Airport Simulation

Project by:
Raymond C Li (cli420@gatech.edu)
Junyang Tang (jtang375@gatech.edu)

CS 6730
Model and Simulation

TABLE OF CONTENTS

Abstract

(GITHUB LINK)

The goal of this project was to create a simulation of an airport terminal environment in order to study the movement of passengers through various airport checkpoints. The procedures that we used in this simulation are the check-in and security screenings of the passengers. By using a neural network, we were able to forecast passengers' arrival timing based on number of workers at the airport and simulated their waiting times while going through the various processes. Data on the passengers waiting times were gathered based on the flights of the day, the number of workers in each checkpoint as well as other factors. According to simulation results, the quantity of available staff had a substantial influence on passengers' wait times.

CHAPTER 1
INTRODUCTION

1.1 Goal

The goal of this project is to simulate the passenger flow through an airport terminal by predicting the arrival patterns of passengers and wait time at different stages of their journey, such as check-in and security checkpoints. By better understanding these patterns, airport management can allocate resources more efficiently, such as staff and equipment, to minimize waiting times and improve the overall passenger experience. This project aims to model the real-world phenomenon of passengers arriving at an airport, checking in, and passing through security checkpoints.

1.2 Relevant phenomenon

- Flight Schedules: Departure time
- Passenger arrival patterns at the airport, assuming passenger arrival pattern is significantly influenced by flight schedule and availability of airport staff.
- Pre-boarding processes, including check-in and security screening processes, which can vary depending on the number of staff available, check-in and security methods used, etc.

1.3 Interpretation of phenomenon

For a typical airport experience, passengers may arrive at the airport a few hours before their flight's scheduled departure. Upon arrival, they proceed to the check-in area, where they either use a self-service kiosk or approach an airline counter to check in for their flight,

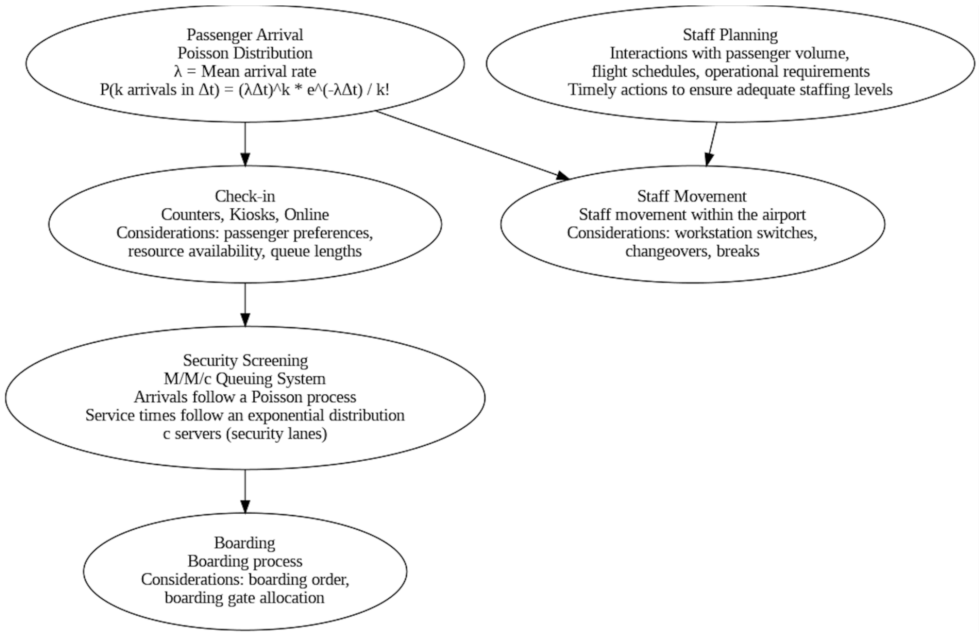


Figure 1.1: Illustrated process diagram

drop off their luggage, and receive their boarding passes.

After completing the check-in process, passengers move on to the security checkpoint, where they are required to present their boarding passes and a form of identification. They then undergo a security screening, which involves placing their carry-on items, such as bags and electronic devices, through an X-ray scanner, and walking through a metal detector or body scanner. The purpose of this process is to ensure that passengers are not carrying prohibited items on board the aircraft.

The time it takes for passengers to complete these processes can vary widely, depending on factors such as the number of staff working at the check-in counters and security checkpoints, the efficiency of the equipment used, and the number of passengers arriving at the airport at any given time. During peak travel times, such as holidays or weekends, the volume of passengers can lead to long queues and waiting times, causing stress and frustration for travelers.

