. First, a brief motivation and overview of this approach will be presented in section 7.1. And then section 7.2 will discuss fundamental issues found with the suggested Bayesian approach.

7.1 Real-Time Updating Motivation & Overview

On larger time scales such as the tactical time frame, ranging from days to weeks in the future, a static forecast is perfectly satisfactory. Especially given the constraints related to shift scheduling, high frequency, small, and short-term updates to the forecast are not particularly useful. This is the domain that most literature deals with. However, two practical applications on the operational time frame would benefit from updating the forecast with real-time data. These are; break management, where security agents need to be provided some short, unscheduled breaks throughout the shift, and checkpoint balancing, where security agents can be shifted between checkpoints depending on necessity. Both of which would benefit from having more accurate short-term forecasts.

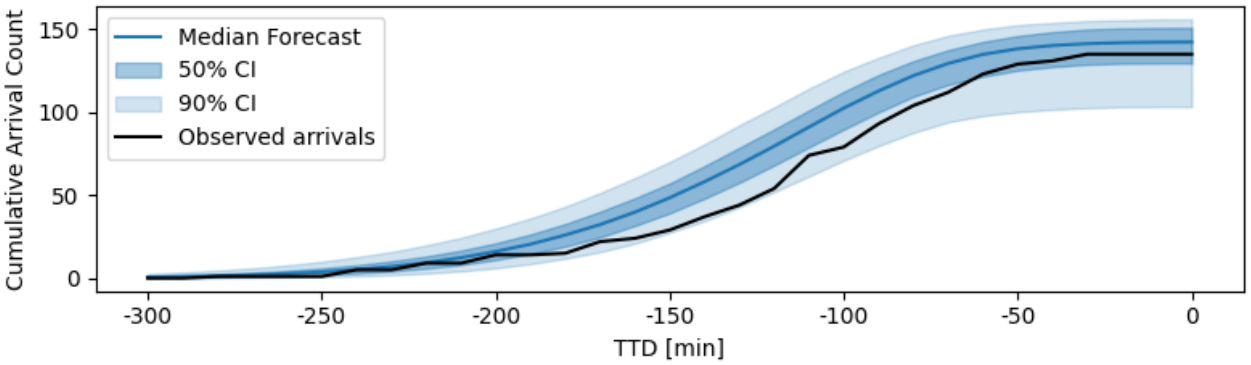


Figure 14: CDF of the forecast and the observed arrivals for a flight

To address this gap, a real-time updating approach is proposed that leverages the Bayesian framework. The output of the static forecasting algorithm is utilised, which is composed of two parts, the TTD distribution, and the count distribution. Given that passengers arrive i.i.d, existing arrivals should not change the shape of the TTD distribution of the remaining passengers. However, it will influence the magnitude of the remaining number of arrivals. Therefore the hypothesis is proposed that updating the count distribution with the already arrived passengers will improve the prediction of the final number of arrivals, and consequently improve the forecast. An example of this problem is visualised in Figure 14 showing the cumulative forecast and actual arrivals. Overall the updating algorithm takes in observations of the already arrived passengers, creating a new beta distribution that represents this observation. This new beta distribution then is used to perform Bayesian

inference on the original count distribution to update the expected number of total passengers. Finally, the remaining arrival rate is adjusted. A more detailed breakdown of the steps proposed is given below:

- **Converting observed count to a representative load factor observation** - For a given flight, observation of the number of arrived passengers is compared against the CDF of the flight's arrival rate prediction. Which contains a distribution of the predicted number of arrived passengers. From this, the percentile of the observation can be found. Using this percentile on the beta distribution used for the count distribution, an observation value can be calculated.
- **Converting the observation to a beta distribution** - In this step, an alternative parameterisation of the beta distribution is used. Using the mean μ, and the "weight" w which is the sum of the α and β parameters can be seen as the confidence in the mean. The observation from the previous point corresponds to μ. The value of w can be set based on arbitrary factors such as; time to departure, number of already arrived passengers, and prediction fraction of arrivals. Given these two values, the observation is converted into a beta distribution.
- **Updating the original count distribution** - Here Bayesian inference is used to update the original distribution with the the observed distribution. The parameters of μ and w can be converted back with $\alpha = \mu * w$ and $\beta = (1 - \mu) * w$. And since the beta distribution is a conjugate prior these parameters can simply be added to the parameters of the initial static prediction of the count distribution. Which results in a new updated count distribution, describing the new expected number of passengers.
- **Adjusting remaining arrival rate** - The above three steps describe how the final count distribution is updated, however, this does not directly take into account the actual number of arrivals. Given the observed and expected total arrivals, the original arrival rate needs to be adjusted to correspond to the newly updated remaining number of arrivals. For instance, if observation indicates the maximum capacity has already arrived (100% load factor), the updated count distribution will reflect this, and the remaining arrival rate should adjust to 0.

7.2 Real-Time Updating Approach Evaluation

The above-outlined method has been implemented, and while the individual steps work, and on average improve the performance of the prediction. When aggregated it decreases the forecasting accuracy. Specifically, a natural asymmetry in the data causes the update to have a consistently negative bias. With nearly all updates predicting fewer passengers than the static forecast. The updating step has been identified as the reason for this, with the subsequent adjusting step amplifying errors/bias. To demonstrate this, experiments have been carried out, using a 10-minute bucket size and are updated at every step. Evaluation of the update uses CRPS, the updated count distribution compared with the final number of arrivals. At each time step the score is divided by the score of the static prediction score. This results in a normalised value that allows for aggregation, and a clearer overview of relative changes. With the relative score ratio being 1 if the update is the same as the static forecast, <1 if they improved, and >1 if they got worse.

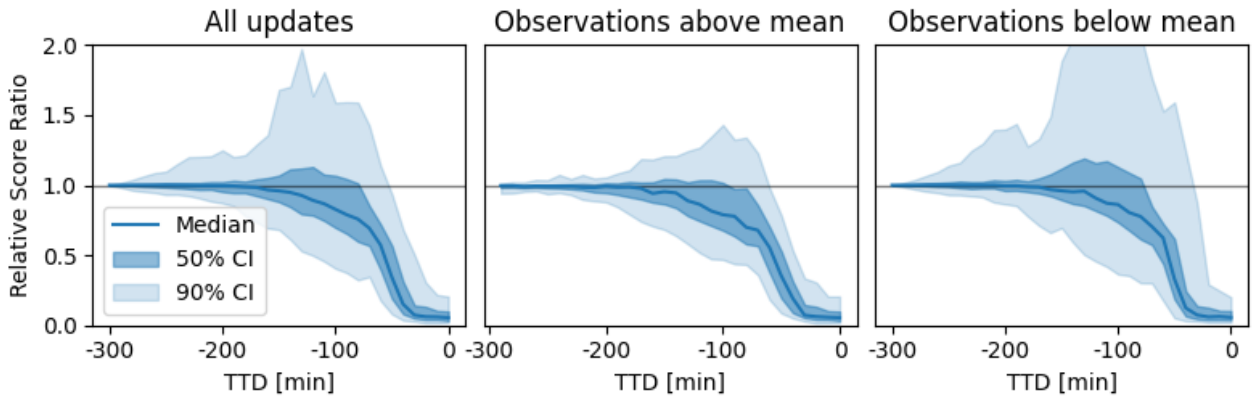


Figure 15: Relative score ratio for updating at a given TTD, containing 200 flights, all scores [left], scores when observation is above the mean of forecast [centre], scores when observation is below the mean of forecast [right]

On the left in Figure 15 the distribution of the relative scores ratio can be seen for 200 flights. Here the effect of updates is shown depending on how much before departure the update took place. The solid blue line, representing the mean, shows that updates at any time are generally as good as or better than the initial

prediction. However, at around 2 hours (-120 min) before departure, a significant proportion of the updates result in worse scores. This means that updates at this time have a reasonably high chance of making the prediction worse, compared to not doing anything. While this might seem to show a flaw in the system, counterintuitively it is expected and desirable. The reason for this is clearer when considering the hypothetical situation where the performance of the prediction for nearly all flights is only improving. With a Bayesian framework, this would only be possible if the static prediction would have a large bias where observations are always either above or below the original forecast for each flight. In a well-calibrated model, it is expected that observation will fluctuate around the mean, occasionally causing incorrect updates.

Since the scores, on average improve for flights, this should still mean an improved forecasting performance. But this is not the case. The more nuanced reason for this can be seen by separating update scores based on whether the observed number of passengers is above or under the mean of the static forecast. When comparing the centre and right plots in Figure 15, it can be seen that when the observation is above the mean, only a relatively small fraction of updates degrade performance, having a relative score ratio above 1. Whereas, as seen in the right plot, a significantly larger fraction of updates degrade performance when the observation is under the mean. In simple terms, when the number of passengers observed is above the mean of the forecast, updating on average improves the score more than when the observed number is below.

The reason for this behaviour stems from the fact that when updating the count distribution, it is not strictly speaking a "pure" Bayesian inference update. The key difference is the additional constraint of the final count never being less than the observed count. Which artificially decreases the range of possible outcomes, proportional to how high the observed number of passengers is. This can be more clearly illustrated by two examples corresponding to the two main cases:

- **90% of the capacity of the aircraft is observed to have arrived** - Depending on the distribution of the initial arrival count this will map to an update observation on the range of 90-100%. Naturally, the final number of passengers will also have to be in the range of 90-100% of capacity. Meaning that even in the worst-case scenario the error will be at most 10% of capacity.
- **10% of the capacity of the aircraft is observed to have arrived** - Regardless of the values or how updating is carried out, now the actual number of passengers can be anywhere between 10-100%. There is no way to determine whether the low number of arrivals is going to continue, or a sudden surge will arrive.

This asymmetry between updating for high and low observations is then turned into a bias by the last step of the updating algorithm, the adjustment of the remaining arrival rate. Using the above mentioned two scenarios. Whenever significantly more passengers are observed than predicted, then the remaining arrival rate can be confidently decreased. Alternatively whenever significantly fewer passengers are observed, then the arrival rate cannot be confidently increased. When updating, there are likely a few flights that fall into each of these categories. This means that there are a few flights for which the arrival rate is confidently decreased, while a few flights increase their remaining arrival rates with low confidence. Leading to a negative bias in updating. This can be seen as an emergent behaviour of the system, where individual updates on average improve performance, however as a system creates new behaviour. In conclusion, the use of real-time data and updating are promising, however, due to the intricacies of the problem, a significantly more sophisticated approach is likely necessary.

8 Case Study - Lane Requirement Estimation

An example use case is presented by utilising the output of the developed static forecasting model to determine the number of required lanes. First, a brief description of the lane requirement estimation algorithm will be presented in section 8.1 followed by an evaluation of the resulting lane requirements in section 8.2.

8.1 Lane Requirement Estimation Algorithm

The goal of the following algorithm is to evaluate the number of required lanes for each time period, given the distribution of possible arrivals. In simple terms, the number of arrivals can be seen as a demand for a given throughput for the checkpoint, while the number of security lanes is the supply. The goal is to determine the number of required lanes, and therefore the supply, that will best fit the demand throughout the day. Ideally, supply meets demand exactly, avoiding too much supply which results in higher operational costs, and too little supply which results in degraded passenger experience. Additionally, the algorithm needs to take into account the uncertainty provided by the forecast. To this end, a stochastic sampling-based algorithm was developed, which leverages Monte Carlo simulation methodology. Which, in its essence, involves repeatedly sampling from the probability distributions of demand and supply to estimate the likely outcomes.

In addition to the uncertainty of the arrival rate, the throughput of a lane is also stochastic, with performance depending on a wide range of factors, such as; experience or fatigue levels of agents, type of passengers, etc. From available data, a Normal distribution with a mean of 200 and a standard deviation of 30 was chosen to represent the possible throughput of a lane in an hour. The algorithm then draws a large number of samples for each number of lanes, providing a distribution of possible throughput for each. Then for each sample for the demand, provided by the forecast, the number of required lanes is estimated by identifying the lowest number of lanes that can supply that demand. Aggregating these then gives the percentage chance that opening a given number of lanes will be able to meet the demand. The performance of the overall system is measured in lane hours, which corresponds to the cumulative time of all open lanes.

8.2 Results

An example output of the above described algorithm is provided in the left plot of Figure 16. Here the darkness of the grid space shows the likelihood of requiring a certain number of lanes for each 30-minute period throughout a day. The plot on the right shows the required number of lanes, this was found through using the observed number of arrivals to determine the number of required lanes. A close agreement can be seen between the two, with the number of required lanes being much more confident because only the throughput has stochasticity. Whereas the estimated number of lanes also has stochasticity from the forecast and therefore is less confident.

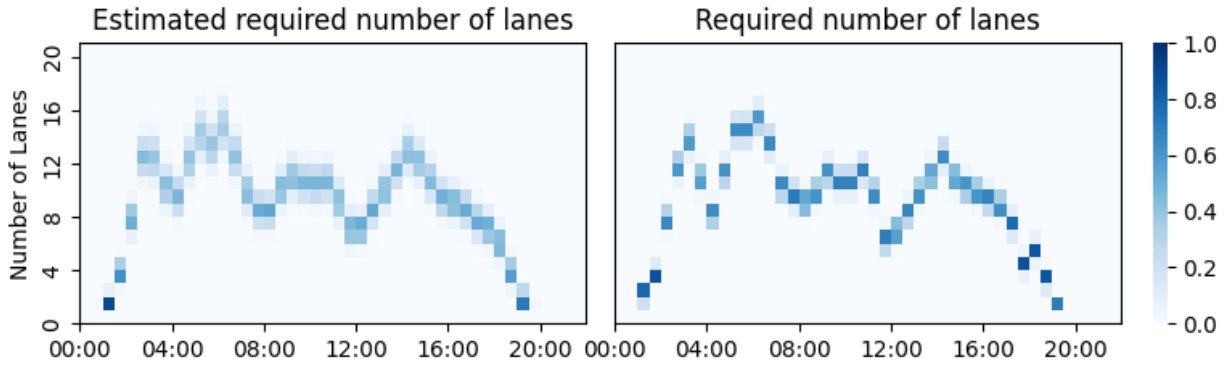


Figure 16: Number of lanes required lanes per 30 minutes, showing the estimation that used the forecast [left], and the actual required number of lanes using observed arrival count [right]

Evaluation of the effectiveness of the estimation must be carried out based on confidence levels. With the confidence level corresponding to the expected probability that a certain number of lanes meets or exceeds the required supply. A high confidence interval provides a high probability of meeting demand but at the cost of requiring more lanes to be open. Whereas at lower confidence the operational costs will be lower, but demand is more likely to exceed supply. To determine the optimal confidence level, the number of lane hours for both surplus (more lanes open than necessary) and deficit (not enough lanes open) must be taken into account. The evaluation was carried out on all days of October. Furthermore, the results below were achieved by applying a correcting factor for the -1.9% bias identified in section 6.3.

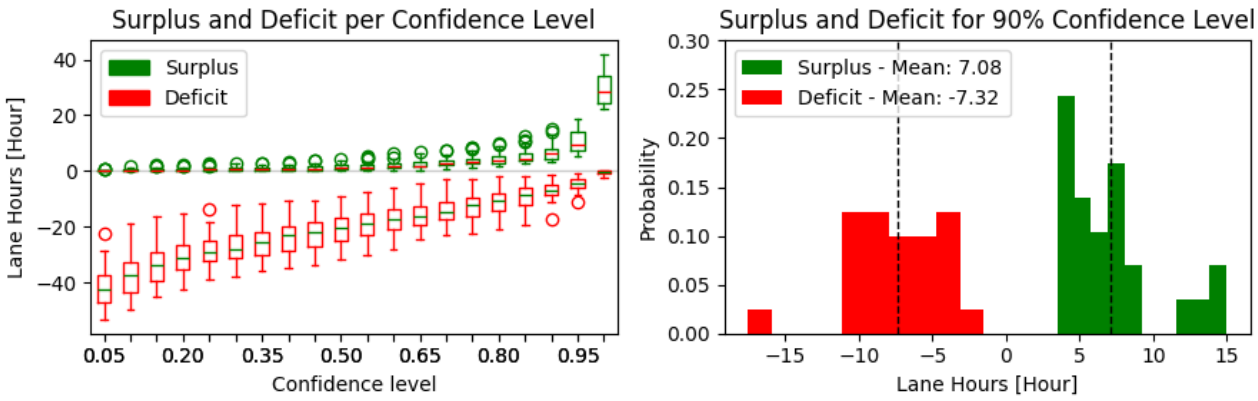


Figure 17: Evaluation of the number of required lanes by comparing the surplus and deficit lane hours [left] and focusing on best confidence level [left]

Figure 17 shows the distribution of daily surplus and deficit lane hours at each confidence level in the left plot. Here a clear bias can be seen for underestimating the required number of lanes, as the deficit is larger than the surplus for all but the highest confidence levels. However, this bias is slightly exaggerated since fractional surplus lane hours are not taken into account. For example, if one lane processes 200 passengers in an hour, and the demand is 201 passengers, then 2 full lanes are determined to be necessary. Here in theory the second lane would add 0.995 hours of surplus lane hours of throughput, but this is not and should not be measured, which results in the bias. The metric that matters is the amount of time when the estimated number of lanes matches the required number of lanes. This is maximised when the sum of surplus and deficit lane hours are at a minimum, which occurs for a 90% confidence level. This is shown in the right plot in Figure 17, where on average there are only around 7 hours of surplus and deficit each day. Since on average between 140-200 lane hours are required to meet daily demand, this corresponds to an average of $\pm 0.3 - 0.5\%$. This is likely not fully representative of the expected performance once deployed, however, these initial results are very encouraging. From this case study, both the accuracy of the forecasting model and the utility of confidence intervals are demonstrated. This approach affords decision-makers nuanced control, enabling them to tailor decisions according to desired risk profiles.

9 Discussion

Finally, the implications and limitations of the presented approach, are discussed. First section 9.1 outlines limitations caused by the data. This is followed by a brief analysis in section 9.2 about the architecture used for the two main models. Then the real-time updating approach is discussed in section 9.3. Concluding with section 9.4 by evaluating the overall effect of the work carried out.

9.1 Data and Data Availability

The primary reason that this work was possible was due to the boarding card reader data, and more specifically knowing which flight a passenger has arrived for. Obtaining this data is likely to be challenging for future work. While in a commercial setting obtaining this data is simpler, it is still a relatively strict requirement to use the developed approach. This will restrict the airports to which this model can be deployed to, with smaller and less technologically driven airports mainly being excluded. However, this is a trade-off for a more informative and high-performance forecasting algorithm. This is quite common for state-of-the-art approaches trying to achieve high performance, and therefore it is not considered to be a significant downside of this method.

The second issue relating to the utilised data has been the quality of it. As already discussed in section 4.2 a large number of days have missing data. While it is assumed that most of these issues have been filtered out, this cannot be stated for certain. This has also led to the evaluation of the forecast to be partially carried out on parts of the data that have been included in the training set. This is sub-optimal, and a more thorough investigation will be required once more "clean" data is available. However, this should not be as significant of an issue as it would have been for example for ML-based approaches. The primary concern usually is about overfitting on the training data, which then would show significantly higher performance compared to unseen data. However, this is a larger concern for models that have a lot of parameters, and have the expressibility to represent individual data points in the training set. While still of concern the relative simplicity and low number of parameters should negate any significant impact caused by this.

9.2 Model architecture

The model developed in this paper is a novel state-of-the-art approach in the field of forecasting passenger arrivals to security checkpoints by capturing uncertainty in the arrival rate. Because of this novelty the utilised models representing the TTD distribution and the count distribution, are both emphasise simplicity and robustness over functionality. The primary example of this is the hierarchical Bayesian model used for the TTD distribution, while it performs well, it cannot handle unseen flights. In the current implementation, the implicit representation of a wide number of features contained in the flight ID is leveraged to allow for a simpler model. Ideally, future models will be able to explicitly represent these features to be able to handle unseen flights. It is highly likely that explicit representations are less powerful due to larger amounts of noise, and should only be used for flights with little or no data. This will not only allow for never before seen flights to be forecasted but could also allow for forecasting on entirely new checkpoints without prior data. Finally, the current model assumes independence between TTD distribution parameters, however as seen in Appendix B this assumption does not hold. While it should be further investigated, it's unlikely to significantly change the performance of the final combined forecast.

Similarly, the count distribution model also relies on the implicit feature representation of the flight ID. This model could also benefit from explicitly modelling some features. Additional features that likely contain valuable

information have been identified, though their analysis and integration exceed this work's scope. These include the load factors of temporally near flights or the load factors of flights going to the same destination. While the presented count distribution model is novel and performs quite well, existing research in this context is relatively established. Albeit only providing point estimates for the number of passengers per flight. Integration of these models using a Bayesian framework is expected to produce good results. Finally, the assumption that the beta distribution can represent the count distribution has not been conclusively verified, however, the quality of the predictions is very good. Therefore without contrary evidence, it seems that the beta distribution is suitable for this task.

9.3 Real time updating

Other than quantification of uncertainty, the second goal of this work has been to update the forecasting model with real-time data. This has not been achieved, with the issue arising from observations of the number of passengers not being a "pure" Bayesian update. Since observations also truncate the distributions. Which affects the confidence with which the initial forecasting problem is adjusted. When a large number of passengers have already arrived the remaining arrival rate can confidently be decreased, however for a low number of arrivals it cannot be confidently increased. This combination of secondary effects then has a negative bias on the forecast. This natural phenomenon is anticipated to cause issues with further models, regardless of the approach used, and will have to be explicitly accounted for. Finally, the experience and intuition gained throughout this work have highlighted the extent to which stochasticity plays a role in the short-term arrival rates. Likely, making even highly sophisticated approaches provide only marginal improvements. From this insight, efforts for updating approaches should focus on adjusting the count distribution utilising information about neighbouring flights.

9.4 Outlook and Implications

As airports continue to modernise, improve customer experience, and streamline their operations, accurate and reliable forecasts will become more and more indispensable. However, existing approaches represent highly stochastic phenomena with point forecasts, missing a lot of information that decision-makers require. The model developed in this work overcomes this with its primary contribution being the quantification of uncertainty in arrival rates. Modelling the stochasticity of the total number of passengers, and the uncertainty of the arrival rate caused by different forecast bucket lengths. This empowers airport operators to make decisions with enhanced confidence and nuance. Particularly through choosing risk profiles that best suit the desires of the users. Allowing airports to make informed trade-offs between operational requirements and operational costs. This ability has been highlighted by a simple case study that determined the number of required lanes for scenarios with different risk profiles. Finally, while this work is very promising, allowing for robust and accurate decision support systems, there are some relatively stringent requirements. Firstly this approach is likely only suitable for large airports since these entities are more likely to have the infrastructure required to collect the required data. Furthermore, the approach performs more reliably with a large arrival rate where the effects of stochastic arrival rates are smoothed out. Overall, the static forecasting model developed here shows great promise in enhancing planning and decision-making processes for security managers. Thereby improving the operational efficiency of the airport without compromising the passenger experience.

10 Conclusions

This study set out to develop and evaluate a real-time probabilistic security checkpoint arrival rate forecasting model by utilising a Bayesian framework. The approach consisted of two main parts. First, the static forecasting model was developed by breaking down the forecasting problem into two prediction problems. The estimation of the time to departure (TTD) arrival distribution for each flight, which utilised a hierarchical Bayesian regression model, employing the flight ID and airline as features. Secondly, the number of arrivals of each flight was modelled with a Bayesian regression model. Using the flight ID, day of the week, and the moving average of previous passenger counts as features. These models were then sampled and combined for each flight with a flight schedule, resulting in the probabilistic forecasting model. Finally, a real-time updating approach was proposed to update the count distribution of each flight as new observations become available.

The proposed probabilistic static forecasting model was found to accurately capture the stochasticity of both the random arrival rate, and the total number of arrivals. This was accompanied by a sensitivity analysis performed on the two sub-models to ensure optimal distributions and parameter choice. The Skew Normal distribution was identified as the optimal fit for TTD distribution, with a 7-period moving average effectively capturing long-term seasonal trends. Unfortunately, it has been found that the real-time updating component of this work is not feasible with the current Bayesian approach. This is caused by an inherent asymmetry present in the problem, resulting in a negative bias when updating. Finally, a case study was carried out to

probabilistically evaluate the number of required lanes to meet the demand placed on them by the arriving passengers. The outcome of this validated the accuracy of the forecasting model and demonstrated the benefits of the probabilistic approach by quantifying risk for the decision-making process.

Based on these findings, and the limitations outlined, the following avenues for future research may be considered. Firstly it has been demonstrated that probabilistic approaches are well suited to this domain and should be considered in future works. Given its novelty, the forecasting model prioritised simplicity and robustness over expressibility and features. Therefore there could be significant potential performance and feature improvements for the TTD and count distribution models. Both of these should aim to incorporate additional features explicitly into their models. Such as the distance to and/or type of the destination, and departure time. Lastly, the utility of true real-time updating in such a stochastic environment is likely to be limited, with updating efforts better spent on improving the count distribution model. For instance, adjusting the model based on the load factors of temporally proximal flights could yield significant improvements.

References

[1] *aerodatabox*. Sept. 2023. URL: https://rapidapi.com/aedbx-aedbx/api/aerodatabox/.

[2] Eli Bingham et al. *Pyro: Deep Universal Probabilistic Programming*. arXiv:1810.09538 [cs, stat]. Oct. 2018. URL: http://arxiv.org/abs/1810.09538 (visited on 10/24/2023).

[3] David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. "Variational Inference: A Review for Statisticians". In: *Journal of the American Statistical Association* 112.518 (Apr. 2018). arXiv:1601.00670 [cs, stat], pp. 859–877. ISSN: 0162-1459, 1537-274X. DOI: 10.1080/01621459.2017.1285773. URL: http://arxiv.org/abs/1601.00670 (visited on 04/25/2023).

[4] Bin Chen, Xing Zhao, and Jin Wu. "Evaluating Prediction Models for Airport Passenger Throughput Using a Hybrid Method". en. In: *Applied Sciences* 13.4 (Feb. 2023), p. 2384. ISSN: 2076-3417. DOI: 10.3390/app13042384. URL: https://www.mdpi.com/2076-3417/13/4/2384 (visited on 04/14/2023).

[5] EUROCONTROL. *EUROCONTROL Forecast Update 2021-2027*. Tech. rep. Oct. 2021.

[6] Fortune Bisuness Insights. *Airport Services Market Size, Share, Growth | Global Report, 2027*. 2020. URL: https://www.fortunebusinessinsights.com/airport-services-market-102855 (visited on 03/07/2022).