In this project, we used machine learning techniques to predict the arrival patterns of passengers at different stages of their airport journey. We trained a neural network model using historical data on flight schedules, passenger arrival times, and the time it takes to complete the check-in and security processes. By accurately predicting these patterns, airport management can better allocate resources, such as staff and equipment, to minimize waiting times and improve the overall passenger experience.

To achieve this, we used a dataset containing information on flight schedules and passenger arrival times, as well as other relevant variables. We preprocessed the data by split it into training and testing sets, and trained a neural network model to predict the arrival patterns of passengers. Once the model neural network was trained, we used it to evaluate its performance based on passenger wait times after security and how many passengers were late.

In conclusion, this project aims to improve the airport experience for passengers by using machine learning to predict and optimize passenger flow through the terminal. By better understanding and anticipating passenger arrival patterns, airport management can allocate resources more efficiently and minimize waiting times, resulting in a more enjoyable travel experience for all.

CHAPTER 2

LITERATURE REVIEW

Airport operations simulations are crucial for analyzing and optimizing the performance of airport systems. Over the years, numerous researchers have developed models, simulators, and simulation techniques to simulate various aspects of airport operations. While these simulations have been helpful in understanding and improving airport operations, there is still room for improvement.

One widely adopted approach in airport operations simulations is the use of discrete event simulation (DES) models. These models simulate airport operations in real-time by breaking down events into discrete time intervals. For example, Kierzkowski and Kisiel [1] developed a simulation model of the security control system operation at Wrocław Airport, a regional airport situated in south-western Poland in Europe. The paper presents a discrete event simulation model implemented in the FlexSim environment. The model uses input data such as the structure of the security control system, the procedure of implementing the security control process, and the flight schedule to analyze the efficiency of the system.

Another noteworthy work in DES is by Pendergraft, Robertson, & Shrader [2], who employed a discrete event simulation model to analyze the impact of an increased number of flights and passengers on airport performance. Their model highlighted the need for efficient resource management and infrastructure development to maintain high service quality in the face of increasing demand.

Another popular approach is agent-based modeling (ABM), which simulates the behavior of individual agents within a system. Several studies have used ABM to simulate airport operations, including Janssen, Sharpanskykh, & Curran [3] who developed operational and security models using ABM to simulate security risks and risk management implementation. Wu and Chen [4], who examined the effects of passenger characteristics and terminal layout on airport retail revenue through an agent-based simulation approach. Their study highlights the potential for utilizing ABM to gain valuable insights into optimizing airport retail revenue by understanding the complex interplay of passenger behavior and terminal design.

While these models have provided valuable insights into airport operations, they have limitations. For example, DES models may oversimplify the behavior of passengers and aircraft, while ABM models may overlook the impact of resource constraints on airport operations. Therefore, there is a need to develop more comprehensive models that integrate multiple simulation techniques. To address these limitations, we plan to develop a new simulation model that integrates both DES and ABM techniques. Our model will simulate the movement of individual passengers and aircraft through an airport system while taking into account factors such as passenger behavior, staff scheduling, and resource constraints. By using a combined DES-ABM approach, we aim to create a more realistic and accurate simulation of airport operations that can help airport managers optimize their systems.

In addition to the integration of multiple simulation techniques, our model will incorporate machine learning algorithms to optimize airport operations. For example, our model can be used to predict passenger flow, identify potential bottlenecks, and optimize staff schedules based on historical data. These predictions can help airport managers make informed decisions and improve the overall efficiency of airport operations.

Machine learning applications in airport operations are not new. For example, Kim et al. [5] used a deep learning approach to predict flight arrival delays, while Ariyawansa and Aponso [6] proposed some ways to employ machine learning algorithms to optimize airport operations. By integrating machine learning into our combined DES-ABM model, we can leverage the power of data-driven techniques to enhance airport operations simulation further.

In summary, while there have been many existing models and techniques used in airport operations simulations, there is still a need for more comprehensive and accurate models.

By integrating multiple simulation techniques and incorporating machine learning algorithms, we hope to improve upon existing airport operations simulations and provide airport managers with a more effective tool for optimizing their systems.

CHAPTER 3

CONCEPTUAL MODEL

3.1 Entities

The main entities in our model are:

- Passenger
- Check-in Personnel
- Security Personnel
- Flight
- Check-in Queue
- Security Queue

3.1.1 Passenger

The Passenger entity represents an individual traveling through the airport. Each passenger has a specific arrival time, airline, and flight departure time. The attributes of the Passenger entity include:

- Arrival time ($t_{arrival}$): The time at which the passenger arrives at the airport, represented as minutes from a reference time (e.g., midnight).
- Flight departure time ($t_{departure}$): The scheduled time for the passenger's flight to depart, represented as minutes from the reference time.
- Check-in time ($t_{checkin}$): The time it takes for the passenger to complete the check-in process, which can be modeled as a random variable with a specified distribution (e.g., normal, exponential) depending on the airport and airline.