[7] Charles Geyer. "Introduction to Markov Chain Monte Carlo". en. In: *Handbook of Markov Chain Monte Carlo*. Ed. by Steve Brooks et al. Vol. 20116022. Series Title: Chapman & Hall/CRC Handbooks of Modern Statistical Methods. Chapman and Hall/CRC, May 2011. ISBN: 978-1-4200-7941-8. DOI: 10.1201/b10905-2. URL: http://www.crcnetbase.com/doi/abs/10.1201/b10905-2 (visited on 04/25/2023).

[8] Alan (Avi) Kirschenbaum. "The cost of airport security: The passenger dilemma". en. In: *Journal of Air Transport Management* 30 (July 2013), pp. 39–45. ISSN: 09696997. DOI: 10.1016/j.jairtraman.2013.05.002. URL: https://linkinghub.elsevier.com/retrieve/pii/S0969699713000458 (visited on 04/30/2023).

[9] Ziyu Li, Jun Bi, and Zhiyin Li. "Passenger Flow Forecasting Research for Airport Terminal Based on SARIMA Time Series Model". en. In: *IOP Conference Series: Earth and Environmental Science* 100.1 (Dec. 2017). Publisher: IOP Publishing, p. 012146. ISSN: 1755-1315. DOI: 10.1088/1755-1315/100/1/012146. URL: https://dx.doi.org/10.1088/1755-1315/100/1/012146 (visited on 01/12/2023).

[10] Lijuan Liu and Rung-Ching Chen. "A novel passenger flow prediction model using deep learning methods". en. In: *Transportation Research Part C: Emerging Technologies* 84 (Nov. 2017), pp. 74–91. ISSN: 0968-090X. DOI: 10.1016/j.trc.2017.08.001. URL: https://www.sciencedirect.com/science/article/pii/S0968090X17302024 (visited on 04/04/2023).

[11] James Mitchell and Kenneth F. Wallis. "Evaluating density forecasts: forecast combinations, model mixtures, calibration and sharpness". en. In: *Journal of Applied Econometrics* 26.6 (2011). _eprint: https://onlinelibrary.wile pp. 1023–1040. ISSN: 1099-1255. DOI: 10.1002/jae.1192. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/jae.1192 (visited on 07/17/2023).

[12] Philippe Monmousseau et al. "Predicting Passenger Flow at Charles De Gaulle Airport Security Checkpoints". In: *2020 International Conference on Artificial Intelligence and Data Analytics for Air Transportation (AIDA-AT)*. Feb. 2020, pp. 1–9. DOI: 10.1109/AIDA-AT48540.2020.9049190.

[13] Maria Nadia Postorino et al. "Airport Passenger Arrival Process: Estimation of Earliness Arrival Functions". en. In: *Transportation Research Procedia*. 21st EURO Working Group on Transportation Meeting, EWGT 2018, 17th – 19th September 2018, Braunschweig, Germany 37 (Jan. 2019), pp. 338–345. ISSN: 2352-1465. DOI: 10.1016/j.trpro.2018.12.201. URL: https://www.sciencedirect.com/science/article/pii/S2352146518306173 (visited on 01/12/2023).

[14] Álvaro Rodríguez-Sanz et al. "Queue behavioural patterns for passengers at airport terminals: A machine learning approach". en. In: *Journal of Air Transport Management* 90 (Jan. 2021), p. 101940. ISSN: 0969-6997. DOI: 10.1016/j.jairtraman.2020.101940. URL: https://www.sciencedirect.com/science/article/pii/S0969699720305238 (visited on 01/12/2023).

[15] Ratna Sulistyowati et al. "Hybrid forecasting model to predict air passenger and cargo in Indonesia". In: *2018 International Conference on Information and Communications Technology (ICOIACT)*. Mar. 2018, pp. 442–447. DOI: 10.1109/ICOIACT.2018.8350816.

[16] Paul Pao-Yen Wu and Kerrie Mengersen. "A review of models and model usage scenarios for an airport complex system". en. In: *Transportation Research Part A: Policy and Practice* 47 (Jan. 2013), pp. 124–140. ISSN: 0965-8564. DOI: 10.1016/j.tra.2012.10.015. URL: https://www.sciencedirect.com/science/article/pii/S0965856412001541 (visited on 04/05/2023).

Appendices

A Data Filtering

In this section, a quick overview of the filtering approach and algorithm is presented, followed by a discussion of the parameters used for this. The data for this work has been collected in collaboration with GRASP Innovations. Specifically, this included using an API provided by a large European airport to request data every minute. This data was not actively monitored, which allowed a database misconfiguration to cause issues with the data. This can be seen in Figure 18 where 3 of these events can be seen, with data being bucketed at 1-minute intervals. The primary point of interest is that all of these anomalies are 5 minutes long, and they start and end at even 5-minute intervals.

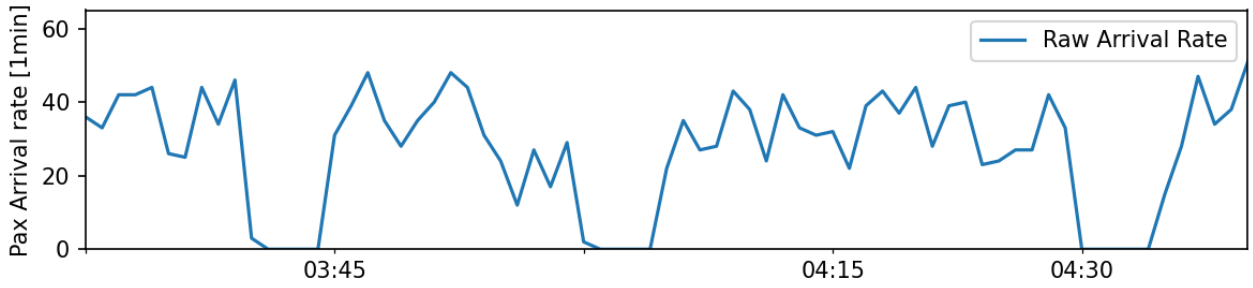


Figure 18: Detailed view of data anomalies presented, showing missing data in 5-minute periods

These two observations have allowed for a significantly simpler anomaly detection algorithm, than without. By simply bucketing the data into 5-minute buckets, buckets will fully encompass periods that have been influenced by the data issue. However occasionally, buckets containing the anomalies will have 1-3 passengers recorded (compared to the 100s in other 5-minute buckets). This means that it is not possible to just identify buckets with no passengers, furthermore, there are periods when there are naturally no or very few passengers. Therefore the value of each bucket is compared against the running average of the number of passengers. If the difference between the running average and the value of the bucket exceeded a threshold value, then the bucket was marked as an anomaly.

Once the temporal location of these events has been identified, the affected flights must be found. Since it has been found that passengers arrive between $[-300, 0]$ minutes before departure, any anomalies in these ranges will affect the flight. From the perspective of the anomaly, any flight departing 300 minutes after or less is affected. And as discussed in the section 4.2 this results in two parameters that can be adjusted. The number of allowable anomalies, and the range that they affect. Table 4 shows the available data, before filtering. For subsequent overviews, November is excluded since it only contains 6 days of data and no anomalies.

Date	Flights	Pax (000s)
2023-06	9872	916.31
2023-07	10457	1122.59
2023-08	10775	1189.87
2023-09	10270	1091.10
2023-10	9556	1010.45
2023-11	1905	211.61
Total	52835	5541.93

Table 4: Total number of flights and thousands of passengers in the dataset.

Given the strictest parameter settings of no anomalies and the full TTD range of $[-300, 0]$ minutes, Table 5 shows that it would exclude 36% of flights and 35% of passengers. This is already a significant improvement over the naive approach where all data for a day with an anomaly would be discarded, which would remove 57% of the data. However, this can be slightly improved if the data quality is allowed to be slightly degraded. One option is to loosen the requirement of $[-300, 0]$ to $[-250, -10]$ minutes range, shown in Table 6. This should have relatively little impact on data quality, as passengers seldom arrive in the difference of the TTD ranges used, however, it also only results in $\sim$ 3% more available data. Alternatively allowing for a single anomaly period results in lower-quality data but also gives a lot more data. As seen in Table 7, this provides $\sim$ 10% more data than 0 anomalies. Since a single period of 5 minutes should on average contain only 0-3 passengers depending on the flight, this is seen as an acceptable trade-off. Finally, if both approaches are combined then, as given in Table 8 $\sim$ 13% more data can be used. This is especially important for July, as discarded data dropped from 80% to 62%, nearly doubling the available data for this month.

Date	Flights	Flights [%]	Pax (000s)	Pax [%]	Days	Days [%]
2023-06	3671	37.19	325.15	35.49	18	60.00
2023-07	8426	80.58	905.23	80.64	31	100.00
2023-08	4199	38.97	469.41	39.45	19	61.29
2023-09	1057	10.29	110.13	10.09	10	33.33
2023-10	1187	12.42	78.74	7.79	9	30.00
Total	18540	36.40	1888.66	35.43	87	57.24

Table 5: Data affected by filtering with parameters: max anomaly periods = 0, TTD range = [-300, 0]

Date	Flights	Flights [%]	Pax (000s)	Pax [%]	Days	Days [%]
2023-06	3318	33.61	294.34	32.12	18	60.00
2023-07	7898	75.53	845.73	75.34	31	100.00
2023-08	3723	34.55	415.94	34.96	19	61.29
2023-09	907	8.83	94.80	8.69	10	33.33
2023-10	1046	10.95	70.51	6.98	9	30.00
Total	16892	33.17	1721.32	32.29	87	57.24

Table 6: Data affected by filtering with parameters: max anomaly periods = 0, TTD range = [-250, -10]

Date	Flights	Flights [%]	Pax (000s)	Pax [%]	Days	Days [%]
2023-06	2946	29.84	254.84	27.81	17	56.67
2023-07	7209	68.94	764.63	68.11	31	100.00
2023-08	2484	23.05	267.35	22.47	14	45.16
2023-09	270	2.63	25.85	2.37	3	10.00
2023-10	875	9.16	50.29	4.98	6	20.00
Total	13784	27.06	1362.96	25.57	71	46.71

Table 7: Data affected by filtering with parameters: max anomaly periods = 1, TTD range = [-300, 0]

Date	Flights	Flights [%]	Pax (000s)	Pax [%]	Days	Days [%]
2023-06	2557	25.90	219.38	23.94	17	56.67
2023-07	6590	63.02	692.75	61.71	31	100.00
2023-08	2093	19.42	224.42	18.86	14	45.16
2023-09	195	1.90	18.98	1.74	3	10.00
2023-10	763	7.98	45.92	4.54	6	20.00
Total	12198	23.95	1201.45	22.54	71	46.71

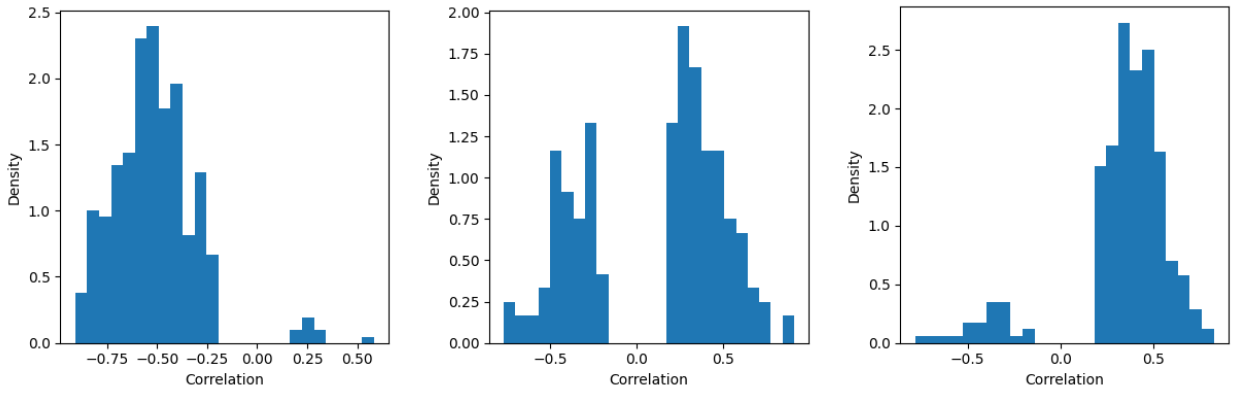
Table 8: Data affected by filtering with parameters: max anomaly periods = 1, TTD range = [-250, -10]

Finally, there were several days from which data was severely "damaged". Here the simple algorithm developed was not able to correctly identify the faulty data. Therefore the following days were discarded manually in whole. The exclusion of especially the data from the 21-25th of October has caused issues with validation, which will be also expanded on in section D.2.

- 10-11 July 2023
- 17 October 2023
- 21-25 October 2023

B TTD Model

The following section outlines additional discussions and analyses performed for the TTD model. An assumption used for the TTD model that has not been discussed in section 5.1 is the covariance between model parameters. In simple terms, the current TTD model fits an independent distribution to each parameter of a parametric distribution. This does not take into account that, as an example, given a lower mean TTD (passengers arrive closer to departure) the distribution is likely to have a lower TTD variance. In Figure 19a - 19c the distribution of correlation values can be seen. These were evaluated by grouping the parameters fit to each flight's distribution by the flight ID. Aggregating the correlation coefficient only for flight IDs for which there was a statistically significant ($p < .05$) correlation.



(a) Normal (mean - std)
23.08% no correlation

(b) Weibull (shape - scale)
60.66% no correlation

(c) Fixed Skew Normal (mean - std)
41.32% no correlation

Figure 19: Distribution of correlation coefficients for different TTD models

Both the Weibull and Fixed Skew Normal distributions have a significant percentage of flight IDs with no significant correlation. Furthermore, both of these distributions have a significant proportion of coefficients close to 0. Only the Normal distribution has a large proportion of correlation coefficients with high and statistically significant values. This could imply the need for a multivariate Gaussian, instead of using two independent Normal distributions to model the parameters. While this could be investigated in future works, the following points outline the reasons that it has not been explored in this paper;

- **Model complexity** - This work already demonstrates a novel concept, developing a full forecasting model pipeline, and real-time updating component, including a case study. Implementation of a multivariate Gaussian for the TTD model would introduce additional complexities in several steps for a novel approach.

- **Distribution choice** - Correlation between parameters is most significantly pronounced for the Normal distribution. The chosen distribution, Fixed Skew Normal, has notably fewer flight IDs with significant or high correlation coefficients.
- **TTD model sensitivity** - Since the analysis of the distributions showed no statistically significant difference, section 6.1. And even for the combined forecast, the differences were low, section 6.3. It is highly unlikely that modelling covariance would have a larger influence than using a different distribution.

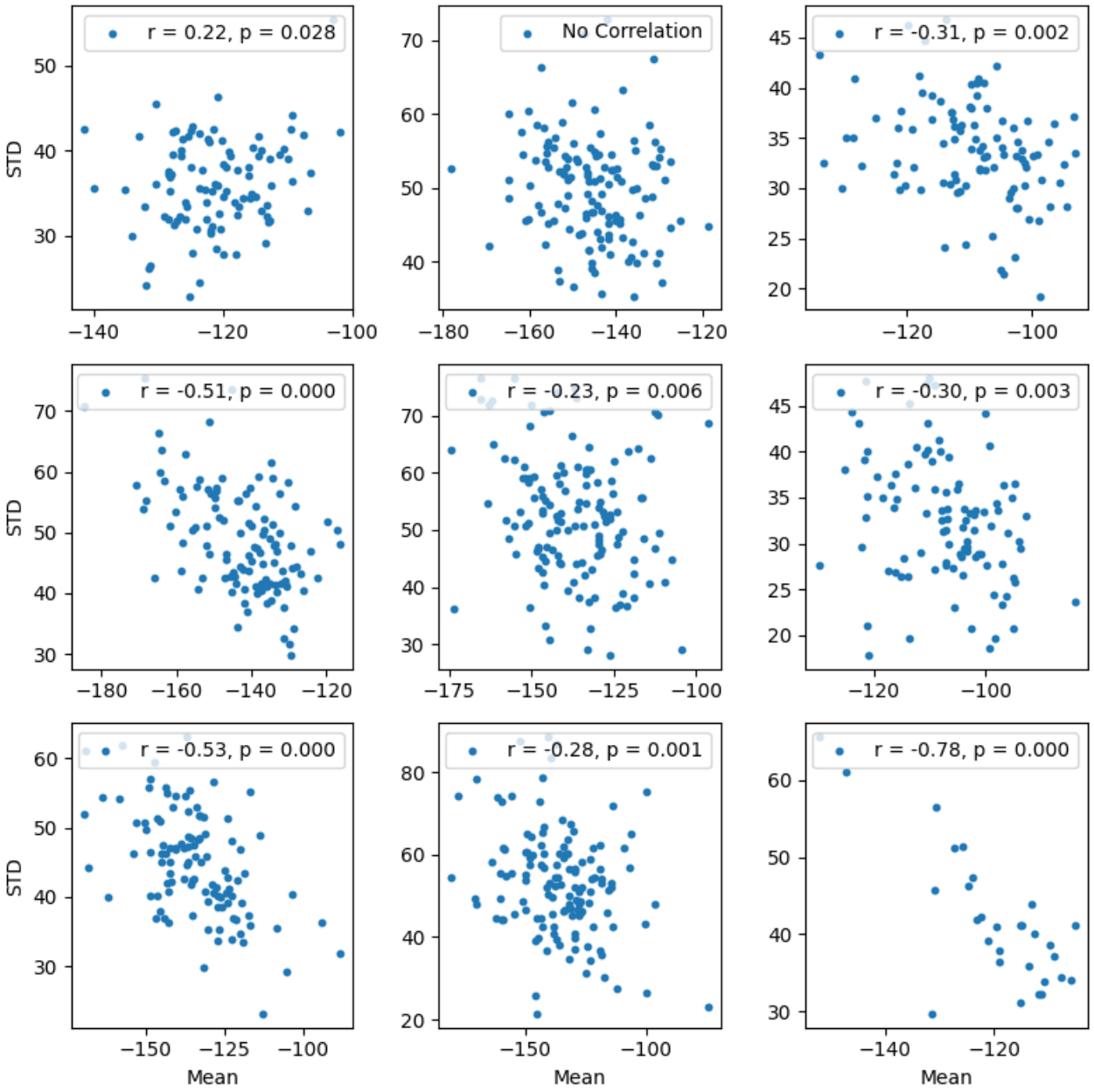


Figure 20: Normal TTD model - Correlation between std and mean per flight ID

Several example scatter plots, per flight ID, of the distribution parameters are given in Figure 20 - 22. While the parameters might have a statistically significant correlation, usually the main "mass" of the parameters is still relatively evenly concentrated. Other than a few outliers, independently fitting parameters does not seem to be a significant modelling issue. Especially when taking into account the above three points.

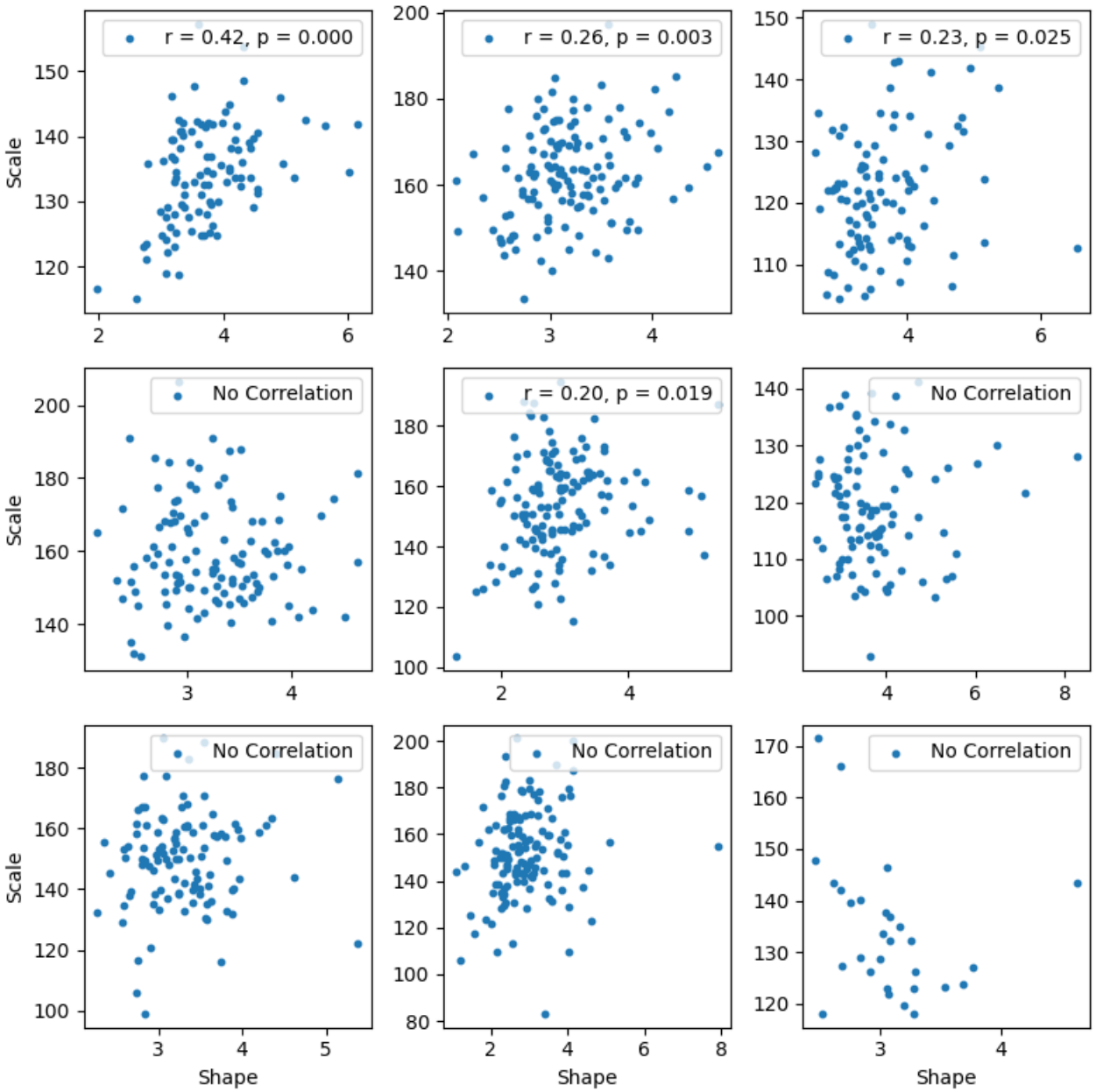


Figure 21: Weibull TTD model - Correlation between scale and shape per flight ID

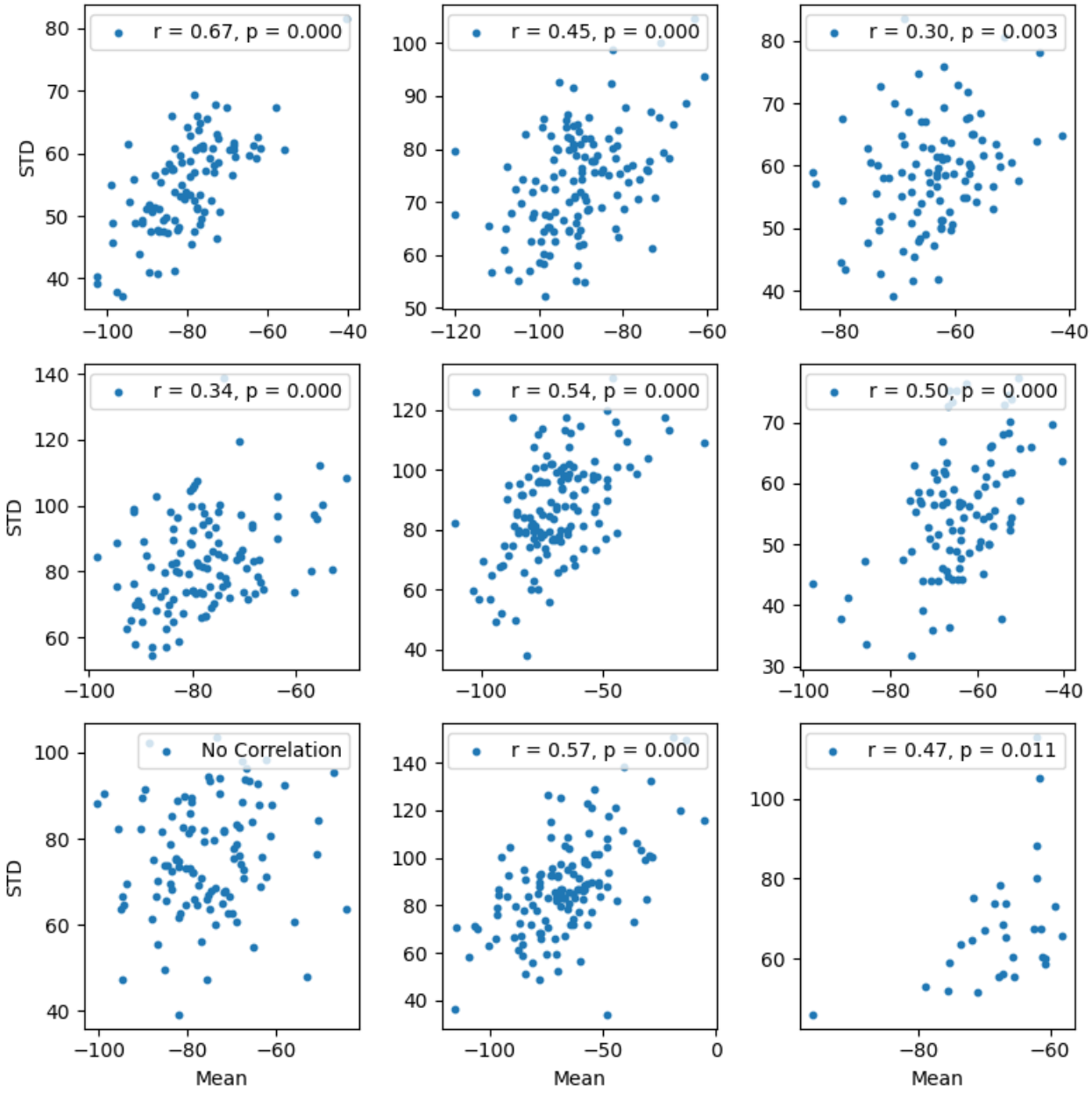


Figure 22: Fixed Skew Normal TTD model - Correlation between std and mean per flight ID

C Combined Static Forecasting Model

This section dwells deeper into the observations from section 5.3 which was the interpretation of the output of the model being the average arrival rate. This is then converted to a distribution of discrete events with the Poisson distribution. The main reason for this is variable bucket sizes. Since the forecasting model uses distributions, they can be sampled at different frequencies. However, if the bucket does not contain enough samples then randomness becomes highly impactful. Figure 23 shows the raw output of the forecast, which is the combination of the TTD and count models and the flight schedule. In comparison Figure 24 applies the Poisson distribution. It is especially evident for smaller bucket sizes where the confidence interval becomes much wider.

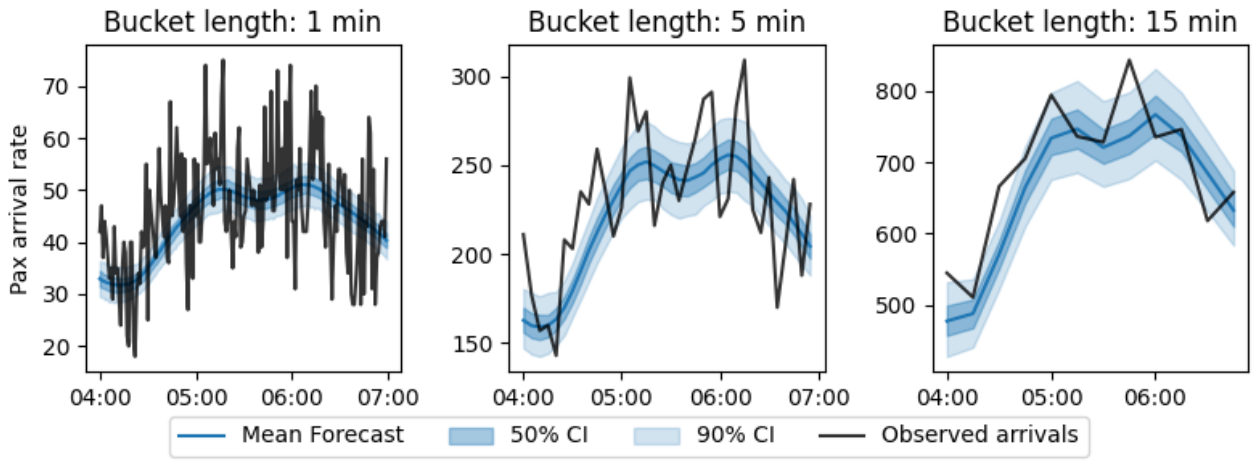


Figure 23: Example raw forecast at different bucket lengths

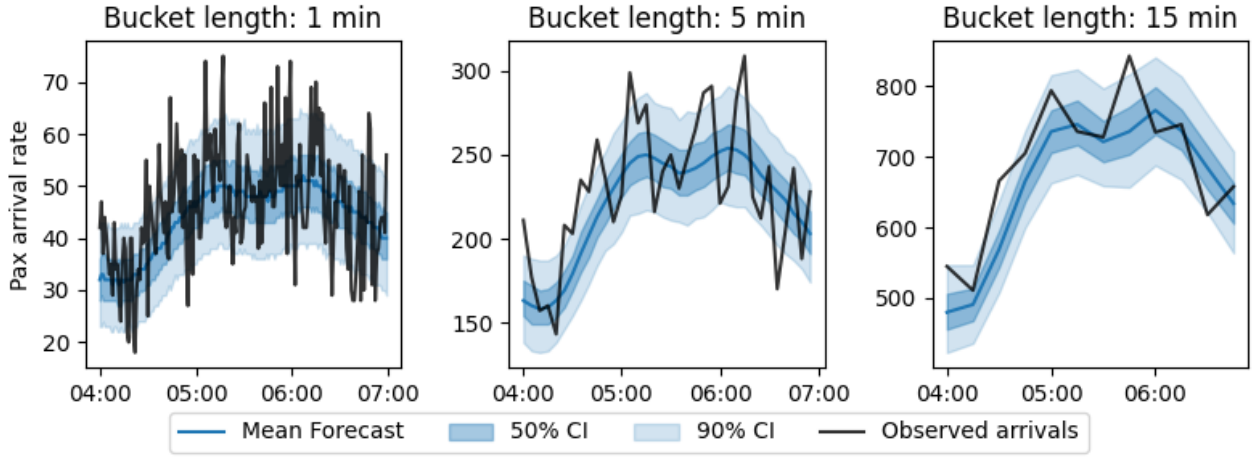


Figure 24: Example forecast converted to discrete events at different bucket lengths

While including the Poisson's process improves the width of the confidence intervals, as can be seen in going from Figure 25 to Figure 26. Where the left plot shows the forecast and the actual number of arrivals/observations. And on the right the PIT (Probability Integral Transform) histogram, which shows the distribution of observations across the density forecast. Ideally this should create a histogram where the probability of observations is equally distributed across all forecasted percentiles. That is, for example, 50% of the data should be in the 50% confidence interval. Or for these plots, 10% of the data should be in each 10% percentile range. This result implies that the final density forecast is too narrow, which can be seen from over represented proportion of observations being in the 0-10 and 90-100 percentiles. The exact reason for this is unclear. However, it likely has to do with the fundamental approach used for the TTD distribution. It was found that the *average* TTD arrival rate could be adequately modelled by a parametric distribution, which provided many beneficial properties. While it correctly captures the overall behaviour of the global arrival rate, it's not able to correctly model outliers. These are relatively common, one such example could be a large group arriving together. To correctly model these, the variance and probability of these events could be directly modelled.

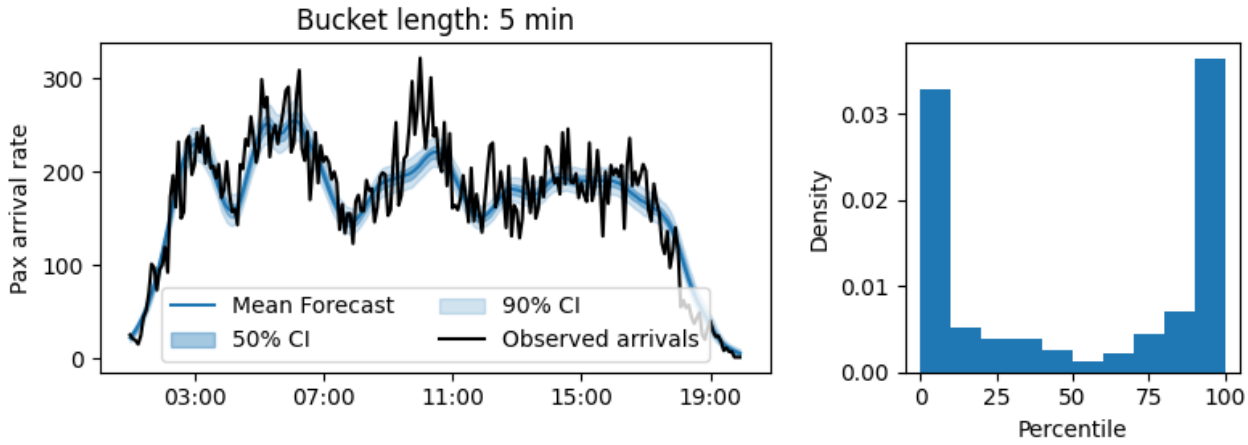


Figure 25: Day forecast [left], and distribution of observed values in the density forecast [right]

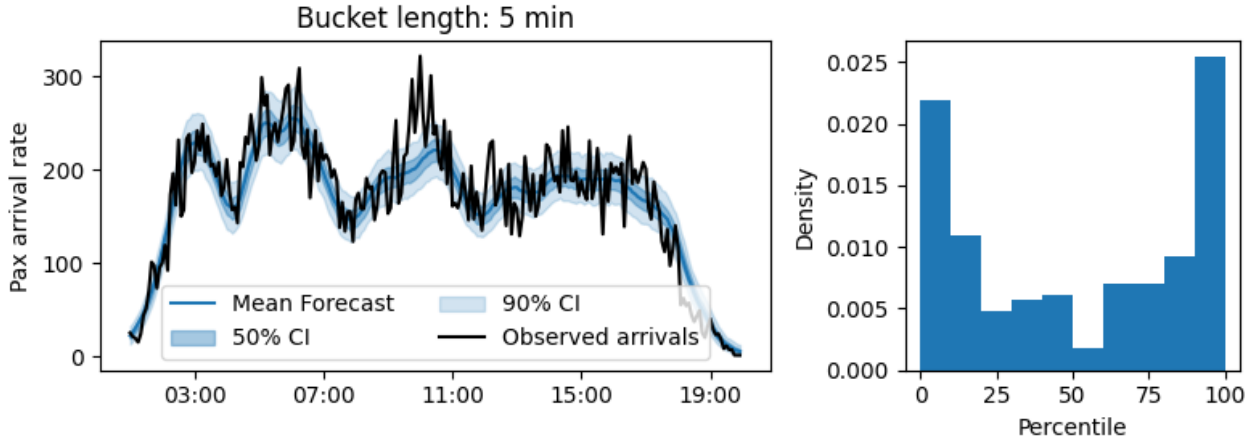
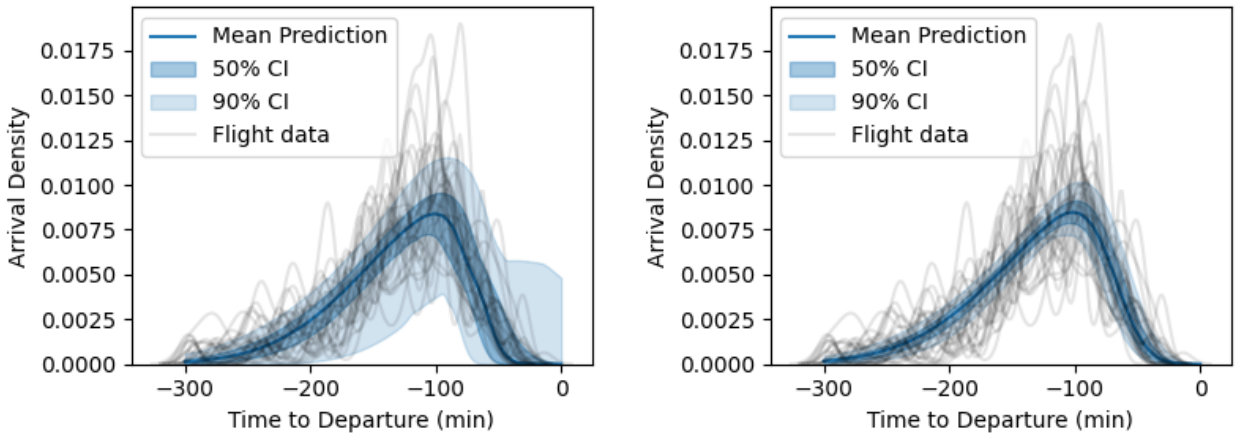


Figure 26: Expanded day forecast [left], and distribution of observed values in the density forecast [right]

D Model evaluations

D.1 TTD model

This section discusses the observed issue with the Skew Normal distribution and the ad hoc solution of the fixed Skew Normal distribution. As shown in section 6.1 the the Skew Normal distribution performed significantly worse than the other distributions evaluated. Figure 27a shows an example flight ID fitted with the Skew Normal distribution. The light grey lines are smoothed arrival rates from individual flights. Here the reason for the poor performance can be seen near the departure, near 0 min. There is an issue with the distribution, this has been attributed to the dependent non-linear relationship between parameters. Figure 28 illustrates this issue, fixing the mean to 0 and the std to 1, changes in the skew parameter influence both the mean and the variance. This behaviour is especially severe with low skew values, changing the skew parameter from 1 to -1 would result in the mean shifting from 0.56 to -0.56. And since the TTD model uses normal distributions for parameter values, if the skew of a flight ID is close to 0, then random sampling will draw positive and negative skew values. This results in the issue outlined above. To overcome this, a Skew Normal distribution is fitted to all flights in a flight ID, which gives the average skew for that flight ID. This value is then fixed for flights of the flight ID, and distributions are fitted to each flight to get varying mean and std values. The new distribution is shown in Figure 27b.



(a) Skew Normal

(b) Fixed Skew Normal

Figure 27: TTD distribution comparison for a single flight

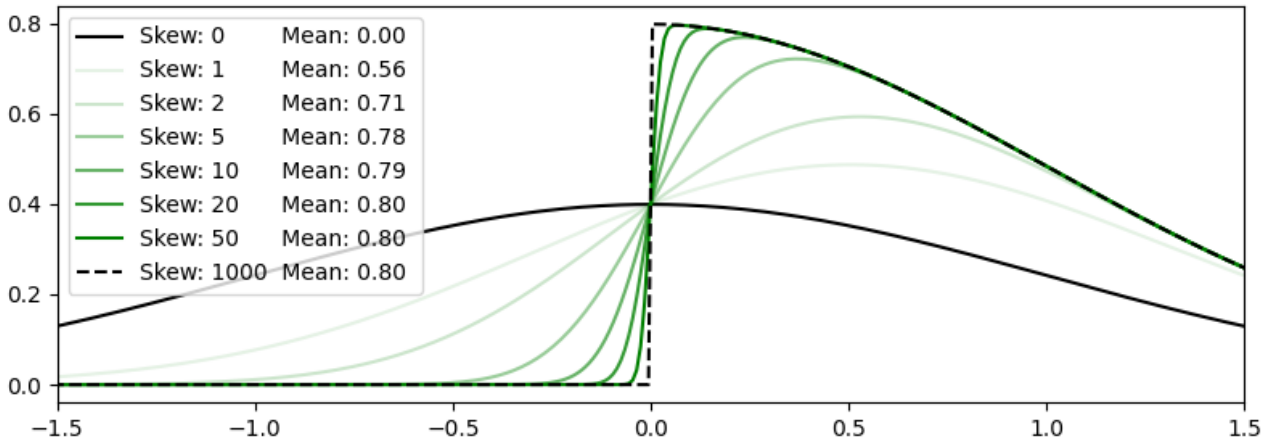
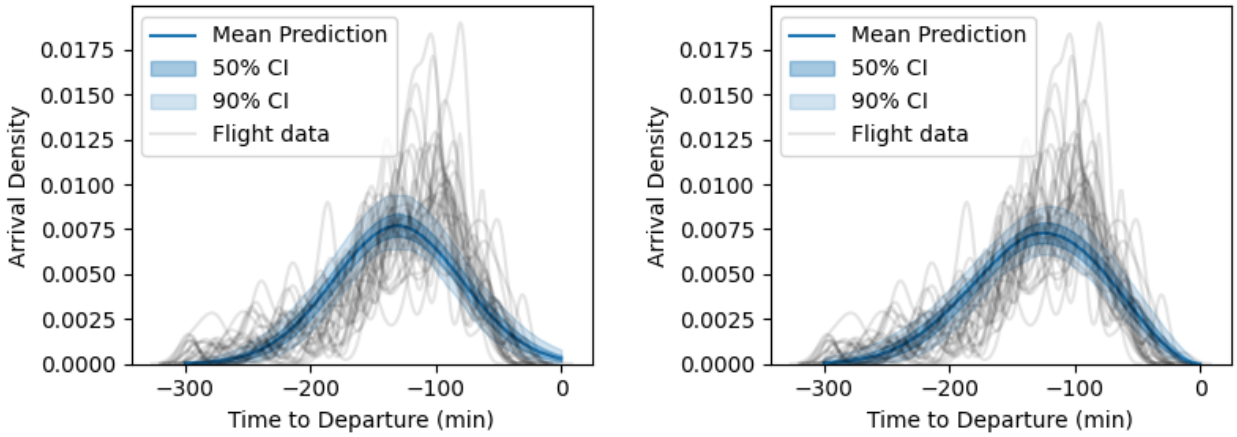


Figure 28: Skew Normal distribution (mean:0, std:1) with varying skew parameter

For completeness Figure 29a and Figure 29b show the fitted TTD model for the Normal and Weibull distributions. For this flight ID, it can be seen that that fixed skew normal distribution captures the underlying average arrival rate the best. However, these plots also exemplify the limitation of using parametric distributions, and their inability to capture large outliers.



(a) Normal

(b) Weibull

Figure 29: TTD distribution comparison for a single flight

D.2 Count model

In this section, a quick overview of the data issues is presented, particularly those concerning the validation of the count distribution model. As discussed in section 6.2 due to issues with the data, a clear training and test sets could not be created. These arose from 2 sources, firstly, which has already been outlined, a large number of days contain missing data. While data from these days has been filtered and cleaned up enough to use for training, due to quality concerns these days have not been included in the test set. These issues persisted from the start of the data set till the end of September. The other issue pertains to an unusually high number of new flights. Figure 30 shows the number of new flight IDs introduced in the data per day. Starting at the end of October there is a surge of never-before-seen flights. On further analysis, these flights are not necessarily new but are seasonal, and due to the limited amount of data available, these winter flights are encountered for the first time. Which further complicates the analysis.

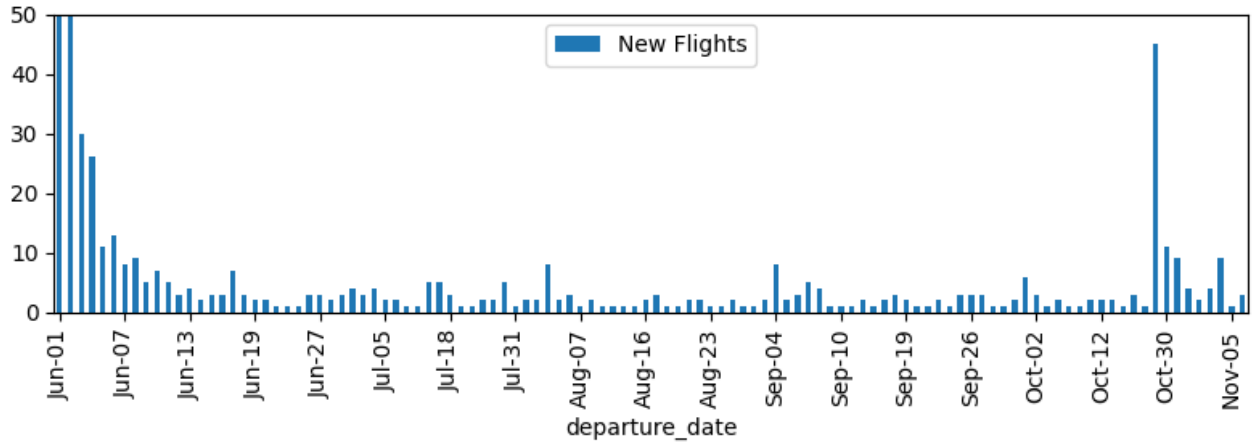


Figure 30: Number of new flight IDs each day

Given another month or two of data, a full analysis could have been carried out with appropriately partitioned training and test sets. While it is expected that this would provide a more reliable and accurate evaluation, due to the relative simplicity of the model no significant differences are anticipated. Although this will be carried out in a non-academic context after the completion of this thesis, with GRASP Innovations. Finally, section 6.2 from the count distribution evaluation shows the distribution of daily differences between the actual and predicted number of passengers. A more detailed overview of that data can be seen in Figure 31 which shows the distribution of the prediction and the observed number.

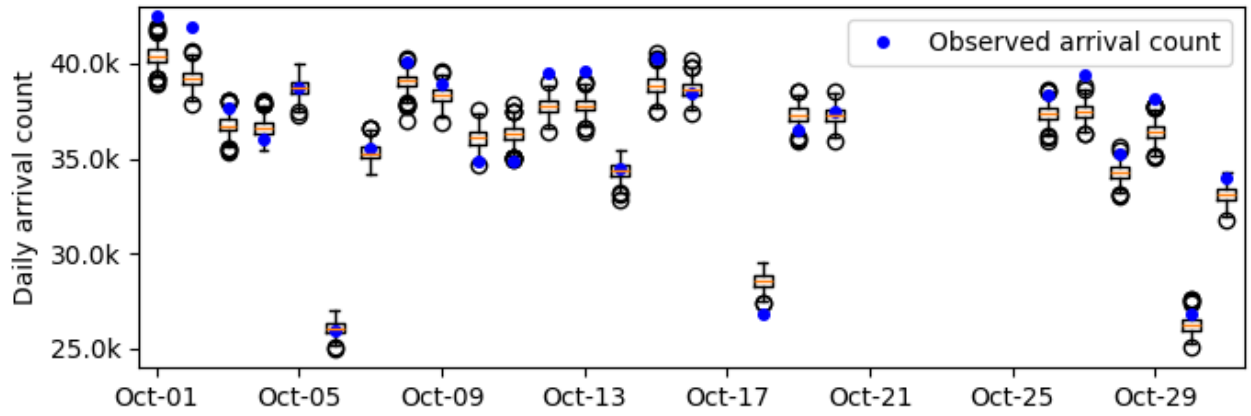


Figure 31: Daily comparison between actual and predicted number of passengers

D.3 Combined model

The following figures show additional forecasts from October 11-16 using the Fixed Skew Normal distribution.