- Security check time ($t_{security}$): The time it takes for the passenger to complete the security check process, which can also be modeled as a random variable with a specified distribution.

3.1.2 Check-in Personnel

The Check-in Personnel entity represents staff responsible for assisting passengers during the check-in process. These personnel are allocated to different airlines and have a limited number of positions available at a time. The attributes of the Check-in Personnel entity include:

- Airline (a): The airline the check-in personnel are assigned to.
- Availability status (s_{avail}): A binary variable indicating whether the check-in personnel is available to serve passengers (1) or occupied (0).
- Number of passengers served (n_{served}): A counter that tracks the number of passengers the check-in personnel has served during a given time period.

3.1.3 Security Personnel

The Security Personnel entity represents staff responsible for conducting security checks on passengers. These personnel work in security checkpoints, and their availability is limited by the number of security lanes. The attributes of the Security Personnel entity include:

- Availability status (s_{avail}): A binary variable indicating whether the security personnel is available to conduct security checks (1) or occupied (0).
- Number of passengers served (n_{served}): A counter that tracks the number of passengers the security personnel has served during a given time period.

3.1.4 Flight

The Flight entity represents a scheduled flight departing from the airport. Each flight has an associated airline and departure time. The attributes of the Flight entity include:

- Airline (a): The airline operating the flight.
- Departure time ($t_{departure}$): The scheduled time for the flight to depart, represented as minutes from the reference time.

3.1.5 Check-in Queue

The Check-in Queue entity represents the waiting line for passengers to complete the check-in process. There is a separate queue for each airline, and passengers join the queue based on their arrival time. The attributes of the Check-in Queue entity include:

- Airline (a): The airline associated with the check-in queue.
- Number of passengers in the queue (n_{queue}): A counter that tracks the current number of passengers waiting in the check-in queue.
- Total waiting time (T_{wait}): The cumulative waiting time for all passengers in the queue, which can be used to compute the average waiting time for the check-in process.

3.1.6 Security Queue

The Security Queue entity represents the waiting line for passengers to undergo security checks. Passengers join the queue after completing the check-in process. The attributes of the Security Queue entity include:

- Number of passengers in the queue (n_{queue}): A counter that tracks the current number of passengers waiting in the security queue.

- Total waiting time (T_{wait}): The cumulative waiting time for all passengers in the queue, which can be used to compute the average waiting time for the security check process.

Security Lanes

Each Security Queue is divided into multiple security lanes, which are managed by individual Security Personnel. The number of security lanes determines the throughput of passengers at the security checkpoint. The attributes of the Security Lane entity include:

- Lane number (l): The unique identifier for each security lane within the Security Queue.
- Security Personnel assigned (sp): The Security Personnel assigned to the specific lane.
- Number of passengers in the lane (n_{lane}): A counter that tracks the current number of passengers waiting in the security lane.
- Total waiting time in the lane (T_{wait_lane}): The cumulative waiting time for all passengers in the security lane, which can be used to compute the average waiting time for the security check process in that specific lane.

These entities, along with their attributes and interactions, form the foundation of our airport operations simulation model. By simulating the movement of individual passengers, the allocation of check-in and security personnel, and the dynamics of queues, we can create a detailed and realistic representation of airport operations. This comprehensive model can help airport managers analyze and optimize their systems, taking into account factors such as passenger behavior, staff scheduling, and resource constraints.

3.2 Events

The main events in our airport operations simulation are:

- Passenger Arrival
- Check-in Process Start
- Check-in Process End
- Security Process Start
- Security Process End
- Flight Departure

3.2.1 Passenger Arrival

The Passenger Arrival event represents a passenger arriving at the airport and joining the check-in queue for their respective airline. This event is triggered based on the passenger's arrival time distribution.

3.2.2 Check-in Process Start

The Check-in Process Start event occurs when a passenger reaches the front of the check-in queue and starts the check-in process with an available check-in personnel. This event marks the beginning of the passenger's interaction with the check-in personnel.

3.2.3 Check-in Process End

The Check-in Process End event occurs when a passenger has completed the check-in process. This event triggers the passenger to leave the check-in queue and join the security queue. The check-in personnel who served the passenger now becomes available to serve the next passenger in the queue.