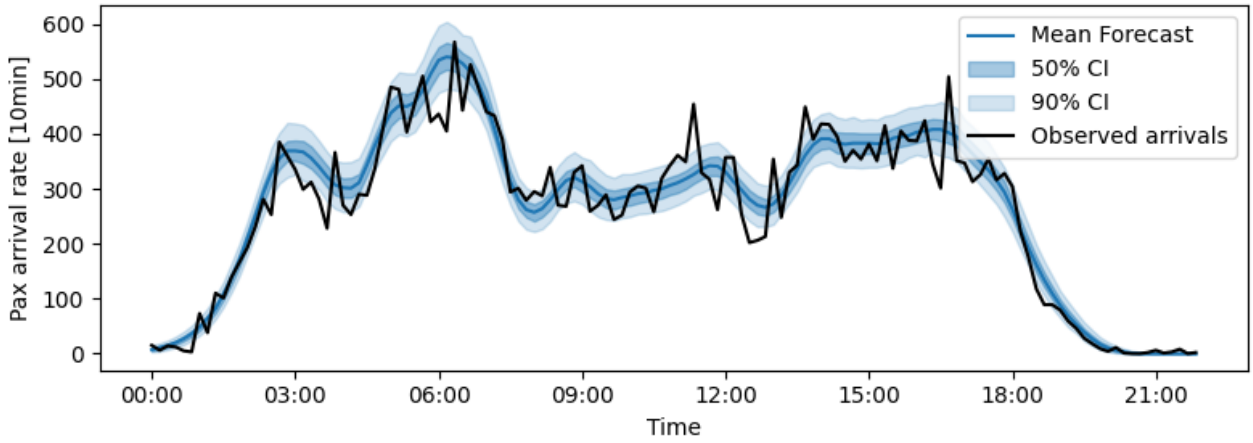


Figure 32: October 11th forecast

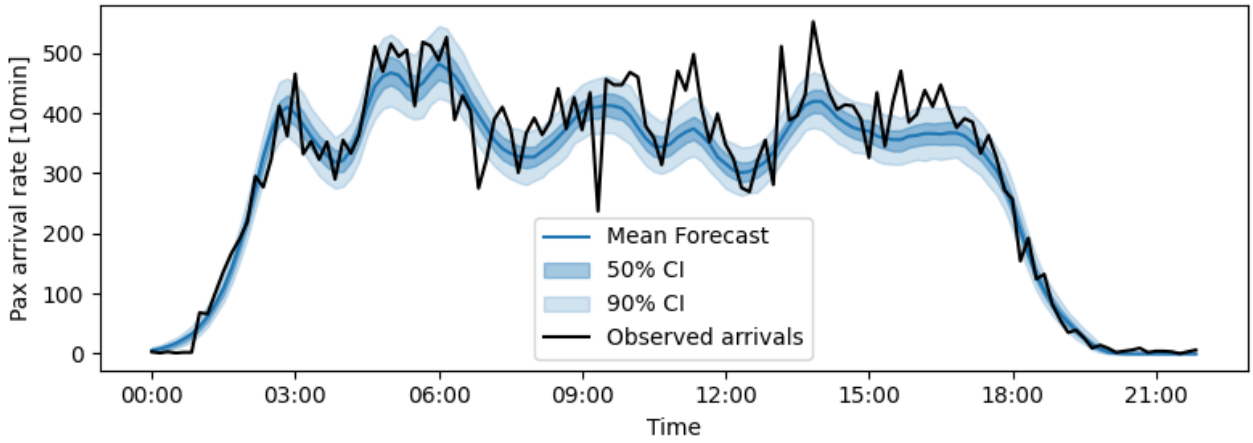


Figure 33: October 12th forecast

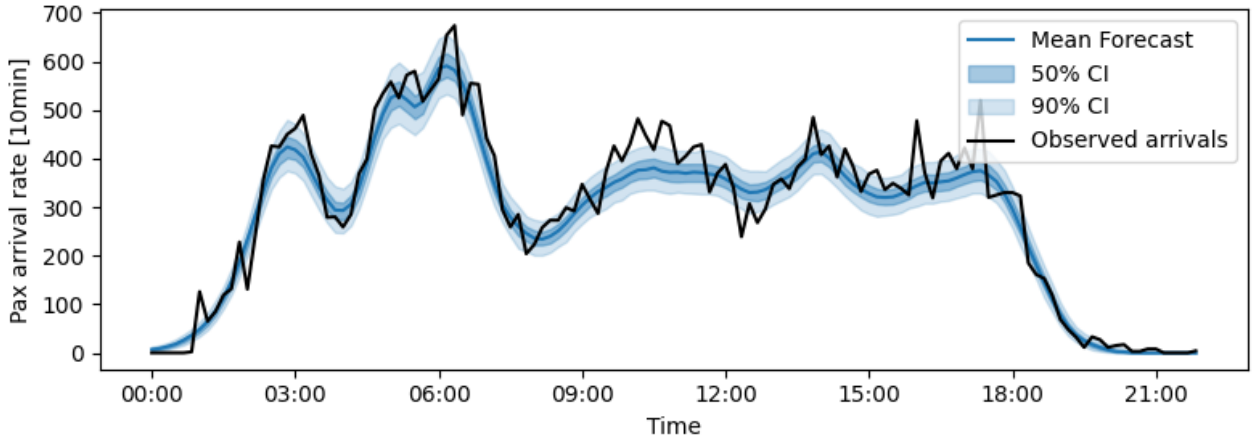


Figure 34: October 13th forecast

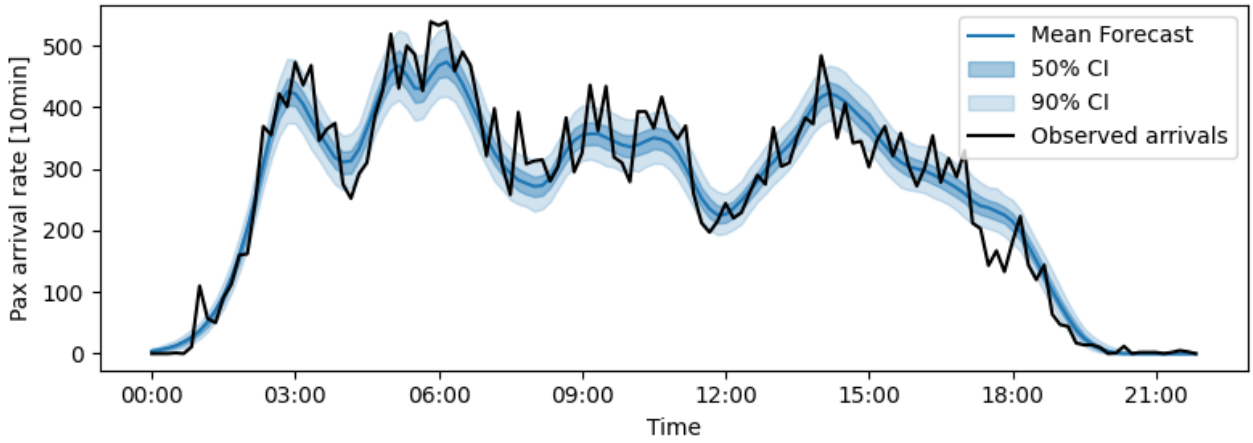


Figure 35: October 14th forecast

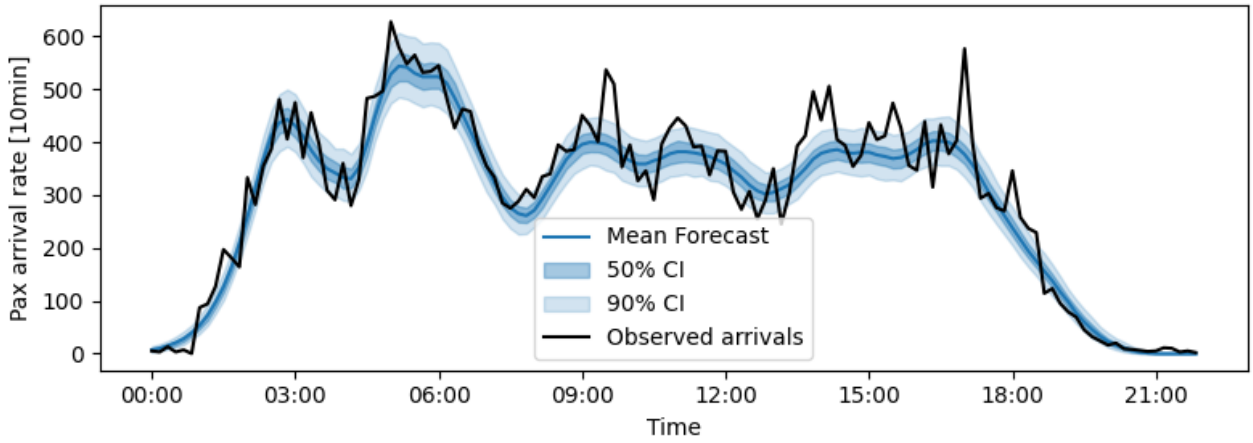


Figure 36: October 15th forecast

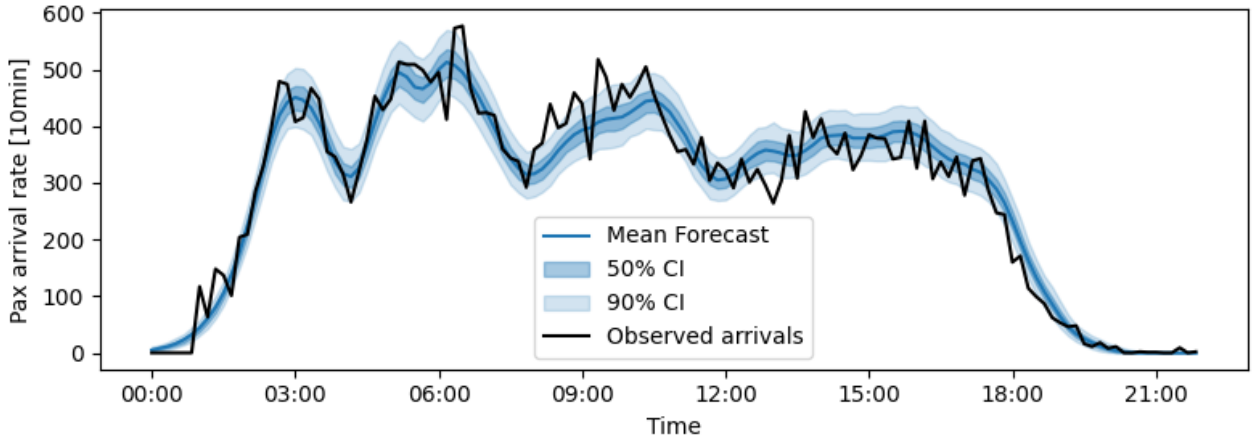


Figure 37: October 16th forecast

II

Literature Study previously graded under AE4020

1

Introduction

Aviation has been one of the most consistently growing industries in the 20 and 21st century, with estimates consistently forecasting at least 4% growth rate [20] [19]. The industry has proven itself to be relatively resilient, even with events such as the September 11th attacks in 2001, and the 2008 financial crisis. Both of which had a significant negative impact on growth and required a number of years before bouncing back to previous levels. However the 2019 COVID pandemic has perhaps been the largest challenge to aviation, putting an incredible strain on the industry as a whole. During initial lock downs traffic was down between 75% and 95% depending on the country, and even by September 2021 average traffic amount is only back to 70% of pre-pandemic levels [17]. Optimistic scenarios forecast reaching 2019 levels of air traffic by 2023-24, and more conservative ones put this date past 2027 [17].

This reduction in traffic has had a devastating effect on airlines, and especially on airports. There have been a number of initiatives from airports to increase their revenue, of which around 40% comes from non-aeronautical aspects, such as, parking, advertisements, and shopping [22]. However even before the pandemic these sources of revenue have been decreasing with the adoption of ride sharing apps, and more stream-lined consumer experience such as online check in, reducing time spent at the terminals. Therefore in order to keep airports running and profitable, operating costs need be reduced. Security checkpoints have been identified as one such area, as currently security checkpoints represent little over a quarter of all operational costs to airports [33].

One approach to improve the operational efficiency of the checkpoints is by gaining better insight into their operations. This is why this research has be done in collaboration with GRASP Innovations who aims to do technology-driven data collection and aggregation to offer clear insights for optimising resource and infrastructure utilisation. One of the challenges related to infrastructure utilisation being the scheduling problem, the need to determine the required number of security lanes and therefore security agents. Both on the long term, as well as on the short term, such as break management. To this end this research will focus on development of a novel probabilistic forecasting algorithm that will allow for a robust decision support system to help with management of checkpoint resources on the short term.

In order to achieve this, a comprehensive review of literature has been completed. Firstly chapter 2 will present an brief overview of the history of airports, the impact of security checkpoints and outline the motivation behind this work. Following this and exploration of previous passenger arrival forecasting techniques will be completed in chapter 3. Using the insights gained chapter 4 will review state of the art forecasting methods, and identify a suitable candidate method. chapter 5 then discusses ways to evaluate the output of the forecasting method, to identify a KPI that can be utilised for comparison. Combining all the insights from previous chapters, the research problem and research question will be formulated and presented in chapter 6. And then the paper will be concluded by presenting a simple case study to which the forecasting model can be applied, as well as the planning of the thesis in chapter 7.

2

Background and Motivation

In this chapter some background information and motivation is presented to aid with better understanding of the subsequent research. In section 2.1 a brief overview of airport history is presented with a focus on events and trends that affected their evolution. Then security checkpoints are discussed as the primary bottleneck in airports, and efforts to optimise them are elaborated in section 2.2. This is followed by section 2.3 where the non academic stake holders are introduced, and their goals outlined. Then the data that will be made available is discussed in section 2.4. And the chapter is concluded by formalising the requirements and KPI's that have been discussed with the company in section 2.5.

2.1. A cursory history of airports

Throughout aviation's relatively brief history, airports have served as gateways between land and air. However, only in recent decades, with the emergence of long-range aircraft and globalisation, have airports evolved into what we recognise today. With over 100 airports accommodating at least 10 million passengers annually, and around a dozen managing 50 million or more, contemporary airports must efficiently process vast numbers of passengers [3].

In aviation's early days, airports were simple structures, consisting of hangars and halls for cargo and passengers. The infrastructure constantly adapted to accommodate the evolving needs of aircraft. Significant developments occurred during World War II due to the emphasis on aerial supremacy, which required more robust infrastructure. The post-war era led to increased civilian use and the emergence of modern terminals. The 1960s saw another shift with the introduction of commercial jet airliners, prompting a boom in airport construction. Eventually, infrastructure began driving aircraft design rather than vice versa, as evidenced by size restrictions influencing designs like the A380 and Boeing 777x, which are constrained by their respective size categories.

Airside infrastructure was significantly influenced by aircraft, while political and economic factors primarily drove landside terminal design. Kazda and Caves pinpointed five prominent factors: the threat of terrorism and unlawful acts, privatisation, deregulation of air transport, the growth of low-cost carriers, and the increasing environmental impact of aviation [31]. The tragedy of 9/11 marked a crucial shift in airport infrastructure, prompting the largest changes to security checkpoint procedures ever implemented. This event emphasised the importance of addressing terrorism threats, which have become a highly visible and often unpleasant aspect of air travel for passengers.

2.2. Security Checkpoints

The primary purpose of an airport terminal is to serve as a gateway between the landside and the airside, with security procedures forming the interface between these two areas [31]. The landside is freely accessible to everyone, while the airside can only be accessed by passengers (and employees) after passing through the security checkpoint. Even before the 9/11 era, security checkpoints naturally created a bottleneck in passenger flow, as all passengers had to pass through them. However, post 9/11, due to significantly heightened scrutiny, throughput rates drastically decreased. In the US, rates fell from 500-600 passengers per hour per lane to 100-150 passengers per hour per lane, exacerbating the bottleneck's severity [12].

This has placed significant pressure on airports, as recent estimates attribute up to one quarter of an airport's operational expense to security [33]. As a result, airports attempt to maintain the minimum number of open lanes necessary to meet throughput requirements; however, this often leads to overcrowding, with 70% of passengers reporting such feelings [60]. Consequently, security checkpoints have become a key focus for improvement in both industry and academia within airport terminals. Recent works emphasises "Airport 4.0" technologies, a term analogous to industry 4.0, aimed at creating "Cyber-Physical Systems" [59]. These technologies either directly enhance operational efficiency or indirectly do so by collecting data for further analysis.

In the realm of security checkpoints, two general categories of approaches exist to enhance system efficiency. The first approach investigates the security checkpoint lane, utilising advanced simulation techniques, such as agent-based simulation, to optimise lane configuration [60]. Alternatively, data from the lanes can be collected to analytically pinpoint bottlenecks in the process and suggest new configurations or procedures [62]. These strategies seek to augment the security checkpoint lane's throughput by implementing fixed-cost infrastructure upgrades, which allow for increased throughput with the same operational resources, thereby enhancing efficiency. The second category of optimisations seeks to minimise "surplus" throughput at the checkpoint, enhancing efficiency by reducing idle lanes and personnel. This can be achieved at the lane level, where security agents are dynamically assigned to various lane areas to maximise throughput [36]. However, this reactive approach has limited upside. Alternatively, checkpoint-level optimisation can be employed, wherein the number of open lanes are adjusted based on a future arrival rate by solving a task allocation problem [26]. The drawback of this method lies in its reliance on the future arrival rates, as well as the relatively higher risk of either having too much or too little throughput capacity. Although the benefits of quicker implementation, diminished capital requirements, and lesser risk, means that operational strategies may be preferred over capital-intensive approaches in improving security checkpoint effectiveness.

2.3. Motivation - GRASP

The subsequent research was conducted in partnership with GRASP Innovations, whose primary objective is to employ technology-driven data collection and aggregation to offer clear insights for optimising resource and infrastructure utilisation. Currently the main emphasis lies on airport security checkpoints, particularly on gaining insight into the performance of security checkpoint lanes using IR-UWB radars that detect passenger presence at each crucial location. Moreover, a large European airport has collaborated with GRASP to develop value-adding functionalities, thereby granting access to a large amount of data for both GRASP and this thesis project. With the ultimate goal of increasing efficiency of the security checkpoint.

Recently, this airport has undergone a large-scale infrastructure upgrade aimed at increasing the throughput of each individual lane. Although minor upgrades and adjustments are planned, no significant changes are anticipated in the foreseeable future. Consequently, the current focus is on operational improvements, aligning with GRASP's ambitions and expertise. One of the challenges faced by the airport concerns optimal staffing decisions at the checkpoint across various time scales. On the tactical time frame, the required number of security personnel must be determined for a specific day, while short-term decisions at the operational level involve deciding when to send security agents on breaks. An additional infrequent, yet intriguing, operational decision entails shifting personnel between security checkpoints, where checkpoints experiencing low demand can transfer staff to other checkpoints that might be understaffed. By addressing these challenges, the airport aims to enhance overall efficiency and provide a seamless experience for travellers.

At present, the airport generates a weekly point forecast using a relatively simplistic regression model based on the previous three weeks of data, yielding a rough 30-minute bucketed forecast for arrivals. This suffices for tactical decisions, such as determining the necessary number of security agents. However, this forecast is poorly suited for operational decisions, mainly due to its coarse nature and inability to account for the dynamic and ever-changing information landscape. This is especially significant given the wealth of sensor data available, from GRASP radars to boarding card readers. Break management and personnel shifting between checkpoints are currently overseen by the checkpoint coordinator, who relies on experience to make decisions. Consequently, GRASP aims to bridge this gap by offering advanced information aggregation to generate operational level forecasts for the number of arriving passengers at each security checkpoint. This enhanced operational forecast will not only facilitate improved decision-making for existing procedures but also pave the way for the potential implementation of novel operational strategies.

2.4. The available data

As mentioned in the previous section the large European airport collaborating with GRASP has provided access to operational data. Specifically boarding-card data, which is timestamped event data of the scanning of each passengers boarding-card before entering the security checkpoint queue. This data is continually being collected for a single checkpoint at the airport, and therefore the exact number of datapoints available at model training is unknown. However roughly 2 million individual passengers, and over 12000 flights are expected to be available. Each data entry contains the timestamp, and the flight ID of the passenger entering the security checkpoint. Given the flight ID, it is possible to use publicly available data sources in the form of flight schedules, and flight data to get the additional following features for each flight shown in Table 2.1.

Feature	Data type	Comments
Departure time	Timestamp	Contains other temporal features such as holidays
Airline	Categorical	~100 unique airlines
Destination airport/country	Categorical	~100 unique airports, ~30 unique countries
Distance to destination	Numerical	Continuous
Aircraft capacity	Numerical	Discrete

Table 2.1: Available raw data description

Additional information can be extracted from each of the above features, for example temporal information such as the weekday, holiday or season can be turned into their own features. Finally the available data is quite sparse, which is related to the Pareto principle, also know as the 80/20 rule. The data contains a large number of airlines, but a minority of them make up the majority of flights, similarly a small number of locations make up the majority of destinations. Therefore there are sparse regions of the feature space, which further complicate modelling it.

2.5. Requirements and KPI

Before delving into previous literature on this topic, there are a few requirements and key performance indicators (KPI's) defined by both GRASP and the airport that will drive modelling decisions. While the airport will be the final consumer the forecasting system, the primary goal is to create a generalised model for GRASP that will allow for providing decision support systems that will leverage the operational forecast. Understanding these requirements and KPIs will serve as a foundation for the research, while the gap in the existing literature and industry practices will be addressed in the subsequent section.

The requirements set by GRASP are:

RQ 1 The forecasting algorithm shall be capable of delivering high-frequency forecasts with at least a minimum granularity of five-minute intervals.

RQ 2 The forecasting algorithm shall provide the capability to adapt the output interval sizes (buckets) as required.

RQ 3 The forecasting algorithm shall be able to leverage real-time data inputs for updating forecasts in near real-time.

RQ 4 The forecasting algorithm shall be able to quantify uncertainty in it's output.

The KPIs are as follows:

- Long-term Total Passenger Accuracy: There shall be a measure of the accuracy of the forecasting algorithm in predicting the total number of passengers over a long-term period. It shall focus on the aggregate passenger count, quantifying the deviation between the predicted values and the actual observed values, emphasising overall passenger volume accuracy.
- Short-term Fluctuation Detection: There shall be a measure of the algorithm's accuracy in identifying and predicting passenger count fluctuations within short-term forecasts, capturing both peak surges and periods of idleness for optimal personnel assignment and break scheduling.

For the partnering European airport, the paramount concern pertains to the management of queue lengths at security checkpoints. The airport's principal objective is to guarantee that the queue length remains within the established performance constraints. In the event that the queue length exceeds the designated threshold, the airport aims to minimise the duration for which it remains above the limit, thus ensuring efficient operations and enhanced passenger experience. Given that these constraints are met, the airport's secondary goal is to minimise operational costs, which are primarily driven by labour expenses.

These requirements and KPIs reflect the priorities of both GRASP and the airport, providing a framework for addressing the challenges faced in optimising security checkpoint efficiency. The subsequent section will delve into the research gap and outline how this thesis aims to contribute to both academic literature and industry practices.

3

Existing Passenger Forecasting approaches

With the ever increasing complexity of both airports and their associated models, the importance of forecasting passenger arrivals cannot be overstated. Among the four primary airport simulation models identified, capacity, operational planning, security and airport performance, forecasting serves as a critical component of capacity and operational planning models [63]. Capacity planning generally utilises coarse strategic-level forecasting, concentrating on long-term trends, with the output typically applied to long-term decision-making related to infrastructure. While operational planning employs operational or tactical-level forecasting to address short-term and medium-term demands, which are used for resource allocation purposes. Provided that these models have access to reliable and accurate forecasts they have the ability to significant impact overall airport efficiency, resource utilisation, and passenger experience.

In their assessment of airport passenger throughput models, [11] classified forecasting models into four approaches: time series models, causal models, artificial intelligence models, and hybrid models. Additionally, two other methods were identified: analogy-based methods, which draw comparisons to other airports with similar initial states, and market share methods that utilise aviation market forecasts to determine expected proportions. However, these latter two methods will not be explored in this section, as their inherent limitations in technical rigour, adaptability to novel situations, and dependence on subjective expert judgement render them less suitable for a thorough, data-driven forecasting analysis. Consequently, the following section will adhere to the structure from [11], concentrating on the four primary forecasting models. The structure comprises time series models discussed in section 3.1, causal models in section 3.2, machine learning models in section 3.3, and hybrid models section 3.4, succeeded by a concluding section that will identify trends in the following papers and outline the research gap in section 3.5.

3.1. Time series

Time series forecasting techniques comprise a wide array of statistical methods designed to predict future values using historical data. These approaches span from basic moving average models to intricate ARIMA and GARCH models. Time series techniques are essential in numerous domains, such as finance, economics, energy, and healthcare, facilitating data-driven decision-making and resource optimisation. Although their popularity has diminished since their inception, particularly in passenger forecasting, these techniques still have valuable use cases. Their enduring relevance can be attributed to the relatively straightforward implementation, abundant technical applications, and robust, predictable properties.

[30] employed Dynamic Tobit models and Generalised Autoregressive Conditional Heteroskedasticity (GARCH) errors to forecast monthly arrivals of domestic and international passengers at Corfu Airport in Greece. The combined model utilised 20 years of time series data, incorporating variables such as the number of arrivals, European GDP per capita, Greek GDP per capita, and disposable income. The paper's primary contribution lies in the application of GARCH errors, which effectively captures the time dependent variability caused by the highly seasonal demand experienced by holiday destinations like Greece. Additionally, the Tobit models allows for handling censored data, which arises during holiday periods when demand approaches airport capacity. Although the paper sufficiently investigates the model's parameters, which primarily focuses on capturing highly seasonal demand with censoring, it lacks a comprehensive analysis of the forecasting model's overall performance. Even so through GARCH extension of the time series model, the

paper successfully quantifies uncertainty of the monthly passenger arrivals.

[39] employs a SARIMA model, a time series approach that combines autoregressive, moving average, and seasonal components to predict arrivals at the security checkpoint. The model exhibits satisfactory performance for predicting total daily passengers, with moderate fine-grained results, but its validation is confined to a small dataset consisting of only four days. Moreover, the forecasting time horizon for the validation is not specified, casting doubt on the promising short-term performance. If the model was utilised to forecast only one time point ahead of real data, its performance would be considerably less remarkable than if it predicted an entire day. However, a key advantage of this model is its capacity to incorporate real-time data for improving short-term predictions. The paper highlights an important limitation: this method is inherently incapable of utilising flight schedule information, which carries significant value for refining arrival rate estimates. This is particularly crucial for short-term prediction problems, where arrivals are highly stochastic but adhere to flight schedules.

On the other hand, [1] focused on forecasting monthly passenger arrivals on a longer time horizon by employing methods such as moving average (MA), single exponential smoothing method (SESM), Holt method (HM), and Holt-Winter method (HWM). The studies objective was to estimate the required daily shuttle service levels, using relatively simple time series techniques as input to enhance operational and strategic level resource allocation. Although the paper did not yield exceptional results, it raised an intriguing point by attempting to create interval forecasts. This implementation was quite rudimentary, since the forecasts output was monthly, individual daily demand was estimated using a fixed error margin. However, unlike other discussed methods that produced point forecasts, the approach sought to capture additional information, emphasising the potential value of interval forecasting. This additional information, if well implemented, is potentially quite valuable, as understanding the model's inherent uncertainty or the stochasticity of the phenomena can significantly enhance decision-making processes.

While time series approaches can achieve reasonable accuracy with a relatively simple model, they do present several drawbacks, as partly demonstrated in the aforementioned papers. First, time series methods suffer from error accumulation, as each new forecasted data point relies on previous data points, resulting in predictions based on predicted values. Moreover, incorporating additional external information into forecasts is not a trivial task, despite potentially containing valuable information – particularly relevant to passenger forecasting, which heavily depends on flight schedules. Lastly, traditional time series methods are fundamentally linear models, necessitating supplementary approaches to introduce non-linearity into the forecast. Despite these challenges, time series approaches remain an intriguing avenue due to their innate capacity to incorporate real-time data and their ease of implementation, making them suitable for integration with other models.

3.2. Causal models

Some of the previously mentioned limitations of time series modelling can be addressed by employing causal models, which explicitly define causal relationships between independent variables. These methods provide the benefit of integrating external information and provide capabilities to capture non-linear relationships, thus offering a more thorough depiction of the underlying process. Moreover, the accuracy of their predictions remains considerably more stable across different time horizons, as long as high-quality input variables are utilised.

[57] developed a complex system to predict future runway and terminal capacity requirements, incorporating variables such as GDP growth, population growth, airline costs, and daily flight numbers. The primary advantage of this approach is the transparency of variable interactions, which facilitates scenario and sensitivity analyses. However, this method also presents significant drawbacks, as it necessitates a thorough understanding of the system, and fine-tuning of variable parameters, either through forecasting or expert input. The complex model utilised can be seen in Figure 3.1 The substantial time and expertise required for these approaches have contributed to their waning popularity. This is particularly evident in the context of long-term forecasting, where the uncertainty of input variables over extended time horizons disproportionately reduces the model's accuracy.

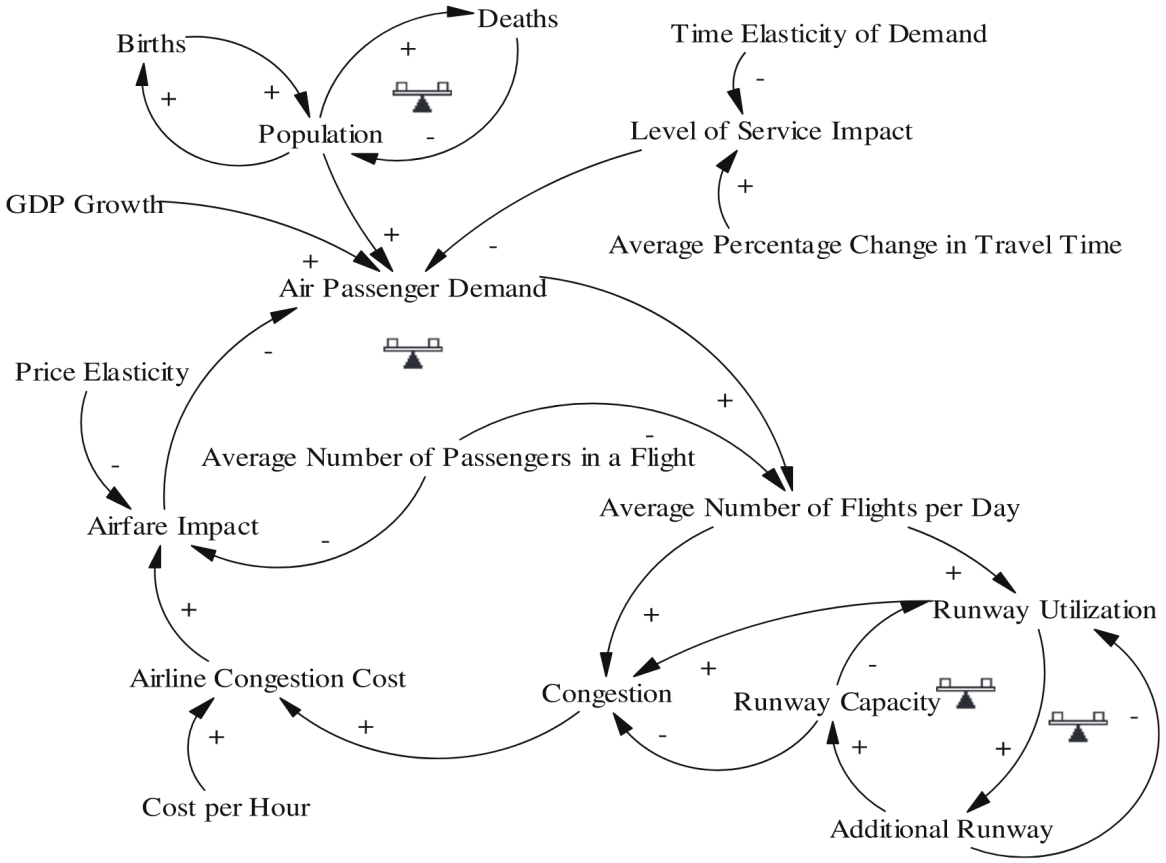


Figure 3.1: System dynamics relationship graph for forecasting air passenger demand [57]

[25] employs a similar yet simpler causal model that utilises Bayesian approaches to quantify the uncertainty of both the input features and the estimated air traffic demand. With 20 years of historical data, the model trains three features; number of passengers, average load factor, and average seats per aircraft using Gaussian process (GP) regression. The importance of feature selection is emphasised in the paper, highlighting that Bayesian approaches become prohibitively expensive with high number of features. Despite this acknowledgement, the paper stops short of justifying their selection. Once the GP is trained, future values are forecasted and, and then using Monte Carlo Markov Chain (MCMC) sampling, a distribution for future air traffic demand is created. Though considerably simpler than [57], the inclusion of confidence intervals provides a substantial degree of assurance for decision-making purposes. The utilised approach demonstrates significant value in the given context, however its applicability may be limited in other scenarios, especially finer forecasts where GP is less well suited at capturing seasonality. Nonetheless, its ability to include confidence intervals adds an intriguing and valuable dimension to the modelling process.

Though causal models are becoming less prevalent for long-term predictions, high frequency and quality data enables the generation of short-term forecasts using these approaches. For instance, [50] utilises boarding card data to estimate individual passengers Time To Departure (TTD) arrival distributions for individual flights, which are then combined to determine the overall short-term arrival rate at a checkpoint. The study found that the Weibull distribution provides the best fit to the TTD arrival distribution from among Gaussian, Poisson, Gamma, and Lognormal distributions. However, the goodness of fit for the Weibull distribution is not thoroughly investigated. Additionally, the distribution fitting is empirically conducted on four groups, combining low-cost and full-cost airlines with early and late departure times, as can be seen in Figure 3.2. While recognising the considerable impact of both carrier type and departure time on TTD distributions, the applied approach is relatively simplistic and mainly serves as a demonstration of utilising probability distributions. [52] adopted the same methodology, employing empirical fitting of Weibull distributions to TTD data to estimate security checkpoint queue behaviour, using the forecast as input. Consequently, this approach was even simpler compared to [50], fitting only two different TTD distributions for flights categorised

as Schengen or non-Schengen. Ultimately, these studies demonstrate the potential for utilising causal models and probability distributions in estimating short-term forecasts, but also highlight the need for further investigation to their applicability.

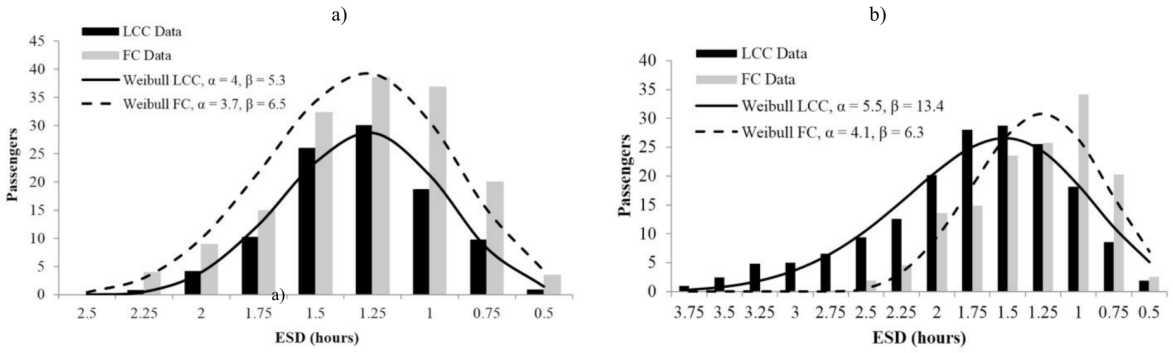


Figure 3.2: Time to departure plots for Low Cost Carriers (LCC) and Full Cost Carriers (FCC) for a) morning, and b) afternoon [50]

[40] introduces an innovative approach to forecasting arrival rates, which can be viewed as a bottom-up, system dynamics-based method. This technique employs historical data to estimate dwell times in the three primary airport areas: check-in hall, security area, and departure hall. A gamma distribution is fitted to represent the probability of the duration a passenger spends in each section of the airport. By tracing back from scheduled flight departures and the estimated number of passengers, it becomes possible to estimate the arrival rate in each airport area. While this approach is inventive, it necessitates homogenising all passengers, with no distinction made for passengers on different flights. Furthermore, the dwell time distributions are assumed to be static, without any variation throughout the day. Despite the novelty of this approach, it does not fully capitalise on a range of important features related to each individual flight that could potentially enhance the accuracy of the forecast.

In conclusion, while causal models present opportunities for improving short-term forecasting by incorporating external information and capturing non-linear relationships, their limitations, such as the need for high-quality input variables, their selection and extensive fine-tuning, must be addressed. TTD distribution estimation appears particularly promising, as it can capture known structured information about the future, potentially allowing for significantly higher accuracy models.

3.3. ML models

In recent years, machine learning (ML) approaches and models have emerged as powerful tools for forecasting across various domains, which of course includes passenger arrival predictions at airports. These advanced techniques offer the ability to learn complex patterns and relationships from within data. And as the volume and quality of available data keeps increasing, the integration of ML approaches are becoming more and more prevalent.

[37] employs a decision tree to estimate the number of passengers for each individual flight by predicting the load factor, which is then used with the available capacity of each aircraft. This is employed for both short and long-term operational planning at the airport involved in this study. Notable features were identified, with particularly significant ones including destination, day of the week, and month of the year. And the developed model performed quite well, with a root mean square error of 3-12% over a month of validation data. While obtaining a precise arrival rate at the security checkpoint is not feasible with just this data, several methods, such as the aforementioned TTD distribution estimation approaches, require passenger number estimates per flight to build up a full arrival pattern.

Although [41] does not focus on airports, it presents a relevant and innovative approach for predicting hourly bus terminal arrivals based on schedules, which bears a reasonable resemblance to security checkpoint arrivals. The study employs a novel combination of autoencoders and deep neural networks (DNNs). Autoencoders are trained on historical data to extract features from schedules, which then serve as a pre-trained basis for the DNN. The paper also provides a comprehensive investigation into the features utilised by the network and identifies holidays, hour of the day, and destination as the most influential factors. The model exhibited high accuracy, however given the nature of the problem it addresses combined with the

coarse forecasting interval, it essentially boils down to predicting the number of passengers for each bus.

In [47], an LSTM is employed to predict the arrival rate at each security checkpoint at Charles De Gaulle Airport using flight schedules and anticipated passenger counts. The study identifies numerous features, including airline, aircraft type, destination, month of the year, day of the month, day of the week, hour of the day, and categorical features such as holidays, weekends, and days before and after holidays. The primary limitation of the implemented LSTM model is its tendency to systematically underestimate the arrival rate as well as the fundamentally difficulty to train the data hungry architecture. There are two notable observations in this paper: first, the forecasting performance decreased at certain checkpoints when the hour of the day was incorporated as a feature. This seems to go against findings of most other papers, and intuitively seems odd that the time of day would decrease the models predictive power. Second, while the LSTM was trained solely on security checkpoint arrivals, and flight schedules, the model had to generate internal implicit representations for TTD distributions per flight.

Machine learning (ML) techniques have shown great potential in enhancing passenger arrival predictions at airports. However, a major limitation and issue with these models is the "black box problem" – the lack of interpretability and transparency in these methods, which makes it difficult to rely on them and make confident decisions with their outputs. This in combination with the difficulty of encoding domain knowledge necessitates increased model complexity which increases the required data along with it. This has lead to a shift from using "pure" ML methods, to methods that leverage ML techniques for sub parts of a problem.

3.4. Hybrid models

As demonstrated in the previous section the use of ML models has been on the rise, however not without their downsides. Recent developments combine ML methods with more traditional approaches, resulting in hybrid models. The advantage over a purely ML approach is that problem aspects that can be easily modelled with more robust and traceable methods need not be captured by the ML algorithm, thus reducing the required model's complexity. These approaches typically employ traditional models that excel in linear problems, while ML algorithms enable the capturing of non-linearity's.

[56] integrated two traditional linear time series approaches, Time Series Regression (TSR) and Autoregressive Integrated Moving Average (ARIMA), with two non-linear machine learning methods, neural networks (NN) and support vector regression (SVM). With the goal to predict long term passenger throughput of an airport using a number of additional temporal features, such as the month of year, week of year, and week and month of major holidays. In this model, time series approaches forecast the next time step, while non-linear methods are trained on the errors of these forecasts. The inclusion of non-linear ML steps improved forecasting performance by up to 36% when using mean absolute percentage errors for the combination of ARIMA and NN.

Similarly, [26] employs a two-phase approach to forecast the arrival rate at a security checkpoint. Instead of a time series forecast, the first phase utilises a causal model which predicts the number of passengers per flight based on the flight schedule, and other external features. This is then combined with a static Time To Departure (TTD) distribution for each flight. The second phase involves adjusting the initial estimate using historical data, partitioned into checkpoint-day-hour combinations, where a coefficient is "learned" for each specific combination. However, the model's performance is lacking, partially due to the assumption of a constant TTD distribution and the simplistic assumption concerning the correction coefficients.

A slightly different field of application is examined in [23], which also employs a two-phased approach and is the second paper to offer more than a mere point forecast. The first phase entails using a regression tree to predict the transfer times between a passenger's landing and their arrival at the immigration desks. A kernel density estimation is fitted to the empirical distribution of each leaf of the regression tree. Various features, including flight origin world region, hour of the day, day of the week, and perceived connection time, were identified and utilised. In the second phase, the arrival distribution of incoming passengers is sampled using passenger attributes form near real-time data, generating a number of quantiles for the expected number of arrivals in each time bucket. This method exhibits substantially tighter intervals and statistically significant improvements compared to the legacy system that the airport employs. This can be partially credited to the integration of near real-time data that describes the characteristics of soon-to-arrive passengers. While the paper claims to use real-time data, this isn't entirely accurate: the attributes of connecting passengers are collected while the passengers are en route, typically 90 minutes before arrival. This, however, underscores the potential of integrating real-time information to enhance short-term forecasting.

[64] builds upon previous causal modelling approaches that utilise probability distribution functions, in-

creasing their ability to express and fit distributions more robustly. This improvement is achieved through the implementation of a Gaussian mixture model, in which a second-order Gaussian was found to adequately capture the underlying behaviour. The model's parameters are fitted using a Radial Basis Function (RBF) neural network, a method particularly well-suited for function approximations. However, the study has some limitations, including a lack of information on individual flights, with only departure time as a feature, and a lack of available data, which spans a small sample of only 15 days. Consequently, the overall forecasted arrival rate at the security checkpoint exhibits relatively poor performance when attempting to predict on fine-grained 10-minute intervals. Nonetheless, the paper presents an intriguing finding that neighbouring flights' TTD distributions influence each other.

In conclusion, the increasing adoption of ML models, despite their drawbacks, has spurred advancements in hybrid approaches that blend traditional methods with machine learning techniques. The above studies demonstrate the potential for improved forecasting by combining more traditional models with non-linear machine learning methods, although certain limitations remain. The ongoing research in this area highlights the evolving nature of predictive modelling, with hybrid models harnessing the strengths of both traditional and ML methods to address complex problems more effectively.

3.5. Discussion

The ever increasing demand being placed on airports, coupled with their intrinsic drive to improve efficiency, reduce costs, and enhance passenger experience, makes the ability to accurately forecast passenger demand indispensable. This chapter has explored numerous methods, classified into four categories: time series, causal, machine learning, and hybrid models. Each employs a unique forecasting approach, yet certain trends and observations can be drawn across them. Notably, the most consistent observation is that the characteristics of the data available significantly influences design decisions, such as the selection of method, features, and forecasting frequency.

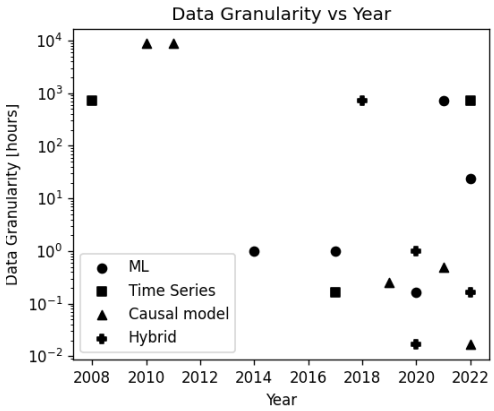
Four high level data related trends were identified from the available studies, that are mostly approach independent.

- **Increasing granularity of data** The data granularity that the models have been trained on has been increasing. Recent papers, such as [50] and [47] harness boarding card reader data, offering event-based data. This trend can be seen in Figure 3.3a.

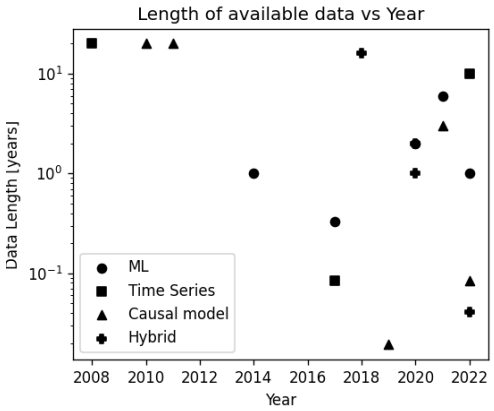
- **Shorter historical data availability** The shift towards greater data granularity, driven by advances in technology, has inadvertently led to a reduction in the span of available historical data, as depicted in Figure 3.3b. This is partially because such high-resolution data has only started to be collected, and made available recently.

- **Increasing number of data points** Despite the somewhat diminishing data horizon, the increase in data granularity overall has still led to a rising trend in the number of available data points. As seen in Figure 3.3c.

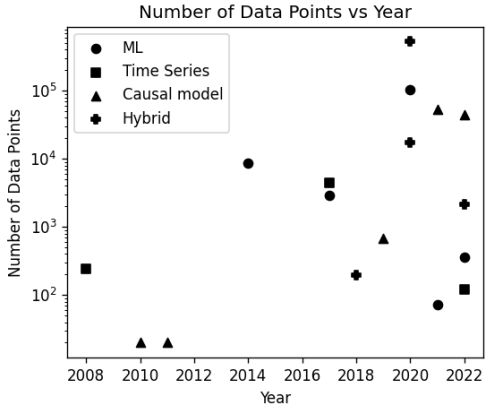
- **Prediction granularity and data granularity highly related** There's almost a 1 to 1 relationship between data and prediction granularity, with two outliers. Methods fitting a continuous distribution to the data can sample it at virtually infinite time points - these methods, either fully or partially causal models, appear as horizontal points on the left of Figure 3.3d. The other outlier, [1], attempts to interpolate monthly data to daily granularity by using confidence intervals.



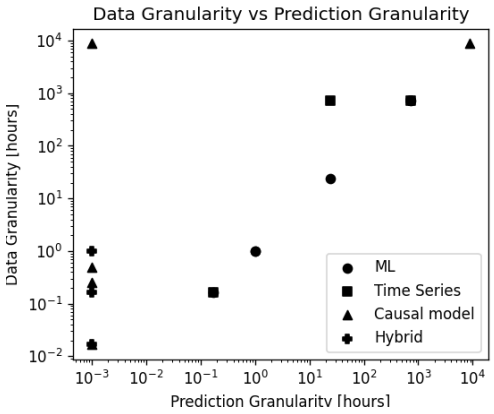
(a) Available data granularity vs year for reviewed papers



(b) Length of available data vs year for reviewed papers



(c) Number of available data points vs year for reviewed papers



(d) Available data granularity vs prediction granuality for reviewed papers

Figure 3.3: Data related trends observed in previous passenger arrival forecasting approaches

In addition to the general trends concerning available data, several other trends have emerged that are more specific to individual studies; their methodologies and approaches:

- **No significant trend in methods used over time** From the figures above, there are no distinct trends in terms of which methods have been applied to the forecasting problems over time. Time series and causal models have a longer history of application, while machine learning approaches emerged somewhat later as adequate amounts of data became available. Notably, hybrid models, which typically combine ML methods with time series or causal models, have only just recently been applied. With the most recent research utilising all four categories.
- **Feature selection often emphasised as important, but this is not fully reflected in the studies** Only [41] and [47] provide detailed breakdown of the importance and performance impact of features. Many studies seemingly utilise all available features or do not elaborate on their selection process. However as demonstrated in [47] certain features may actually degrade performance.
- **Recent trends in increased forecasting granularity has a large impact in how the problems need to be approached** With the increase in data frequency, forecasting frequency was able to rise along with it. This has led to changes in what is being forecasted. Coarse forecasts generally predict the number of people arriving at an airport or checkpoint within a larger timeframe. When the required granularity falls below approximately 30 minutes, the focus seems to shift towards predicting the arrival rate of passengers for individual flights, which are then aggregated to obtain predictions at the checkpoint and airport levels.
- **Forecasting problem is being decomposed** The shift towards more granular forecasting via individual flight predictions has led to a growing trend of studies addressing only a portion of the overall problem.

Several studies, including [50], [64], and [40], focus on predicting the shape of the TTD distribution, but they require external prediction of the number of passengers per flight to construct a complete arrival pattern at the checkpoint. Conversely, [37] forecasts the expected number of passengers per flight, but to generate short-term, high-granularity arrival patterns, it would need an additional prediction of the TTD distribution.

- **Increased reliance on flight schedules** As a consequence of the above two points, the highly valuable information from flight schedules is utilised by more and more methods. This bottom-up approach combines valuable external information with domain understanding to enhance forecasting performance. This is particularly evident in the increased emphasis on estimating the TTD distribution for individual flights.

- **Point forecasts dominate despite a highly stochastic environment** Almost all forecasts in the passenger arrival domain are point forecasts, with only three exceptions: one concerning interval forecasts of connection time of passengers [23]. The other two produce full density, long term, low granularity forecasts for total airport passenger demand [30], [25]. None of which are suitable for short term, high granularity passenger arrival forecasts. Given the inherent unpredictability of the arrival process and the number of passengers on a flight, this appears to be a significant gap in the research.

- **Underutilisation of real-time information integration** Among the studies, only [23] makes use of real-time data, and even in this case, it could be more accurately described as short-term scheduling information regarding the number and type of individuals expected to arrive. [39] does possess the capability to integrate real-time information as it develops short term time series forecasts, but this aspect is not discussed or elaborated in the study. Other time series approaches involve data granularities too large for real-time data to substantially alter their forecasts. The studies generally adopt a static approach, training and validating models on distinct datasets, which may overlook valuable information that could refine short-term forecasts.

Given the above observations, it is apparent that the field of passenger arrival forecasting is not yet well established or "standardised". This has led to a vast array of methods being employed across varying datasets, with no single approach demonstrating clear superiority, as mentioned in the first point. However, there has been a gradual shift towards higher granularity forecasts as noted in the third point, aligning with an increased focus on flight schedules and individual flight arrival patterns (points 4 and 5). The highly stochastic nature of passenger arrivals suggests that the prevalent use of point forecasts may exclude crucial information for decision-making, as discussed in point 6. Furthermore, current literature does not address the integration of real-time information. Even approaches like time series forecasting, which inherently can integrate such data, either deal with too large a time frame or neglect this aspect entirely. In summary, the most significant trend in these approaches has been the shift towards the decomposition of the forecasting problem into smaller components. Through the use of flight schedules individual flights arrival patter and number of passengers can be separated into separate problems.

Considering these trends and limitations, along with GRASP's expressed preference for a forecasting solution conducive to decision-making, a compelling opportunity for research presents itself. The two primary gaps in the current literature are; **a lack of quantification of both model and output uncertainty, and the absence of real-time information integration in current passenger forecasting models**. To exploit this research gap, the following section will delve into the technical details of forecasting approaches and methods that can integrate both interval or density forecasts with real-time forecast updating.

4

Forecasting Methods

Forecasting is the process of making predictions about future events and trends using historical data, typically to improve decision making. In the chapter 3 forecasting methods specifically applied to passenger arrival forecasting were explored, identifying trends and a research gap. The following sections will explore state of the art forecasting methods in detail, taking into account the previously identified trends, continuing to refine the research gap while taking into account requirements set out in chapter 2. First a quick note will be made about this specific forecasting problem in section 4.1, then the selection criteria will be presented in section 4.2. This is then followed by an exploration of the state of the art methods for time series methods in section 4.3, causal models in section 4.4, ML methods in section 4.5, and their extension with the Bayesian framework explored in section 4.6. The chapter is then concluded by providing a trade off table between the methods and selecting a suitable candidate in section 4.7.

4.1. Forecasting vs prediction

One of the most important trends identified in chapter 3 was that the forecasting problem has been getting decomposed into smaller components. Namely using the flight schedule allows approaches to predict the arrival pattern for individual flights, as well as the number of people expected to show up. The schedule provides the temporal relationship between flights, allowing for the more complicated forecasting problem to be broken down into simpler prediction components, differentiating it from more traditional time series forecasting problems. For this section a distinction will be made between prediction and forecasting, prediction concerns creating an estimator $\hat{f}(x)$ that is able to make predictions for new samples x. Forecasting is a sub-discipline of predictions focusing on predictions in the future on the basis of time-series data [14]. For the remainder of this section forecasting will refer to the totality of the passenger arrival problem, whereas prediction will be used to refer to the components of the larger forecasting problem.

4.2. Selection criteria

Before delving into exploring the possible solution approaches, for both forecasting and prediction, selection criteria must be established which will aid with choosing the right approach. Additionally this will also facilitate a focused discussion for each of the proposed algorithms. The primary source for these criteria are the requirements set out by GRASP, and additionally guided by the identified research gap. The combination of which have resulted in two categories of criteria. First, is the performance criteria of the forecasting solution, these are the attributes by which each method will be evaluated throughout this section. Second are the two hard requirements from GRASP that are binary in nature and are either met or not.

Performance criteria

- **Computational efficiency** - From **RQ 3**, near real time updating of the forecast is required, this necessitates quick and therefore computationally efficient algorithms.

- **Temporal resolution** - From **RQ 1**, a minimum granularity of at least 5 minutes is needed, therefore the chosen algorithm should perform well at high temporal resolution.

- **Feature flexibility** - The model needs be able to handle a large number of different features and types, arising from the large number of potentially useful features identified in chapter 3 and available as discussed in section 2.4.
- **Data efficiency** - The number of available data points is relatively limited, with ~12000 available unique flights and therefore the algorithm should have relatively low data requirement.
- **Explainability** - The model will be used for decision support system for non technical stakeholders, making explainability desirable.

Requirements

- **Real-Time Adaptability** - The forecasting model needs to be able to incorporate real-time information about the number of passengers, dynamically updating the forecast. From **RQ 3**
- **Probabilistic Forecasting** - The forecasting model shall be able to quantify the uncertainty of the forecasts. This comes from a **RQ 4**

4.3. Time series models

When dealing with forecasting time series data, classical time series models provide a tried and tested option. These methods typically are very robust, efficient, and flexible while remaining relatively simple and interpretable, this does come at the cost of decreased expressibility. They especially excel at forecasting short-term when there are stable temporal patterns such as trends and seasonality in univariate time series data. While more complex models typically outperform them, they still provide a good base line to measure other models against. For the purpose of this paper special focus is paid to ARIMA, primarily due to it's prevalence in literature and industry as well as it's extensibility.

4.3.1. The ARMA model

Perhaps the most prevalent time series model is the Autoregressive Moving Average Model (ARMA) originally proposed by by George Box and Gwilym Jenkins [10]. The ARMA model, given by Equation 4.1, consists of two parts:

$$ARMA(p,q) = AR(p) + MA(q) \tag{4.1}$$

Where p is the order of the autoregressive term, and q represents the order of the moving average term. The autoregressive (AR) term for predicted value at time t is given by Equation 4.2.

$$y_t = c + \sum_{n=1}^{p} \phi_n y_{t-n} + e_t \tag{4.2}$$

Where y_t is the time series value at time t, c is a constant, p is the number of previous time period's values (lags) used, ϕ is the autoregressive coefficient, and e_t is white noise which follows a normal distribution with a mean of zero. The moving average (MA) term tracks the previous errors of the prediction and is given by Equation 4.3.

$$y_t = c + \sum_{n=1}^{q} \theta_n e_{t-n} + e_t \tag{4.3}$$

Where q is the number of previous error terms used, θ is the moving average coefficient, and e_t is white noise. With the final ARMA equation given by Equation 4.4.

$$y_t = c + \sum_{n=1}^{p} \phi_n y_{t-n} + \sum_{n=1}^{q} \theta_n e_{t-n} + e_t \tag{4.4}$$

A large number of extensions exist for the ARMA class of models which allow it to deal with a wide range of different phenomena. However, unless explicitly dealt with, these classes of models must meet the tree conditions of stationary, which can be checked with Dickey-Fuller Test

- Mean μ is constant
- Standard Deviation σ is constant
- Seasonality doesn't exist

4.3.2. Extending the ARMA model

If the series is not stationary, then differencing can be employed, which results in the Autoregressive Integrated Moving Average (ARIMA) model. This model allows for a non constant mean, especially for non-seasonal trends, by taking the difference between subsequent data points in the series. Additionally this step might help with stabilising the variance. This can be seen as a pre-processing step. If the non stationarity of the series includes seasonality then Seasonal Autoregressive Integrated Moving Average (SARIMA) should be used, which augments ARIMA by introducing a seasonality term [10]. Here the assumption is made that there is a fixed patter that repeats after every m time intervals. The seasonality influences the AR and MA terms by including an additional component that provides offsets to their observations equivalent to the length of the season. The extended model is given as Equation 4.5.

$$y_t = c + \sum_{n=1}^{p} \phi_n y_{t-n} + \sum_{n=1}^{q} \theta_n e_{t-n} + \sum_{n=1}^{P} \eta_n y_{t-nm} + \sum_{n=1}^{Q} \omega_n e_{t-nm} + e_t \tag{4.5}$$

Where the additional variables P and Q are the order of the seasonal autoregressive and moving average components, with η and ω being the vector of coefficients. By default SARIMA models can only incorporate one temporal "level" of seasonality, which means that daily seasonality cannot be combined with weekly seasonality. Extending SARIMA to handle multiple "levels" of seasonality is highly non-trivial, or require specialised models like TBATS [42]. A limitation of the aforementioned methods is their sole reliance on the values of the predicted time series. Since most real world phenomena interact with external variables, these can contain valuable information on the likely value of the predicted series. To address this Autoregressive Integrated Moving Average Model with Exogenous Variable (ARIMAX) expands ARIMA by allowing external variables to influence the prediction. Here multiple time series are used to predict a single value, with the extended equatio given by Equation 4.6

$$y_t = c + \sum_{n=1}^{b} \beta_n X_{n_t} + \sum_{n=1}^{p} \phi_n y_{t-n} + \sum_{n=1}^{q} \theta_n e_{t-n} + e_t \tag{4.6}$$

Where the additional terms β and X represent the exogenous variables coefficient and the exogenous variable respectively. An arbitrary number of these external variables can be included in the model, however they are restricted to inputs that can be transformed to numerical data. Categorical data can also be incorporated by using a one hot encoding strategy, however model complexity and stability are negatively affected if the number of external variables becomes too large [51]. A significant limitation of this model is that it assumes knowledge of the values of the external variables for the same time as the prediction is made for. Furthermore it is also possible to extend the ARIMAX model to include seasonal components, which results in the SARIMAX model. By adding the P and Q summation terms to the ARIMAX equation. Future variants of ARIMA models do exist, but for the scope of this discussion, their capabilities and drawbacks are sufficiently encapsulated by the approaches already discussed.

4.3.3. Modelling uncertainty

While the previously mentioned extensions allow for more powerful forecasting by either taking into account seasonality and/or incorporating external information, they still only return a point forecast. To overcome this issue, additional models can be employed to model the expected volatility, which also relax the constant variance constraint of the stationary requirement. A widely used method is Autoregressive Conditional Heteroskedasticity (ARCH) which is used to model conditional volatility of time series data, and falls under the broader category of stochastic volatility [16]. This model requires the mean to be forecasted, by approaches such as ARIMA, and then using the residuals, the volatility of the errors are predicted. The primary reason for the popularity of this approach is their ability to handle non constant variance in the data (heteroskedasticity) and their parameter efficiency. The error term ϵ_t (residuals) are split into two parts given by Equation 4.7

$$\epsilon_t = \sigma_t Z_t \tag{4.7}$$

Where the random variable Z_t is a strong white noise process, and σ_t^2 is a series of standard deviations modelled by Equation 4.8.

$$\sigma_t^2 = \alpha_0 + \sum_{n=1}^{q} \alpha_t \epsilon_{t-n}^2 \tag{4.8}$$

Where q is the order of the ARCH process, and α is a vector of parameters, with the following conditions of $\alpha_0 > 0$ and $\alpha_i \geq 0$, $i > 0$. Effectively this means that the standard deviation of the next time step is a weighted sum of the previous time steps residual errors. The ARCH model is only suitable if the error variance can be represented using a time series that follows an autoregressive model. If the error variance follows a ARMA model then the Generalised Autoregressive Conditional Heteroskedasticity (GARCH) model must be employed [9]. With the series of standard deviations σ_t^2 being modelled by Equation 4.9.

$$\sigma_t^2 = \omega + \sum_{n=1}^{q} \alpha_n \epsilon_{t-n}^2 + \sum_{n=1}^{p} \beta_n \sigma_{t-n}^2 \tag{4.9}$$

Where ω is a constant, q is the order of the ARCH terms ϵ^2, p is the order of the GARCH terms σ^2, and α and β are vectors of parameters. Effectively this extends ARCH by also taking into account a running weighted sum of the previous time steps estimated conditional variance. (G)ARCH models can be hard to fit due to constraints on the parameter values, and have low prediction power over longer time horizons. Similarly to ARIMA models, there exists a large number of variants of (G)ARCH models, which for the purpose of this discussion have the same high level benefits and drawbacks. A number of other alternative stochastic variance models exist, however these approaches are primarily focused at forecasting volatility in the specific context of financial markets.

4.3.4. Conclusion

For the purpose of this section only ARMA based approaches have been considered, and while there exists other time series approaches such as exponential smoothing models, or error trend seasonality (ETS) models. These are either too simplistic, or are very infrequently used in literature. On the surface SARIMAX with GARCH errors seems to be an excellent fit to the forecasting problem. It inherently can use real time data to make forecasts, and with the GARCH errors it also has the capability to quantify uncertainty in the model. Additionally it has reasonable scores for the performance criteria.

- **Very computationally efficient** - Both training and inference is very fast for these simple models
- **Low feature flexibility** - While external regressors can be used the type and amount of features that it can incorporate is limited, and quickly increases model complexity and decreases explainability.
- **High data efficiency** - A reasonable model fit can be achieved with relatively little data.
- **Reasonable temporal resolution** - The output resolution purely depends on the resolution of the available data, however there are practical limits to the temporal resolution that are determined by how much noise is in the data, which could break underlying assumptions.
- **Low to moderate explainability** - While the fit parameters are interpretable in a statistical sense, they usually do not translate to real-world insights, especially for non-technical stakeholders.

4.4. Causal Models

Causal models are statistical or computational models that aim to capture the cause and effect relationship between variables. This is achieved by representing the causal relationships of the variables either through directed graphs, mathematical relationships, or rules. The main benefit of these approaches are their inherent ability to incorporate varied types of external sources of information and represent the structure of the problem in the models. These models have a wide range of applicability including predicting outcomes based on interventions, identifying confounding variables, analysing feedback loops, understanding temporal relationships, and simulating complex dynamic systems. For the purpose of this paper the two main categories of statistical causal inference, and simulation modelling are of interest due to their applicability to forecasting and prediction.

4.4.1. Statistical Causal Inference

This category includes methods that primarily use statistical techniques to infer causal relationships from data. These methods often deal with observational data, and aim to discern and quantify cause-and-effect relationships among variables. The most common and perhaps most widely know approach in this category is regression, where the relationship between the dependent (outcome) variable and one or more independent variables is established. The simplest form of which is multiple linear regression, which has the form given by Equation 4.10.

$$y = a_0 + \sum_{i=1}^{n} a_i x_i + \epsilon \tag{4.10}$$

Where y is the dependent variable, x_i are the independent variables, a_i are the coefficient to be estimated, and ϵ is the error term. Using a similar approach, other regression models such as polynomial regression (that allow for nonlinear relationships), or other more sophisticated models such as ridge and lasso regression are available. The previous two extend any regression model by providing a penalty term that encourages parameter values to remain as small as possible, which reduces overfitting of the model [44]. The primary benefit of these approaches is their ease of implementation, explainability, and data efficiency when dealing with smaller models. However these approaches usually perform poorly in high dimensional non-linear systems, furthermore regression is primarily used for prediction rather than forecasting.

However direct forecasting might not actually be desirable, a significant observation from chapter 3 was that there is a shift to represent the arrival rate of individual flights. This allows the transformation of the forecasting problem, which involves estimating the number of arriving passengers, into a prediction problem. Where the arrival patter of each individual flight needs to be predicted, which then can be summed up using the flight schedule. Here a number of features can be used to predict both the arrival pattern and the number of people that will show up to a flight. This approach allows to indirectly incorporate the incredibly rich information of the flight schedules, which is a non trivial task for other forecasting approaches.

A possible downside of "plain" regressions models is that they assume direct causal relationship between the independent and dependent variables. Methods like Structural Equation Modelling (SEM) combines factor analysis with simultaneous equation modelling, and allows for representation of latent variables. In these models the causal relationship between variables is modelled through equations, which then allows for estimating the strength of the relationship within both observed and latent variables. This is desirable in a number of application fields where there are latent variables that are difficult or impossible to observe. Generally speaking techniques in the SEM family can be summarised by the following three steps [48]:

- Definition of Equations or model specification
- Estimation of free parameters
- Evaluation of the model and model fit

Other methods such as Granger Causality, Instrumental variable estimation, and Propensity Score Matching also fall under statistical causal inference, however these approaches typically follow the structure outlined for SEM [53]. Furthermore these approaches usually are not employed for forecasting and predictions, but for estimating effects of a treatment, policy or other intervention. While most of the methods that fall under statistical causal inference are not well suited for prediction tasks, the general framework used by them provides a robust platform into which other approaches can be integrated into, thereby increasing their versatility. Finally these causal models allow for the possibility to transform the forecasting problem into components which can simplify the overall model, requiring prediction models with less expressibility. That can allow for smaller, faster, more explainable, and more data efficient algorithms.

4.4.2. Simulation based modelling

For complex systems with stochasticity or uncertainty, statistical causal modelling quickly becomes insufficient, here simulation based modelling can be used, as it inherently captures cause and effect relationships between components in a system. By creating representations of the underlying mechanisms and dynamic interactions. One of the first applications of this approach is system dynamics (SD) which was first introduced in 1961, focusing on systems thinking, with combination of constructing and testing a computer simulation model [18]. These models are typically modelled when a complex system needs to be represented that has

time varying behaviour change, as well as the existence of closed loop feedback. Here a causal loop diagram is used, which is a directed graph, to indicate which variables are connected, and the direction of causality. When constructing and refining a system dynamics model the following 5 steps must be iterated on [57]:

- **Problem articulation** - Identify the problem and the key variables and concepts, determine the time horizon and characterise the problem dynamically for understanding.
- **Dynamic hypothesis** - Develop a theory of how the identified problem, and develop causal links between the variables, which allows the construction of the causal loop model.
- **Formulation** - Translate the system description into rate, level, and auxiliary equations. This step will help identify parameters to estimate, and identify inconsistencies in the model.
- **Testing** - Compare the simulated behaviour of the model to the actual behaviour of the system.
- **Policy formulation and evaluation** - If there is sufficient agreement between the model and the system, then the model can be utilised to design and evaluate policies.

SD models allow for modelling a wide range of problems, and are particularly well suited for modelling "transparent" systems where identifying relationships between different components are conceptually or intuitively clear, but the strength of these relationships are unknown. However these approaches are not well suited for fine grained and short term analysis, as they are more oriented towards understanding long-term trends and behaviours. If micro level behaviour of the system needs to be represented, then agent based modelling (ABM) is better suited for the problem [43].

ABM's are computational models that simulate interactions of individual agents within an environment, and is particularly well suited for analysis of emergent behaviours and phenomena in complex systems. This is achieved by defining agent behaviour and interactions, allowing for heterogeneity at the micro-level among agents, thereby replicating macro level patterns and dynamics. In these simulations agents are autonomous entities that observe their environment, update their belief about the world, then use their goals or basic behaviour to determine the next action that they will take. In Figure 4.1 an example architecture for a passenger agent in an airport simulation is presented.
ABM simulation models have been applied to airport operations with great success, providing a solid platform that allows for a wide range of research to be conducted [28], [60], [36]. In these studies ABM allows for evaluation of micro to macro level policy changes, the effects of which would be near impossible to replicate in a top down simulation approach. However there are a number of drawbacks to this modelling paradigm. Firstly ABM typically have substantial computational costs, especially when compared to non-simulation based approaches, this is compounded if stochasticity is introduced into the system where a large number of runs might be required for stable and statistically significant results. Second, calibrating ABMs can be exceptionally challenging, and impractical when lacking granular data on agent interactions and behaviours [24].

In conclusion for problems where micro level interactions are of importance, ABM provides the most comprehensive representation possible, if interactions between individuals can be modelled. Whereas for more macroscopic interactions approaches such as SD are better suited. And while there are other simulation based approaches, ABM and SD provide a good overview along the primary problem dimension ranging from individual-level complexities to system-level dynamics. However these approaches both are frameworks with which system behaviour can be explored. Since they require estimation (forecasting) of all their independent parameters and their interactions. Which can be beneficial when individual parameters are easily estimated, and interactions clearly defined. This means that they are less well suited to systems where the process to be forecasted is more of a black box. Finally even if the system can be well defined, all inputs to the model either need to be forecasted or estimated themselves, meaning that other methods must be used for this task, increasing model complexity and decreasing traceability.

4.4.3. Conclusion

Both categories of modelling approaches have significant downsides when trying to apply them to the forecasting problem at hand. Firstly statistical causal approaches are suited best for prediction problems, however as discussed can be transformed to do forecasting with the use of flight schedules. Nevertheless these approaches can neither update their predictions based on new data available, or incorporate uncertainty without significant adjustments to the models. Which makes them unsuitable in their current state for application to the proposed problem. Yet they do have desirable performance characteristics:

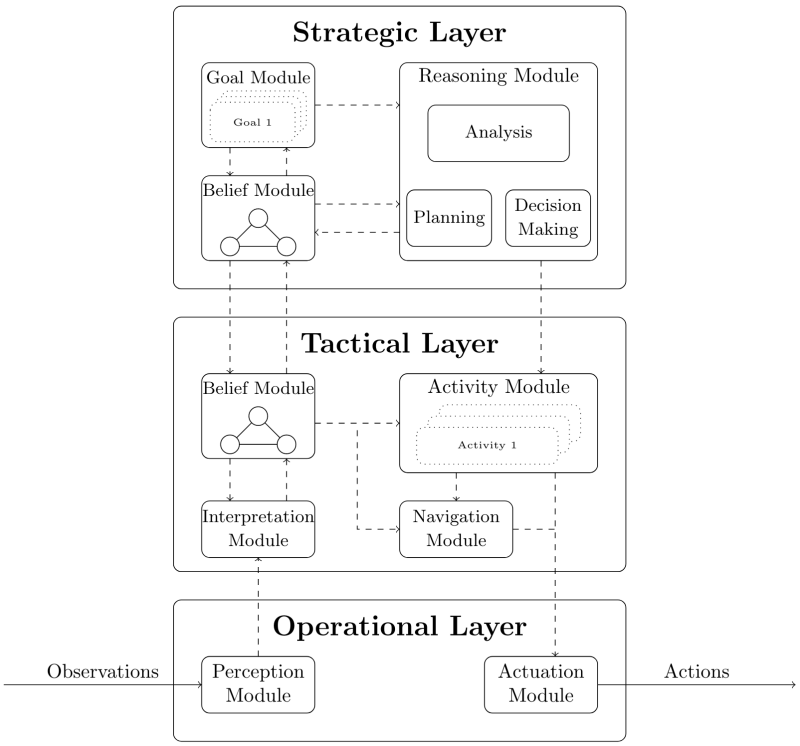


Figure 4.1: AATOM ABM architecture of an airport passenger agent, the operational layer is responsible for low level interactions with the environment, the tactical layer is responsible for interpreting the observations and making short term decision such as pathing, and the strategic layer determines goals, reasons and updates the agent's belief [28].

- **High computational efficiency** - a wide number of very efficient optimisation techniques such as least squared, and maximum likelihood estimation exist to fit parameter values. And inference is incredibly quick even with reasonable large regression models.
- **Moderate/poor feature flexibility** - Addition or removal of features is quite easy, however model performance decrease with high number of features.
- **Good data efficiency** - With the use of domain knowledge, regression can extract meaningful insights from limited datasets.
- **High temporal resolution** - If the arrival patter distribution of each individual flight is modelled then very high frequency forecasts can be produced
- **Highly explainable** - Influence of each variable is easy to interpret.

On the other hand simulation based approaches are able to produce probabilistic forecasts, by introducing stochasticity into the models and sampling through a large number of runs. Additionally real time data can be used to update model parameters and inputs that enable new runs to incorporate the information. However while they meet the required capabilities of the desired forecasting tool, their performance characteristics are quite poor.

- **Incredibly computationally heavy** - This is primarily an issue for ABM, however once uncertainty must be represented both models become prohibitively expensive to run.
- **Moderate feature flexibility** - Depending on the type of feature integration could be quite straightforward or difficult.
- **Low data efficiency** - Depending on the number of observable features, very large amounts of data might be required to fit hidden features to the simulation based models.

- **High temporal resolution** - ABM allows for extremely fine temporal resolution, however SD is less well able to model and represent the short term micro interactions.

- **Good explainability** - Both ABM and SD variables try and represent their real world counterparts, or some useful abstraction of them.

4.5. Machine Learning (ML) Models

As mentioned in the previous section the forecasting problem can be broken down into a prediction problem in combination with the flight schedules. Machine learning (ML) based approaches excel at capturing complex nonlinear relationship in data and therefore are very well suited for the task of prediction. Furthermore specialised ML methods such as LSTM's have the capability to directly solve the forecasting problem with their recurrent architecture that is designed to process time series data. The main advantage ML methods is their ability to handle nearly arbitrary data with very little assumptions.

4.5.1. Traditional ML Models

More "traditional" ML models nearly exclusively deal with prediction, where there is an input output relationship. And there are a large number of models in this context, however for this paper only decision and regression trees will be explored. This is primarily done since these were the only two traditional ML models employed in the reviewed literature, as well as their benefits and limitations being representative enough of models in this class. Both decision and regression trees create tree like structures which partition the data by selecting features by which each "branch" discriminates which sub branch an input belongs to. Decision trees were first proposed in 1959 and still enjoy wide spread use due to their ease of implementation, ability to handle mixed and non pre-processed data, and robust performance [4]. Decision trees work on categorical target values, this was later extended by Morgan in 1963 to work on continuous target values, resulting in regression trees [49].

Both decision and regression trees follow a 3 step process to produce predictions [34]:

- **Tree growing** - A tree like data structure is recursively built up where the training data is split according to some feature or criteria. For nodes where no further splits are required a leaf is generated, and a target value is assigned to it. The pseudo code for which can be seen in **cite code block**, which shows how to build decision trees, however the algorithm is virtually the same for regression trees. Where instead of categories target values are taken for leafs, typically by averaging the samples that fall in them.

- **Tree pruning** - Once the tree has been constructed, the predictive utility of all the nodes are evaluated, and nodes that don't improve the predictive ability get converted to a leaf. This step aims to reduce overfitting of the model.

- **Predictions** - An input with some features is passed to the first node, and depending on the split criteria of the node, an appropriate child node is selected. This continues until a leaf is reached, which then is the output of the prediction.

Algorithm 1: Decision tree generation pseudo code [34]

```
S ← Samples
F ← Features
Function Gen_Decision_Tree(S, F):
    if S meets the stopping criterion then
        return leaf with average category
    end
    A ← Select best feature to split on from F
    V ← Split feature A into distinct values
    for v in V do
        create new branch with A = v
        S_v ← subset of S where A = v
        if S_v is empty then
            add to branch leaf with most common category in S
        end
        else
            add to branch Gen_Decision_Tree(S_v, F)
        end
    end
```

The most important central choice to make in both of these algorithms is finding the best split at each node. With the goal of reducing the variance of the target value for samples in a node. There exists a number of different splitting algorithms to achieve this task. The CART algorithm follows a more greedy splitting criteria search approach, that constructs an overfitted tree, which then is pruned using cross-validation estimate to identify which "branches" to prune. However this algorithm only works on ordinal data, this is solved by C4.5 which is able to handle both ordinal and categorical data for the tree creation. Furthermore there exists a number of other methods for determining the best splitting criteria, however these methods are quite situational, and don't contribute much to the overview of the models. Overall traditional ML methods like the above mentioned ones are quite flexible with the type of features they can interpret, however increasing the number of features either requires more training data, or reduces model performance.

Uncertainty using random Forest

An extension to both of the previously discussed approaches is random forest, this can be seen as an ensemble model. Here instead of fitting one tree to the data, multiple trees are fit with bootstrap aggregated data, where a random sample of the original dataset is taken with replacement. The output of the model is a number of predictions made by each of the decision/regression trees fitted. This provides 3 significant benefits, firstly they are much less susceptible to overfitting compared to their single tree versions due to the randomness of sampling, secondly they tend to have better prediction performance, and lastly they are able to capture uncertainty of the model. This last point is achieved by either calculating the class probabilities, or taking the standard deviation of the predicted value.

4.5.2. RNN's

Traditional ML models don't have a clear and direct application for forecasting time series data, this is due to the fact that these models have a fixed number of inputs that is not well suited to time series data. To overcome this recurrent neural networks (RNN) were first hypothesised in 1986 and then established as we know them today in 1990 [15]. These networks attempt to capture temporal relationships between datapoints by continually managing an internal state which gets updated with each new data point. Hence the "recurrent" in the name, since this internal state is directly fed from the output of the network to the input, along with a new data point, producing cyclic connections.

An issue with early RNN architectures was the "vanishing gradient" problem, where the gradient of the loss function would tend to 0 or infinity, this made learning long term dependencies nearly impossible. To overcome this the Long Short-Term Memory (LSTM) RNN architecture was proposed in 1997, which has two internal states that can be seen as describing short and long term information, that it updates relatively independently [27]. This in combination with the introduction of gating units, that are responsible for selectively remembering and forgetting information, allows the LSTM to maintain a constant error flow, and therefore traceable gradients. The architecture of the LSTM can be seen in the figure below

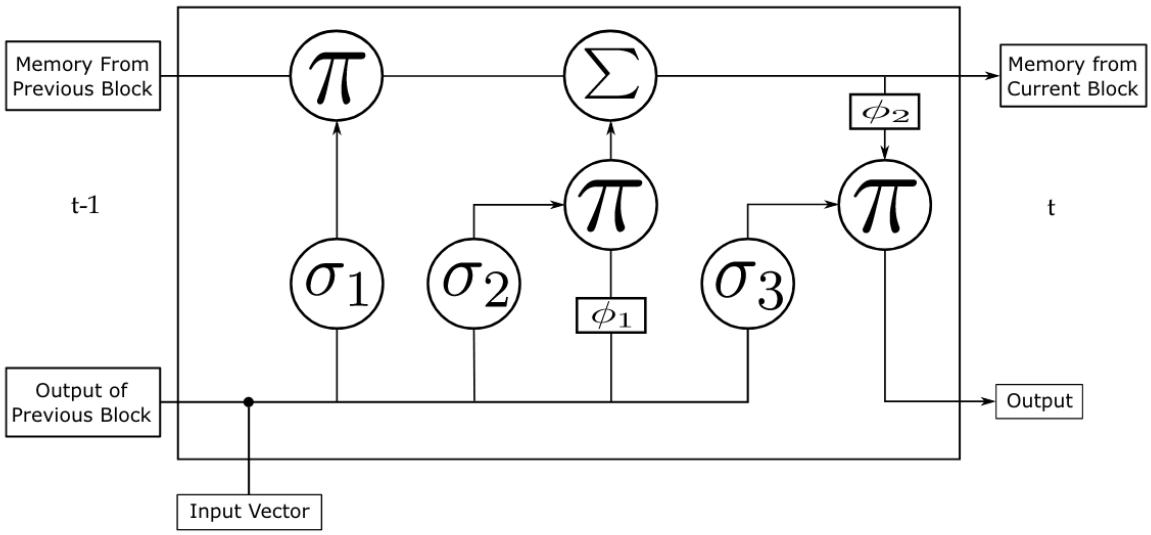


Figure 4.2: LSTM architecture, σ represents the gate functions responsible for retaining and discarding information, ϕ are the output activation functions [35]

LSTM's have enjoyed wide spread use for the purpose of time series forecasting, and while alternatives exist, LSTM's are still the default choice for these types of problems. For certain situations Gated Recurrent Units (GRU) are sometime preferred, this model is computationally more efficient version of the LSTM, since it contains less games and parameters. While this model is less powerful it is especially well suited to problems where there is limited data availability. On the whole, RNN based architectures have been explicitly made to handle time series data, and they have very good performance. They are able to handle arbitrary inputs that can be encoded numerically or categorically (1/k hot encoding). However they require a very large amount of data to train, with additional features exponentially increasing both data requirements and computational resource utilisation.

Uncertainty in RNN

Uncertainty in RNN can be achieved through the same mechanism as done with random forest for decision and regression trees. Through bootstrapping, multiple models are trained on different subsets of the data, then evaluation of the output of all the models can be aggregated and the variance in the outputs treated as the uncertainty. However this can very quickly become prohibitively expensive as single LSTM models are already considered to be computationally expensive to train on moderate data sets. An alternative approach is to use mixture density networks (MDN) which can extend the LSTM architecture, by adding a number of layers on the output of the LSTM [6]. These additional layers try and predict parameters of a pre specified probability distribution, instead of the single output value, thereby quantifying uncertainty. And while this is less computationally expensive than bootstrapping multiple models the addition of MDN adds increases complexity to the model, making the already data hungry LSTM require even more training data.

4.5.3. Transformers

One of the largest and most recent shifts in ML has been the introduction of transformers, which use self-attention to simultaneously weigh and process all elements in an input sequence [61]. This new architecture has taken the ML community by storm, and has proven to be widely applicable and very powerful. This has also been the case for time series forecasting where the attention mechanism allows for similar behaviour as the LSTM's remembering/forgetting gates. However as with all new novel technologies, their true limitations have not been fully explored, and this has lead to over utilisation of this technique. A recent study brought this issue to light where it found that an extremely simple single linear layer model was able to outperform transformers on a number of established data sets [65]. This was primarily attributed to the difficulty of encoding the temporal aspect of the data, as transformers were originally developed for natural language processing, and therefore require work arounds to facilitate proper time series data representation.

4.5.4. Conclusion

One of the main benefits of ML techniques is their flexibility both with the type of inputs they are able to process, and their ability to represent highly non linear complex phenomena. However, especially the last point, comes at a cost. This problem is know as the bias-variance trade-off, where high expressibility models have low bias but high variance, meaning that they tend to overfit or require large amounts of data. Conversely lower expressibility models usually have high bias but low variance, which can lead to under fitting. From the above methods the "traditional" ML tree based methods have lower expressibility, and need to solve the simpler prediction problem, however they are not able to incorporate real time data to update their forecasts. The extension of random forest does allow for return probabilistic predictions, and they have relatively desirable performance characteristics:

- **Moderate computational efficiency** - Single trees are very cheap to evaluate, and this does not become a significant challenge even when ensemble methods are used.
- **Moderate feature flexibility** - Numerical features are easy to add, however categorical features can cause issues if the number of features becomes high.
- **Moderate data efficiency** - Due to its lower expressibility decision/regression trees are able to construct models from moderate amount of data.
- **High temporal resolution** - Since the prediction problem could be used for arrival patter distribution parameters, the resulting forecast can have very high resolution.
- **High/Moderate explainability** - One of the primary reasons to use decision/regression trees is for their inherent explainability within the ML context, however practically speaking fully understanding the reason for a result might be difficult to understand if the tree is deep.

LSTM's on the other hand are a lot more expressive and are much more capable at representing complex phenomena. However they have to solve the more difficult problem of forecasting, additionally encoding flight schedules and information about each flight would result in a very large input vector, further complicating training and increasing data requirements. However if enough data is available LSTM's are able to both use real time data to update their forecasts, and incorporate uncertainty using MDN layers.

- **Medium computational cost** - Training LSTM's is incredibly expensive, however at inference it is relatively cheap to run.
- **Moderate feature flexibility** - Similarly to decision/regression trees features are easy to add, but significantly increase the required amount of training data.
- **Very low data efficiency** - Especially when having to incorporate probabilistic outputs the amount of training data required is incredibly high.
- **High temporal resolution** - The output temporal resolution only depends on the input resolution, however noise in the data limits reasonable minimum resolution.
- **No explainability** - Practically speaking model parameters cannot be interpreted, this model is fully a black box.

4.6. Bayesian Framework

As discussed in the chapter 3 there is a lack of representation of uncertainty, with only one paper using a Bayesian framework, and even in this situation it was only applied to long term passenger forecasts. However there are numerous benefits to the use of this framework. Bayes theorem is the most fundamental equation in probability theory, and is the foundation for the Bayesian framework, it describes the probability of an event based on prior beliefs and the likelihood of observing the new evidence given that the event occurs. Which mathematically is given by Equation 4.11

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)} \tag{4.11}$$

Where $P(A|B)$ is the posterior probability of event A given evidence B, $P(B|A)$ is the likelihood of A given B, and $P(A)$ and $P(B)$ are the probabilities of observing A and B respectively known as the prior probability and marginal probability.

In the context of forecasting, this means that prior knowledge or beliefs can be incorporated about the values of model parameters and update these beliefs as new data is observed. This not only allows for dynamically updating, but by maintaining distributions over possible parameter values, Bayesian approaches also represent uncertainty in their predictions. However outside of textbook exercises and toy examples, "pure" application of Bayes' theorem is seldom used. Instead, Bayes' theorem often forms the foundation of a Bayesian framework, which enables incorporation of prior beliefs and uncertainties in a systematic way. This framework is then employed to augment other statistical models, enabling them to account for uncertainty, and therefore make probabilistic predictions.

When extending an existing approach with the Bayesian framework, there are a number of consequences. Firstly all probabilistic parameters in the models require estimates of the priors, for explainable models this allows domain and expert knowledge to "inject" information into the model. Which then reduces the required number of datapoints to construct a sufficient model. However models where priors are hard to estimate, due to lack of information or domain knowledge, or because parameters do not directly represent values from the system it is trying to represent, e.g. black box models. Here the amount of training data required to construct a model could increases, as very wide initial priors have to be set. Second, as previously mentioned, the addition of the bayesian framework allows to quantify model and data uncertainty. And lastly the ability of bayes theorem to incorporate new new observations, allows for online updating of the model. These last two points make the bayesian framework especially attractive for the purpose of this research. In the following sections each of the aforementioned main categories of forecasting methods will be explored using a bayesian framework.

4.6.1. Bayesian Time series models

As discussed in the time series section ARIMA models and their extensions are widely used, and can even be extended to take into account uncertainty through the use of (G)ARCH model. An alternative approach would be to treat all variables of the ARIMA model as distributions, which when sampled would produce a density forecast. However a more bayesian approach was developed and refined in 2014 called Bayesian Structural Time Series (BSTS) which can be seen as the bayesian extension of ARIMA [54]. The focus of which is to allow an arbitrary number of structural bayesian components such as trend, seasonality, and regression elements to be integrated, allowing greater flexibility compared to ARIMA. There are three key features that represent BSTS:

- **Uncertainty quantification** - the Bayesian approach inherently is able to capture uncertainty.
- **Explainability** - each structural component can easily be retrieved and interpreted.
- **Extensibility** - the model can easily be extended with external regressors.

There are two equations that govern BSTS. Where Equation 4.12 is the observation equation that links the observed data y_t to a vector of latent variables α_t that is called the "state". The Equation 4.13 is the state transition equation, which describes how the latent state evolves over time [29]:

$$y_t = Z_t^T \alpha_t + \varepsilon_t \quad \varepsilon_t \sim N(0, H_t) \tag{4.12}$$

$$\alpha_{t+1} = T_t \alpha_t + R_t \eta_t \quad \eta_t \sim N(0, Q_t) \tag{4.13}$$

With Z_t, T_t, and R_t being structural parameters, which contain parameters determining the relationships between each component. And the residuals ε_t and η_t are independent of each other represented by normal distributions with 0 mean. The main benefit of BSTS models is their modularity and flexibility, however these still only include the trends in the time series data, with the possibility of using external time series regressors. And while this approach typically performs better than ARIMA, it is also requires significantly more computational resources for both training and at inference. Finally BSTS has similar downsides to the more traditional time series methods like ARIMA, which is that it is only able to integrate time series regressors. That is more complex data, like flight schedules, cannot easily be integrated. To summaries compared to traditional time series models BTST: decreases computational efficiency, has a greater capability for additional features, decreases data efficiency, and slightly increases explainability.

4.6.2. Bayesian Causal models

Bayesian frameworks offer a very natural and powerful extension to causal models. This can be attributed to several properties of both approaches that complement each other especially well. Firstly, one of the main benefits of causal models is their interpretability, which typically means that the parameters of the models represent certain aspects of the real world system or phenomena. This allows for an intuitive and simple way to incorporate prior knowledge of the underlying system. Additionally not only is it possible to pass prior knowledge through parameter priors, but the architecture of the causal model also allows to provide priors in the form of model structure. Secondly the mathematical structure of causal models, and especially regression models, allows for easy integration with bayesian methods. This is due to most causal modes using a likelihood function to drive parameter estimation, which is a similar approach to how bayesian approaches update their posterior distributions. Finally Bayesian approaches aid with reducing model overfitting, as they tend to capture noise as uncertainty. These previous benefits are especially apparent for simpler regression models such as SEM and related statistical causal models [38].

A natural extension to Bayesian regression is hierarchical bayesian models, which are especially well suited for grouped/clustered data. This approach allows for estimating parameters at different aggregation levels in the data, with parameters "deeper" in the model able to use information about the broader group higher in the hierarchy. This allows for groups with little available data to incorporate additional information, and thereby improve their predictive performance [2]. Effectively these hierarchical models can be seen as nested regression models, where each group-level distribution provides priors to the individual-level parameters. Mathematically this can be represented as follows, assuming there are 2 levels (for example, individuals within groups).

$$Y_{ij} | \beta_i, \sigma^2 \sim N(x_{ij}\beta_i, \sigma^2) \tag{4.14}$$

$$\beta_i | \mu, \tau^2 \sim N(\mu, \tau^2) \tag{4.15}$$

Equation 4.14 represents the likelihood, Y_{ij} which represents the j-th observation in the i-th group, and describes how the observations are conditionally distributed based on the group level parameters β_i and model parameter σ^2. Equation 4.15 provides the group specific parameters conditioned on hyper-priors μ, and τ^2. Here all hyper-priors must be represented by some distribution, and represents the prior information provided to the model, with normal distributions commonly used for their favourable mathematical properties. This extension to regression models is especially well suited to the prediction problem at hand, as it resolves the two main issues with the frequentist regression approach, namely the lack of uncertainty quantification and inability to incorporate real time data. However the additional predictive power and versatility does come at the cost of increased computational complexity, for implementation, training, and inference. To summarise bayesian regression compared to plain regression, decreases computational efficiency, and retains the other characteristics, while providing uncertainty quantifications.

Finally simulation based modelling can also be extended to utilise bayesian frameworks. This is more interesting for SD approaches, as ABM typically already have the ability represents uncertainty by the introduction of stochasticity between runs. An interesting extension to SD is in the form of Bayesian belief nets (BBN). BBN's are probabilistic directed acyclic graphs (DAG) where edges represent the conditional dependencies of each node (variable) on each other. With a significant benefit of this approach that not only does it learn the relationship between variables, but it is also able to learn the structure of the DAG. However this flexibility and adaptability requires prohibitive amounts of data, with learning of the structural representation being the more difficult aspect [32]. This is where the simpler SD approach is able to compliment BNN's by providing the initial deterministic structure. Which then allows for an iterative process where the BNN is able to provide probabilistic insights into the parameters and structure, while SD provides a more efficient way to evaluate the network [46]. Effectively allowing to incorporate both direct priors for the variables in the BNN as well as priors on the structure of the system in the form of the SD model. While the extension of SD with BNN provides a probabilistic extension, for the purpose of the forecasting problem, it increases the computational requirements, the amount of data necessary, and make it more difficult to adapt different features. For these reasons simulation based models with bayesian extensions will not be considered as the performance degradation is too significant.

4.6.3. Bayesian ML models

One of the consequences of applying a bayesian framework to black box models where domain knowledge cannot informs priors, is that it usually increases the data required, model complexity, and computational cost. All of which are already downsides of ML approaches for the forecasting problem outlined. Furthermore all discussed ML models already have a way to incorporate uncertainty into their models, and the real time updating capability of bayesian approaches has little use with methods like LSTM's. However since traditional ML methods would be used to produce predictions on the arrival pattern, the probabilistic output of them could be updated with new data. This extension allows them to meet the real time data integration criteria, making them an eligible option. However the bayesian framework: further decreases computational efficiency, and also decreases data efficiency.

As opposed to all previously mentioned methods Gaussian Process (GP) is a fundamentally bayesian approach, and more specifically a ML model that is well suited to represent time series data. GP is a nonparametric, stochastic process which defines joint Gaussian distribution over random variables. Function $f(x)$ that follows GP is defined by the following Equation 4.16.

$$f(x) \sim GP(m(x),\, k(x, x')) \tag{4.16}$$

Where $m(x)$ is the mean function and $k(x, x'))$ is the covariance function, here the mean function gives the expected value at each point x and the covariance function describes the relationship between function values at different points (between x and x') [48]. The most important choice to be made with GP is the choice of the covariance, also known as the kernel, function. There are two broad options to choose from, more standard and established kernels are typically easier to tune, more efficient, and can be interpretable. Alternatively a custom kernel function can be designed that take into account characteristics of the problem, which can result in better performance given sufficient domain understanding and if validation is possible [55]. By themselves GP is just a distribution over functions, to apply it in a forecasting setting GP must be extended to Gaussian process regression (GPR). GPR conditions the underlying GP on observed data, thereby getting a posterior distribution over functions that aligns with this data. GP(R) are able to capture both the noise inherent in the data as well as the errors in the parameter estimation precess, and are relatively computationally efficient given the use of sparse approximation in lower dimensions [55] [13]. However these methods typically work with continuous inputs, meaning that integration of more complex features can prove difficult, furthermore the interpretability of the model is quite poor. Finally they can have poor computational scaling with data and features.

4.6.4. Conclusion

Bayesian frameworks are incredibly flexible and are able to extend most models, providing uncertainty quantification, as well as the ability to update model predictions using new data. However the costs and benefits of applying them to different classes of models is not equally shared. On one had statistical causal models like regression is able to synergies especially well with bayesian methods, where domain knowledge can provide a lot of information for both parameter and structure priors. While approaches such as BSTS provide much more modest improvements, that mostly stem from bayesian approaches modularity, without improving the inherent shortcomings of traditional time series approaches. Finally in the context of the forecasting problem bayesian extension of ML models amplify the downsides without providing appreciable benefits compared to established methods to quantify uncertainty in those models. While GPR is a native bayesian ML approach and potentially reasonable performance, difficulty related to encoding more complex features decreases its viability.

4.7. Model selection and conclusion

In the current section a number of models were explored that can either directly solve the forecasting problem or its prediction components. Using the criteria outlined, methods that meet the requirements of **real-time adaptability**, and **probabilistic output** will be scored and compared. For each criteria a score from 1-5 will be assigned to each of the models, with 1 corresponding to poor performance and 5 to excellent performance. The justification for scores are presented in their respective sections.

Following are justification for the inclusion or exclusion of certain models. SARIMAX-GARCH is the most powerful time series method available, and simpler time series models have lower expressibility and no appreciable benefit. Pure causal models were excluded because statistical methods did not meet the probabilistic and real time requirement, and simulation based methods are more of frameworks as opposed to

forecasting tools themselves. Traditional ML methods were excluded as they are not able to use real time data to update their predictions. Bayesian extensions to simulation based models were not included because they did not solve the underlying issue with them. While bayesian extension to LSTM architecture would increase computational complexity, and further decrease data efficiency.

	Comp	Temp	Feature	Data	Explain	**Score**
SARIMAX-GARCH	5	3	2	4	3	**17**
LSTM-MDN	3	4	3	1	1	**12**
BTST	4	3	3	3	4	**17**
Bayesian regression	4	5	3	4	4	**20**
Bayesian ML methods	3	5	3	1	3	**15**
GPR	4	5	2	3	1	**15**

Table 4.1: Trade of table for suitable forecasting methods

Table 4.1 gives the full trade-off table, with the best model being Bayesian regression. This is primarily due to a good fit with the prediction problem, and it's computational and data efficiency are especially good. Originating from how well prior information, both structural and parametric, can be integrated into the models. The following section will discuss how to evaluate the performance of a probabilistic forecasting model.

5

Performance evaluation

Once a forecasting model has been calibrated it's performance must be evaluated. Since in the previous chapter a probabilistic forecasting model was chosen, the quantification of error becomes non-trivial. In the following chapter a general overview of how evaluation is performed on forecasts is presented in section 5.1, followed by an in depth discussion on how to evaluate probabilistic forecasts in section 5.2. And the section is concluded by a brief discussion on which metric is most suitable in section 5.3

5.1. Forecasting evaluation overview

In order for a forecast to be reliable and usable, it must be evaluated, which entails components of both verification and validation of the model. Traditionally forecasting methods produce point forecasts, like the simple time series approaches. Here the predicted value can be directly compared with observations, allowing for errors to be distance measurements between the two values. There exists a wide number of scoring algorithms, each with their own inherent benefits and downsides. With three widely used measures being; the Mean Absolute Error (MAE) which measures the average error magnitude Equation 5.1. The Root Mean Squared Error (RMSE) which penalises outliers more severely Equation 5.2. And the Mean Absolute Percentage Error (MAPE) which provides a relative measure of prediction error Equation 5.3 [21].

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|\hat{y}_i - y_i| \tag{5.1}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2} \tag{5.2}$$

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}|\frac{y_i - \hat{y}_i}{y_i}| * 100 \tag{5.3}$$

However these metrics require a single forecasted value, and would not be able to adequately asses the quality of a density forecast. Once the forecasting problem is extended out to be probabilistic there are a number of additional considerations that have to be made. First and foremost, when evaluating a density forecast there are two distinct measurements that can be optimised for, the calibration and sharpness of the models. The calibration is defined as how valid or reliable the forecast is, a model is well-calibrated if the observed frequency of events matches the predicted probability over a large number of forecasts [45]. An event predicted to have a 50% chance of occurring should occur 50% of the time. This however is not a complete evaluation of the model, a weather model the predicts 30% rain every day in an area where on average it rains 30% might be calibrated but it's not informative. The sharpness of output is related to the refinement or resolution of the model, and describes how "narrow" the forecasted densities are [45]. Therefore a good measure should be able to represent both the calibration and sharpness of the output. And while it is possible to use multiple performance metrics, it is preferable to have a single value. These two evaluation dimensions also line up well with the two indicators that GRASP wants to evaluate the forecast on, first is a long term accuracy of the number of passengers. This can be seen as the calibration of the model. Secondly there is a desire for capturing the short term fluctuations, which is analogous to the sharpness of the model.

5.2. Probabilistic evaluation metrics

When dealing with probabilistic output there are two broad categories of methods that can be employed, quantitative and qualitative. While for the purpose of this research quantitative metrics are preferred, qualitative evaluation can aid with describing model behaviour and reveal crucial information. The most common qualitative evaluation is the Probability Integral Transform (PIT), which is used in combination with a histogram. Given an observation y_t at time t and forecast density f the PIT z_t is given by Equation 5.4 , and should make up a uniform distribution between 0-1 [7]. The Figure 5.1 gives two examples, of a good fit, and a bad fit. This is an incomplete evolution metric, however it can be especially useful during model development.

$$z_t = \int_{-\infty}^{y_t} f(x)dx \tag{5.4}$$

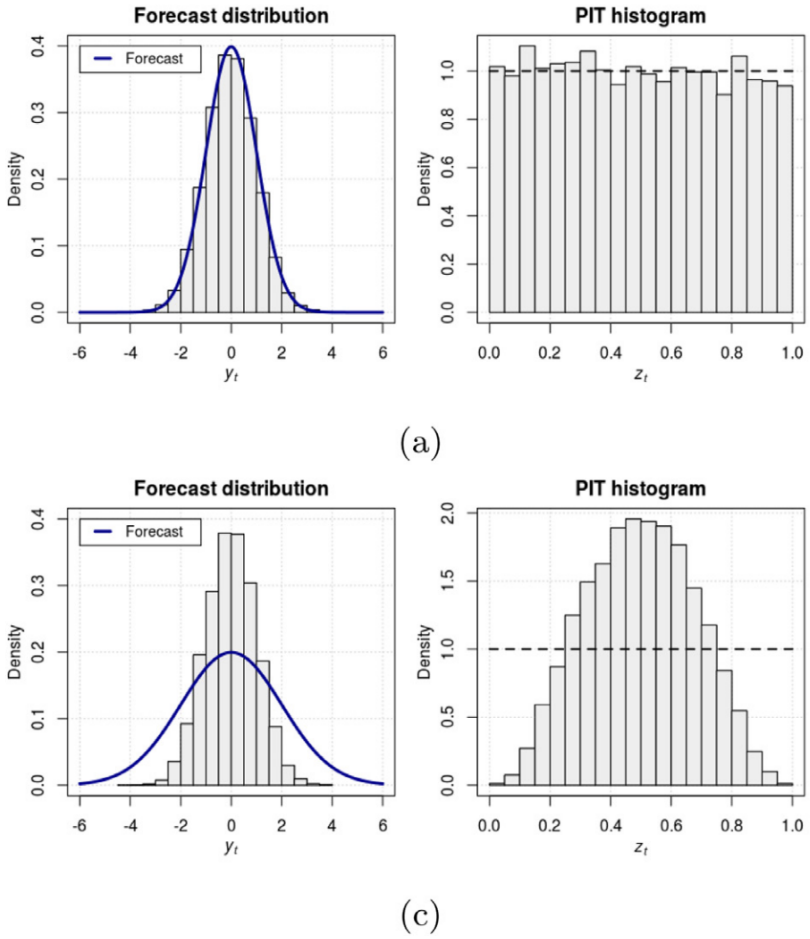


Figure 5.1: a) Shows a good forecast fit, which can be seen by the uniform PIT histogram, c) shows a bad forecasting fit, which is visualised in the PIT histogram by a concentration of values around 0.5[7]

5.2.1. Predictive Interval (PI)

The Predictive Interval (PI) provides an intuitive evaluation of a forecast, it gives the expectation of future values falling within a specified probability range. It is a quantitative analog to the PIT histograms, by specifying a significance level α the percentage occurrence of observations within a confidence range and the size of α can be compared. The most common implementation is the Prediction interval coverage probability (PICP), this metric measures the calibration of the model and is given by Equation 5.5. Effectively if a 50% significance level is chosen, 50% of observations should fall into this range. Mathematically this is given by

$$PICP = \frac{1}{N}\sum_{i=1}^{N} c_i \tag{5.5}$$

Where N is the number of observations and c_i is given by Equation 5.6:

$$c_i = \begin{cases} 1 & \text{if } y_i \in \hat{I}_i^{\alpha} \\ 0 & \text{otherwise} \end{cases} \tag{5.6}$$

Where y_i are observations and $\hat{I}_i^{\alpha}$ is the range of values determined by Equation 5.7

$$\hat{I}_i^{\alpha} = \hat{U}_i^{\alpha} - \hat{L}_i^{\alpha} \tag{5.7}$$

Where $\hat{U}_i^{\alpha}$ and $\hat{L}_i^{\alpha}$ being the upper and lover boundaries of the PI respectively, which are determined by the significance level α [21]. An alternative Prediction interval normalised average width (PINAW) which only evaluates the sharpness of the forecast by calculating the average width of a confidence interval $\hat{I}_i^{\alpha}$ and is given by Equation 5.8.

$$PINAW = \frac{1}{NR}\sum_{i=1}^{N} \hat{I}_i^{\alpha} \tag{5.8}$$

PI approaches are intuitive and easy to understand, which can be useful for non-technical stakeholders, however they are not fully able to capture both calibration and sharpness of the model in a single metric. Therefore they are not as suitable to fully evaluate and compare probabilistic predictions.

5.2.2. Logarithmic scoring rule (LogS)

A commonly used scoring rule in the Bayesian setting is Logarithmic scoring rule (LogS), and a suitable evaluation metric for probabilistic outputs. It measures how likely an observation is given the predictive distribution, giving a lower score to better models. In other words it penalises if an outcome observed has a low probability, and optimises the values to lie close to the mode of the output distribution. The scoring rule is given by Equation 5.9:

$$LogS(f, y) = -log\, f(y) \tag{5.9}$$

Where y_t is a univariate time series and f is the evaluation of a univariate density forecast (the PDF of the distribution) [5]. Which is then evaluated for each observation, providing an overall score. LogS is able to represent both calibration and sharpness, and therefore it is a complete metric. It captures calibration by rewarding forecasts that assign high probability to the actual outcomes, and sharpness by weighting these rewards according to the probability of the observation occurring. However a significant downside of this approach is being very sensitive to outliers, even a small number of outliers can significantly skew results making evaluations and comparisons difficult.

5.2.3. Continuous ranked probability score (CRPS)

A somewhat similar approach to LogS is the Continuous ranked probability score (CRPS). One of the main differences being that CRPS operates on the CDF of the probabilistic forecast, as opposed to the PDF, which have computational benefits. Typically CDF is easier to numerically evaluate than the PDF [7]. CRPS can be thought of as the distance between the empirical distribution of the forecast and the actual outcome, with lower values indicating better models, optimising values for the median of the forecast. The scoring function is given in Equation 5.10.

$$CRPS(F, y) = \int_{-\infty}^{\infty} (F(x) - H(y - x))^2 dx \tag{5.10}$$

Where y is the observation, F is the CDF of the forecast, and H is the Heaviside step function (that evaluates to 0 if the value is negative and 1 otherwise). Similarly to the LogS scoring function CRPS evaluates both the calibration and sharpness of the forecast. This is done by penalising both discrepancies between the

forecast and outcomes (calibration) and penalising wide output densities (sharpness). However it also has it's own limitations, with the need for numerical integration requiring significant computational resources. This could be an issues where the performance of the model needs to be evaluated on a regular basis, or continuously [8]. Secondly it can also be susceptible to extreme values, but in the context of passenger arrival rate forecasting extreme events will not be as extreme as other fields that successfully employ CRPS [58]. And is significantly less sensitive than LogS. Both these downsides can either be mitigated or are not of primary concern for the purpose of comparing models in an academic setting.

5.3. Conclusion

Evaluation of probabilistic forecasting models in not a trivial tasks and there is no silver bullet. However in the context of the forecasting problem CRPS provides a good scoring rule for model evaluation and comparison. While both it and LogS could be suitable evaluation metrics, outliers have a significantly larger impact on LogS and therefore make it less suitable. While PI methods will not be used to directly evaluate the model, their intuitive representation might make them suitable for presenting model performance to non technical stakeholders. Finally during model development PIT histograms will be used for continuous model validation. In conclusion CRPS will be used for the primary scoring function to evaluate the probabilistic forecast due to its ability to represent both the calibration and sharpness of the model. This concludes the selection of technical methods and evaluation metrics, and following chapters will present the academic perspective.

6

Research Proposal

From the literature survey completed, trends and gaps have been identified, and state of the art forecasting methods have been explored. First the problem definition will be presented by reviewing the identified trends in literature and the gap found in section 6.1. This will be followed by presenting the research question, and its sub questions, and the research objective in section 6.2.

6.1. Problem definition

The initial problem set out by GRASP was for a more robust operational level forecasting solution, which would allow for enhanced decision making capabilities. First through the literature review on existing passenger arrival forecasting approaches, a number of trends have been identified, which in turn helped inform the research gap. The following points summarise the most significant and impactful trends and observations:

- **The forecasting problem is increasingly decomposed into prediction problems** - Especially with the relatively recent introduction of boarding card readers the available granularity of data has significantly increased. Additionally this also allowed existing approaches to segregate passengers by the flight they are arriving for. As a consequence the forecasting problem is being decomposed into estimating the arrival rate pattern, as well as the number of expected passengers. This allows a very natural way to utilise the incredibly "rich" temporal information of flight schedules.

- **Uncertainty is nearly never quantified** - Even though the arrival rate of passengers is a highly stochastic progress, especially when finer granulates are desired, existing approaches mostly deal with point forecasts. The lack of uncertainty is especially impactful in the context of decision making where both the certainty of a value, as well as confidence bounds are incredibly valuable.

- **Underutilisation of real-time information** - Even methods that could inherently integrate real time information to improve their short term performance do not discuss this aspect of the problem. Especially for decomposed approaches information about the number of passengers that have already arrived can significantly increase accuracy and decrease uncertainty.

Given these three primary trends and observations, two primary research gaps has been identified, **a lack of quantification of both model and output uncertainty, and the absence of real-time information integration in current passenger forecasting models**. This identified research gap then directed the exploration of the state of the art forecasting methods where three main categories were explored, traditional time series, causal model, and machine learning approaches. Furthermore Bayesian framework was explored in relationship to the previously mentioned approaches since it's benefits closely align with the identified gaps. Reviewing a large number of possible approaches, Bayesian regression and it's variants such as Bayesian hierarchical regression was identified to be best suited to solve the problem. Summarising the above discussed points the following problem definition is presented:

Airport security checkpoints face a significant challenge in managing passenger flow due to fluctuating arrival rates. Throughout the day, the checkpoint should roughly match arriving flow rate of passengers with

a comparable throughput. Failing to do so leads to either frustratingly long wait time for passengers, or increased operational costs for the airport. However both literature and industry rely on static point forecasting methods that are not able to incorporate valuable real time information. Therefore, developing a more accurate forecasting model for the passenger arrival rates that is able to integrate new information and provide more insight into forecasted scenarios has great value. Potentially allowing for better scheduling and therefore reducing wait times for passengers and operational costs for airports alike.

6.2. Research question and objective

From the above problem statement the following research question was derived:

> How can uncertainty in passenger arrival rate forecast be captured and quantified, and then updated in the presence of real time information?

The above questions will be evaluated using the CRPS method identified in chapter 5. Furthermore in order to better answer the research question, and to steer the thesis project in the right direction, the above question has been divided into sub questions. These have been grouped under three main categories.

Model - The first group of sub questions are related to the model, and more specifically relating to the architectural decision that will have to be made.

- What model architecture is best suited to the forecasting problem?
  - How will the forecasting problem be decomposed?
  - How will real time data be integrated into the forecast?
  - How will the probabilistic forecast be represented in the output of the model?
- How to effectively leverage Bayesian methodologies to accurately model passenger arrival rates?
- How to balance model performance with computational requirements?

Features - The second group of sub questions are related to the features and feature selection process, questions of which features and how to integrate them.

- How to identify features that could be used in the model?
  - What flight specific features should be considered? I.e. destination, airline.
  - What temporal features should be considered? I.e. time of day, holidays.
  - What airport features should be considered? I.e. weather, current arrival rate.
- How will complex features be encoded to work as inputs?

Performance - The third group of sub questions are related to the performance and evaluation of the forecasting model.

- How will the performance impact of real-time updating of the model be quantified?
- What baseline model will comparisons be made against?
- How will the results be verified and validated?

6.2.1. Research objective

In order to answer the main research question that was derived from the problem statement, in combination with the exploration of the state of the art forecasting approaches. The following research objective has been established:

> To develop and evaluate a real-time probabilistic security checkpoint arrival rate forecasting model by utilising a bayesian framework.

With the research question and objective defined and expanded, the following chapter will conclude this literature review by discussing a case study as well as the methodology and planning of the thesis.

7

Case study & Methodology

This chapter aims to elaborate and motivate the objective of this research by first discussing the case study that will employ the real time probabilistic forecasting method within the operations of a security checkpoint in section 7.1. After which the methodology, and planning of the whole project will be presented in section 7.2.

7.1. Case study

As discussed in chapter 2 the primary desire for an accurate forecasting model is to enable the creation of decision support systems. There were two problems identified by the airport that could be improved in an operational time frame; determining optimal times to send security agents on their breaks, and shifting agents between checkpoints based on the differential load. Since the first problem is encountered more frequency, and is a simpler problem, it will be chosen to be used as the case study for utilisation of the output of the forecasting model. First the problem will be shorty introduced providing background, motivation, and constraints, then a solution approach will be discussed that will use the probabilistic forecast.

Each security checkpoint will have the number of required open lanes determined for each time block throughout a day well in advance. With each lane requiring a security team to man them, which is done in shifts, usually 4 or 8 hours, with each requiring one and two 15 minute breaks respectively. These breaks must occur roughly within a 1 hour time window during the shifts, and preferably send the whole lane on a break at once, or send individuals if it is busy. Currently this is done reactively by the checkpoint manager, who only evaluates the current state to make a decision. However using the probabilistic forecasting would allow to model different scenarios, to identify and then determine the suitability of potential time slots for breaks. This then could provide valuable additional information to make optimal staffing decision by the checkpoint manager. There are also some additional points of data that will be available to help with making a decision support system. The size of the queue (count of people), and the real time average throughput of each lane is available. Finally to reiterate, the main concern of the airport is to not break the 10 minute maximum queue length performance constraint.

For the purpose of the case study the exact scheduling of each security team and lane is not going to be available, and therefore the goal is not to optimally distribute the break times between all lanes and agents. However utilising the two available data points from the airport, the queue size, and current throughput, in combination with the output of the forecast, a decision support system can be created. The three data sources should allow for prediction of the queue size under different circumstances. The goal of which is to identify periods during which security agents can be sent on their breaks with the lowest probability of breaking the queue length performance constraint.

7.2. Planning and Methodology

Now that the research opportunities have been identified, and a suitable case study has been outlined, a comprehensive plan and methodology will be presented in this section. The required work has been split up into 7 work packages, 3 milestones and 3 reporting and preparation phases, each being allocated an estimated time frame. This timeline excludes the literature study, as it has already been completed with this document. Finally the work packages below have been visualised in a gantt chart in Figure 7.1.

- **WP 1 - Data analysis - 2 weeks**
  - This work package aims to create a data ingestion framework with which raw data from GRASP and the airport can be pre processed and transformed into a suitable format. First the raw data will be cleaned by handling missing data, removing duplicates and ensuring a consistent format. After which exploratory data analysis will be performed to gain better insight into the available data, and identify high level trends and observations.
- **WP 2 - Framework selection - 2 weeks**
  - This work package will first identify suitable programs and programming libraries that allow for the application of Bayesian frameworks. Then a comparison will be made, and a winner chosen, after which the remainder of the allocated time will be spent on becoming proficient with the selected framework.
- **WP 3 - Baseline time series model - 2 weeks**
  - This work package will first create a supporting data pipeline that will allow for easy integration for training and verification of the arrival rate forecasting model. After which a baseline time series model, SARIMA with GARCH errors will be implemented to be used as a point of comparison.
- **WP 4 - Baseline Bayesian model - 2 weeks**
  - This work package will create the baseline Bayesian model for prediction of both the TTD pattern, as well as the number of expected people for each flight. Which then will be used with the flight schedule to generate the full checkpoint arrival rate forecast. Particular focus will be on creating a modular structure that will allow easy evaluation for the feature selection.
- **WP 5 - Feature analysis - 4 weeks**
  - In this work package possible features from the literature are going to be explored by using statistical methods and then evaluated by integration into the baseline Bayesian model. Additionally different encoding approaches will be explored for complex features, trying to balance expressibility with data compactness. With the final goal of this work package being the selection of the set of most impactful features.
- **R&P 1 - Reporting and preparation for midterm meeting - 2 weeks**
- **Milestone 1 - Midterm meeting**
- **WP 6 - Case study implementation - 2 weeks**
  - This work package aims to create the logic required for the identification and evaluation of possible break periods by simulating possible checkpoint state outcomes. The algorithm will estimate future states of the security checkpoint queue, by using the current state, the current processing rate of the checkpoint, as well as the probabilistic forecasted arrival rate of passengers.
- **WP 7 - Case study evaluation - 2 weeks**
  - This work package will evaluate the suitability of break periods identified, and compare it to a naive approach. The naive approach will only observe the current queue state to make decisions about when to send security agents on break.
- **R&P 2 - Reporting and preparation for green light meeting - 2 weeks**
- **Milestone 2 - Green light meeting**
- **R&P 3 - Reporting and prepare for Thesis defence - 4 weeks**
- **Milestone 3 - Thesis defence**

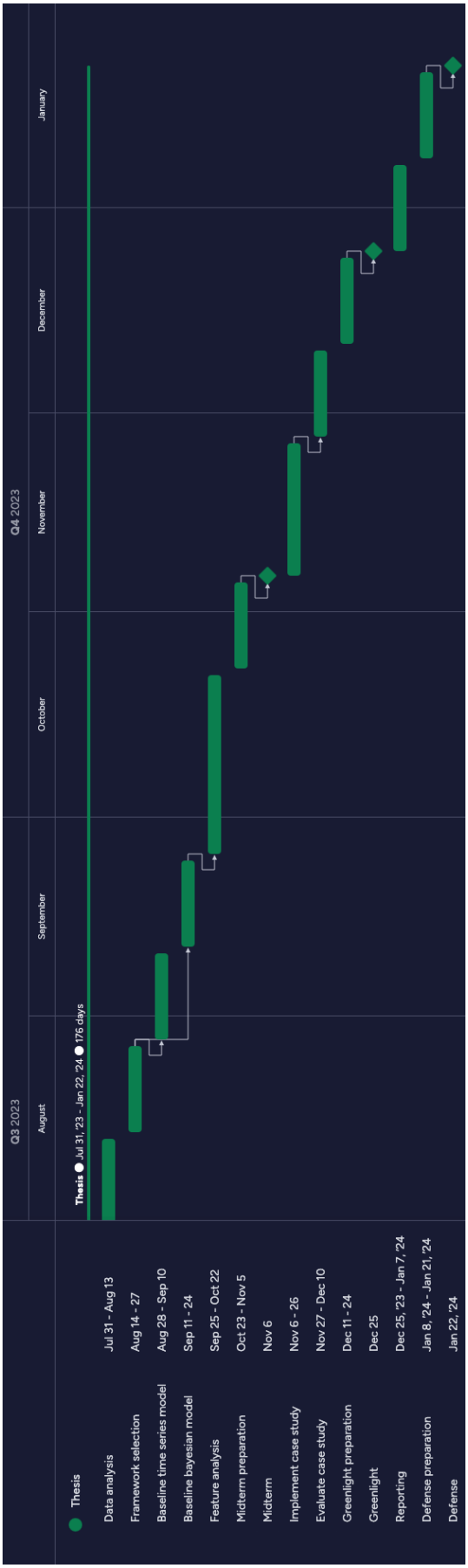


Figure 7.1: Project planning gantt chart

Bibliography

[1] Aykan Akincilar. A Methodology for Shuttle Scheduling in Airports That Ensures Mitigating Arriving Passenger Congestion Under Uncertain Demand. IEEE Intelligent Transportation Systems Magazine, 14 (2):105–114, March 2022. ISSN 1941-1197. doi: 10.1109/MITS.2021.3049359. Conference Name: IEEE Intelligent Transportation Systems Magazine.

[2] Rosa María Arnaldo Valdés, V. Fernando Gómez Comendador, Luis Perez Sanz, and Alvaro Rodriguez Sanz. Prediction of aircraft safety incidents using Bayesian inference and hierarchical structures. Safety Science, 104:216–230, April 2018. ISSN 09257535. doi: 10.1016/j.ssci.2018.01.008. URL https://linkinghub.elsevier.com/retrieve/pii/S0925753517301868.

[3] Norman Ashford. Airport, July 2019. URL https://www.britannica.com/technology/airport.

[4] William A. Belson. Matching and Prediction on the Principle of Biological Classification. Journal of the Royal Statistical Society Series C, 8(2):65–75, 1959. URL https://ideas.repec.org//a/bla/jorssc/v8y1959i2p65-75.html. Publisher: Royal Statistical Society.

[5] Caio Vitor Beojone and Regiane Máximo de Souza. IMPROVING THE SHIFT-SCHEDULING PROBLEM USING NON-STATIONARY QUEUEING MODELS WITH LOCAL HEURISTIC AND GENETIC ALGORITHM. Pesquisa Operacional, 40, May 2020. ISSN 0101-7438, 1678-5142. doi: 10.1590/0101-7438.2020.040.00220764. URL http://www.scielo.br/j/pope/a/6cP5pLKf9tqd4pnfXHCNkLH/abstract/?lang=en. Publisher: Sociedade Brasileira de Pesquisa Operacional.

[6] Christopher M. Bishop. Mixture density networks. 1994. URL https://publications.aston.ac.uk/id/eprint/373/. Num Pages: 26 Place: Birmingham Publisher: Aston University.

[7] Mathias Blicher Bjerregård, Jan Kloppenborg Møller, and Henrik Madsen. An introduction to multivariate probabilistic forecast evaluation. Energy and AI, 4:100058, June 2021. ISSN 2666-5468. doi: 10.1016/j.egyai.2021.100058. URL https://www.sciencedirect.com/science/article/pii/S2666546821000124.

[8] David Bolin and Jonas Wallin. Scale dependence: Why the average CRPS often is inappropriate for ranking probabilistic forecasts. December 2019.

[9] Tim Bollerslev. Generalized autoregressive conditional heteroskedasticity. Journal of Econometrics, 31(3):307–327, April 1986. ISSN 0304-4076. doi: 10.1016/0304-4076(86)90063-1. URL https://www.sciencedirect.com/science/article/pii/0304407686900631.

[10] George E. P. Box and Gwilym M. Jenkins. Time series analysis: forecasting and control. Holden-Day series in time series analysis and digital processing. Holden-Day, San Francisco, rev. ed edition, 1976. ISBN 978-0-8162-1104-3.

[11] Bin Chen, Xing Zhao, and Jin Wu. Evaluating Prediction Models for Airport Passenger Throughput Using a Hybrid Method. Applied Sciences, 13(4):2384, February 2023. ISSN 2076-3417. doi: 10.3390/app13042384. URL https://www.mdpi.com/2076-3417/13/4/2384.

[12] Alexandre G. de Barros and David D. Tomber. Quantitative analysis of passenger and baggage security screening at airports. Journal of Advanced Transportation, 41(2):171–193, 2007. ISSN 2042-3195. doi: 10.1002/atr.5670410204. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/atr.5670410204. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/atr.5670410204.

[13] Choung Do. Gaussian processes, 2007. URL https://see.stanford.edu/materials/aimlcs229/cs229-gp.pdf.

[14] Matthias Döring. Prediction vs Forecasting, December 2018. URL https://www.datascienceblog.net/post/machine-learning/forecasting_vs_prediction/.

[15] Jeffrey L. Elman. Finding structure in time. Cognitive Science, 14(2):179–211, April 1990. ISSN 0364-0213. doi: 10.1016/0364-0213(90)90002-E. URL https://www.sciencedirect.com/science/article/pii/036402139090002E.

[16] Robert F. Engle. Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation. Econometrica, 50(4):987–1007, 1982. ISSN 0012-9682. doi: 10.2307/1912773. URL https://www.jstor.org/stable/1912773. Publisher: [Wiley, Econometric Society].

[17] EUROCONTROL. EUROCONTROL Forecast Update 2021-2027. Technical report, October 2021.

[18] Jay Wright Forrester. Industrial dynamics. M.I.T. Pr, Cambridge, Mass, students' ed., 8. print edition, 1973. ISBN 978-0-262-06003-5.

[19] Fortune Bisuness Insights. Airport Services Market Size, Share, Growth | Global Report, 2027, 2020. URL https://www.fortunebusinessinsights.com/airport-services-market-102855.

[20] David Gillen and William G. Morrison. Aviation security: Costing, pricing, finance and performance. Journal of Air Transport Management, 48:1–12, September 2015. ISSN 0969-6997. doi: 10.1016/j.jairtraman.2014.12.005. URL https://www.sciencedirect.com/science/article/pii/S0969699714001537.

[21] J. M. González-Sopeña, V. Pakrashi, and B. Ghosh. An overview of performance evaluation metrics for short-term statistical wind power forecasting. Renewable and Sustainable Energy Reviews, 138:110515, March 2021. ISSN 1364-0321. doi: 10.1016/j.rser.2020.110515. URL https://www.sciencedirect.com/science/article/pii/S1364032120308005.

[22] Hengsheng Gu. Airport Revenue Diversification. Journal of Management Science & Engineering research, 2(1), August 2019. ISSN 2630-4953. doi: 10.30564/jmser.v2i1.1122. URL https://www.bilpublishing.com/index.php/jmser/article/view/1122.

[23] Xiaojia Guo, Yael Grushka-Cockayne, and Bert De Reyck. Forecasting Airport Transfer Passenger Flow Using Real-Time Data and Machine Learning. Manufacturing & Service Operations Management, 24(6): 3193–3214, August 2020. ISSN 1523-4614. doi: 10.1287/msom.2021.0975. URL https://pubsonline.informs.org/doi/full/10.1287/msom.2021.0975. Publisher: INFORMS.

[24] Önder Gürcan, Oguz Dikenelli, and Carole Bernon. A generic testing framework for agent-based simulation models. Journal of Simulation, vol. 7:pp. 183–201, March 2015. doi: 10.1057/jos.2012.26. URL https://hal.archives-ouvertes.fr/hal-01128680. Publisher: Palgrave Macmillan.

[25] SeungYeob Han and Daniel DeLaurentis. A Bayesian Analysis (Gaussian Process Model) for Air Traffic Demand Forecast at a Commercial Airport. In 10th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference, Fort Worth, Texas, September 2010. American Institute of Aeronautics and Astronautics. ISBN 978-1-62410-159-5. doi: 10.2514/6.2010-9218. URL https://arc.aiaa.org/doi/10.2514/6.2010-9218.

[26] Girish Jampani Hanumantha, Berkin T. Arici, Jorge A. Sefair, and Ronald Askin. Demand prediction and dynamic workforce allocation to improve airport screening operations. IISE Transactions, 52(12):1324–1342, December 2020. ISSN 2472-5854. doi: 10.1080/24725854.2020.1749765. URL https://doi.org/10.1080/24725854.2020.1749765. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/24725854.2020.1749765.

[27] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-term Memory. Neural computation, 9:1735–80, December 1997. doi: 10.1162/neco.1997.9.8.1735.

[28] Stef Janssen, Anne-Nynke Blok, and Arthur Knol. AATOM - An Agent-based Airport Terminal Operations Model. 2019.

[29] Irina Kalinina and Aleksandr Gozhyj. Modeling and forecasting of nonlinear nonstationary processes based on the Bayesian structural time series. 5(3):240–255, October 2022. ISSN 2663-7723. URL http://aait.ccs.od.ua/index.php/journal/article/view/148. Number: 3.

[30] Matthew G Karlaftis. DEMAND FORECASTING IN REGIONAL AIRPORTS: DYNAMIC TOBIT MODELS WITH GARCH ERRORS. 2008.

[31] Antonín Kazda and Robert E Caves. Airport Design and Operation. 3. 2015.

[32] Denis Kirchhübel and Thomas Martini Jørgensen. Generating Diagnostic Bayesian Networks from Qualitative Causal Models. In 2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), pages 1239–1242, September 2019. doi: 10.1109/ETFA.2019.8869461. ISSN: 1946-0759.

[33] Alan (Avi) Kirschenbaum. The cost of airport security: The passenger dilemma. Journal of Air Transport Management, 30:39–45, July 2013. ISSN 09696997. doi: 10.1016/j.jairtraman.2013.05.002. URL https://linkinghub.elsevier.com/retrieve/pii/S0969699713000458.

[34] S. B. Kotsiantis. Decision trees: a recent overview. Artificial Intelligence Review, 39(4):261–283, April 2013. ISSN 1573-7462. doi: 10.1007/s10462-011-9272-4. URL https://doi.org/10.1007/s10462-011-9272-4.

[35] Jitendra Kumar, Rimsha Goomer, and Ashutosh Kumar Singh. Long Short Term Memory Recurrent Neural Network (LSTM-RNN) Based Workload Forecasting Model For Cloud Datacenters. Procedia Computer Science, 125:676–682, 2018. ISSN 18770509. doi: 10.1016/j.procs.2017.12.087. URL https://linkinghub.elsevier.com/retrieve/pii/S1877050917328557.

[36] Klemens Köstler and Alexei Sharpanskykh. Simulating an Scheduling Airport Security Checkpoints: Q-Learning-Based Allocation of Operators to Security Teams at an Airport Security Checkpoint. 2021. URL https://repository.tudelft.nl/islandora/object/uuid%3A4269ed22-debe-432c-8c0f-ac7127341001.

[37] Ma Nang Laik, Murphy Choy, and Prabir Sen. Predicting Airline Passenger Load: A Case Study. In 2014 IEEE 16th Conference on Business Informatics, volume 1, pages 33–38, July 2014. doi: 10.1109/CBI.2014.39. ISSN: 2378-1971.

[38] Sik-Yum Lee and Xin-Yuan Song. Bayesian structural equation model. WIREs Computational Statistics, 6(4):276–287, 2014. ISSN 1939-0068. doi: 10.1002/wics.1311. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/wics.1311. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/wics.1311.

[39] Ziyu Li, Jun Bi, and Zhiyin Li. Passenger Flow Forecasting Research for Airport Terminal Based on SARIMA Time Series Model. IOP Conference Series: Earth and Environmental Science, 100(1):012146, December 2017. ISSN 1755-1315. doi: 10.1088/1755-1315/100/1/012146. URL https://dx.doi.org/10.1088/1755-1315/100/1/012146. Publisher: IOP Publishing.

[40] Lin Lin, Xiaochen Liu, Xiaohua Liu, Tao Zhang, and Yang Cao. A prediction model to forecast passenger flow based on flight arrangement in airport terminals. Energy and Built Environment, June 2022. ISSN 2666-1233. doi: 10.1016/j.enbenv.2022.06.006. URL https://www.sciencedirect.com/science/article/pii/S2666123322000423.

[41] Lijuan Liu and Rung-Ching Chen. A novel passenger flow prediction model using deep learning methods. Transportation Research Part C: Emerging Technologies, 84:74–91, November 2017. ISSN 0968-090X. doi: 10.1016/j.trc.2017.08.001. URL https://www.sciencedirect.com/science/article/pii/S0968090X17302024.

[42] Yunhao Liu, Gengzhong Feng, Kwai-Sang Chin, Shaolong Sun, and Shouyang Wang. Daily tourism demand forecasting: the impact of complex seasonal patterns and holiday effects. Current Issues in Tourism, 26(10):1573–1592, May 2023. ISSN 1368-3500. doi: 10.1080/13683500.2022.2060067. URL https://doi.org/10.1080/13683500.2022.2060067. Publisher: Routledge _eprint: https://doi.org/10.1080/13683500.2022.2060067.

[43] Gianluca Manzo. Agent-based Models and Causal Inference | Wiley, 2022. URL https://www.wiley.com/en-gb/Agent+based+Models+and+Causal+Inference-p-9781119704461.

[44] L. E. Melkumova and S. Ya. Shatskikh. Comparing Ridge and LASSO estimators for data analysis. Procedia Engineering, 201:746–755, January 2017. ISSN 1877-7058. doi: 10.1016/j.proeng.2017.09.615. URL https://www.sciencedirect.com/science/article/pii/S1877705817341474.

[45] James Mitchell and Kenneth F. Wallis. Evaluating density forecasts: forecast combinations, model mixtures, calibration and sharpness. Journal of Applied Econometrics, 26(6):1023–1040, 2011. ISSN 1099-1255. doi: 10.1002/jae.1192. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/jae.1192. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/jae.1192.

[46] Zahra Mohaghegh. Combining System Dynamics and Bayesian Belief Networks for Socio-Technical Risk Analysis. In 2010 IEEE International Conference on Intelligence and Security Informatics, pages 196–201, May 2010. doi: 10.1109/ISI.2010.5484736.

[47] Philippe Monmousseau, Gabriel Jarry, Florian Bertosio, Daniel Delahaye, and Marc Houalla. Predicting Passenger Flow at Charles De Gaulle Airport Security Checkpoints. In 2020 International Conference on Artificial Intelligence and Data Analytics for Air Transportation (AIDA-AT), pages 1–9, February 2020. doi: 10.1109/AIDA-AT48540.2020.9049190.

[48] Raha Moraffah, Paras Sheth, Mansooreh Karami, Anchit Bhattacharya, Qianru Wang, Anique Tahir, Adrienne Raglin, and Huan Liu. Causal Inference for Time series Analysis: Problems, Methods and Evaluation, February 2021. URL http://arxiv.org/abs/2102.05829. arXiv:2102.05829 [cs, stat].

[49] James N. Morgan and John A. Sonquist. Problems in the Analysis of Survey Data, and a Proposal. Journal of the American Statistical Association, 58(302):415–434, June 1963. ISSN 0162-1459, 1537-274X. doi: 10.1080/01621459.1963.10500855. URL http://www.tandfonline.com/doi/abs/10.1080/01621459.1963.10500855.

[50] Maria Nadia Postorino, Luca Mantecchini, Caterina Malandri, and Filippo Paganelli. Airport Passenger Arrival Process: Estimation of Earliness Arrival Functions. Transportation Research Procedia, 37:338–345, January 2019. ISSN 2352-1465. doi: 10.1016/j.trpro.2018.12.201. URL https://www.sciencedirect.com/science/article/pii/S2352146518306173.

[51] Priyamvada and Rajesh Wadhvani. Review on various models for time series forecasting. In 2017 International Conference on Inventive Computing and Informatics (ICICI), pages 405–410, November 2017. doi: 10.1109/ICICI.2017.8365383.

[52] Álvaro Rodríguez-Sanz, Alberto Fernández de Marcos, Javier A. Pérez-Castán, Fernando Gómez Comendador, Rosa Arnaldo Valdés, and Ángel París Loreiro. Queue behavioural patterns for passengers at airport terminals: A machine learning approach. Journal of Air Transport Management, 90:101940, January 2021. ISSN 0969-6997. doi: 10.1016/j.jairtraman.2020.101940. URL https://www.sciencedirect.com/science/article/pii/S0969699720305238.

[53] Jakob Runge, Sebastian Bathiany, Erik Bollt, Gustau Camps-Valls, Dim Coumou, Ethan Deyle, Clark Glymour, Marlene Kretschmer, Miguel Mahecha, Jordi Muñoz, Egbert Nes, Jonas Peters, Rick Quax, Markus Reichstein, Marten Scheffer, Bernhard Schölkopf, Peter Spirtes, George Sugihara, Jie Sun, and Jakob Zscheischler. Inferring causation from time series in Earth system sciences. Nature Communications, 10, December 2019. doi: 10.1038/s41467-019-10105-3.

[54] Steven L. Scott and Hal R. Varian. Predicting the present with Bayesian structural time series. International Journal of Mathematical Modelling and Numerical Optimisation, 5(1-2):4–23, January 2014. ISSN 2040-3607. doi: 10.1504/IJMMNO.2014.059942. URL https://www.inderscienceonline.com/doi/abs/10.1504/IJMMNO.2014.059942. Publisher: Inderscience Publishers.

[55] Matthias Seeger. GAUSSIAN PROCESSES FOR MACHINE LEARNING. International Journal of Neural Systems, 14:69–106, 2004. doi: 10.1142/S0129065704001899. URL https://www.worldscientific.com/doi/epdf/10.1142/S0129065704001899.

[56] Ratna Sulistyowati, Suhartono, Heri Kuswanto, Setiawan, and Erni Tri Astuti. Hybrid forecasting model to predict air passenger and cargo in Indonesia. In 2018 International Conference on Information and Communications Technology (ICOIACT), pages 442–447, March 2018. doi: 10.1109/ICOIACT.2018.8350816.

[57] Erma Suryani, Shuo-Yan Chou, and Chih-Hsien Chen. Air passenger demand forecasting and passenger terminal capacity expansion: A system dynamics framework. Expert Systems with Applications, 37(3):2324–2339, March 2010. ISSN 0957-4174. doi: 10.1016/j.eswa.2009.07.041. URL https://www.sciencedirect.com/science/article/pii/S0957417409007076.

[58] Maxime Taillardat, Anne-Laure Fougères, Philippe Naveau, and Raphaël de Fondeville. Evaluating probabilistic forecasts of extremes using continuous ranked probability score distributions. International Journal of Forecasting, 39(3):1448–1459, July 2023. ISSN 01692070. doi: 10.1016/j.ijforecast.2022.07.003. URL http://arxiv.org/abs/1905.04022. arXiv:1905.04022 [math, stat].

[59] Jia Hao Tan and Tariq Masood. Adoption of Industry 4.0 technologies in airports – A systematic literature review, December 2021. URL http://arxiv.org/abs/2112.14333. arXiv:2112.14333 [cs, eess].

[60] Didier van der Horst and Alexei Sharpanskykh. An improved Tabu Search for optimising the con- figuration of an agent-based simulation model of a novel security checkpoint. page 163, 2021.

[61] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In Advances in Neural Information Processing Systems, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

[62] Martin Vikse and Robin Fjørtoft. Finding and Eliminating Bottlenecks: A Case Study of the Security Checkpoint at Oslo Lufthavn Terminal 2. PhD thesis, May 2019. URL https://himolde.brage.unit.no/himolde-xmlui/bitstream/handle/11250/2628132/master_vikse.pdf?sequence=1.

[63] Paul Pao-Yen Wu and Kerrie Mengersen. A review of models and model usage scenarios for an airport complex system. Transportation Research Part A: Policy and Practice, 47:124–140, January 2013. ISSN 0965-8564. doi: 10.1016/j.tra.2012.10.015. URL https://www.sciencedirect.com/science/article/pii/S0965856412001541.

[64] Zhiwei Xing, Ruohong Ling, Chao Wang, and Biao Li. Prediction of passenger arrival distribution in security area based on MRBF-GMM model. In Sixth International Conference on Electromechanical Control Technology and Transportation (ICECTT 2021), volume 12081, pages 1015–1021. SPIE, February 2022. doi: 10.1117/12.2623837. URL https://www.spiedigitallibrary.org/conference-proceedings-of-spie/12081/1208130/Prediction-of-passenger-arrival-distribution-in-security-area-based-on/10.1117/12.2623837.full.

[65] Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are Transformers Effective for Time Series Forecasting?, August 2022. URL http://arxiv.org/abs/2205.13504. arXiv:2205.13504 [cs].