3.2.4 Security Process Start

The Security Process Start event represents a passenger reaching the front of the security queue and beginning the security check process with an available security personnel. This event initiates the passenger's interaction with the security personnel.

3.2.5 Security Process End

The Security Process End event occurs when a passenger has completed the security check process. This event causes the passenger to leave the security queue and proceed to their boarding gate. The security personnel who conducted the security check now becomes available to serve the next passenger in the queue.

3.2.6 Flight Departure

The Flight Departure event represents a scheduled flight leaving the airport. This event is triggered based on the flight's departure time. Passengers who have completed the check-in and security processes for this flight will have boarded the aircraft by this time.

In addition to these main events, there are several utility events in the software implementation, such as updating the passenger arrival distribution, updating the status of check-in and security personnel, and synchronizing events between different components of the simulation model. These utility events help maintain the accuracy and realism of the simulation as it evolves over time.

3.3 Activities

3.3.1 Passenger

The major activities for a passenger are:

- Arrive at the airport

- Join check-in queue
- Complete check-in process
- Join security queue
- Complete security check
- Proceed to boarding gate

3.3.2 Check-in Personnel

The main activities for check-in personnel include:

- Serve passengers in the check-in queue
- Complete check-in process for passengers
- Update availability status

3.3.3 Security Personnel

The primary activities for security personnel consist of:

- Serve passengers in the security queue
- Complete security check for passengers
- Update availability status

3.3.4 Flight

The main activities for a flight are:

- Board passengers
- Depart from the airport

3.3.5 Check-in Queue

The main activities in the check-in queue involve:

- Add passengers to the queue based on arrival time
- Remove passengers from the queue upon completion of the check-in process
- Update the number of passengers and total waiting time in the queue

3.3.6 Security Queue

The primary activities in the security queue include:

- Add passengers to the queue after completing the check-in process
- Remove passengers from the queue upon completion of the security check
- Update the number of passengers and total waiting time in the queue

Security Lanes

The main activities within the security lanes involve:

- Assign passengers to specific security lanes
- Manage passenger flow through the lanes
- Update the number of passengers and total waiting time in each lane

These activities, along with the events and entities described earlier, make up the airport operations simulation model. By simulating the various activities and their interactions, the model can provide valuable insights into the performance of airport operations and inform decisions on resource allocation and process improvements.

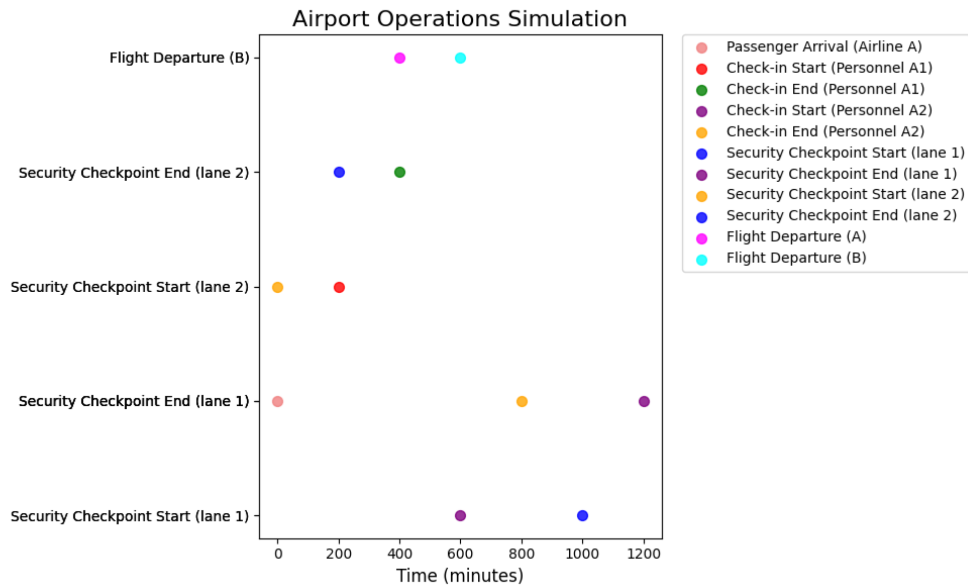


Figure 3.1: Simulation process

3.4 Mathematical Formulations

3.4.1 Passenger movement

We will implement models for the following sub processes to simulate airport operations:

i) **Passenger Arrival:** We have already discussed the Poisson distribution for modeling passenger arrivals. We will use this model to generate random passenger arrival patterns based on historical data.

ii) **Check-in:** For the check-in process, we can use a multi-server queuing model for each check-in option (counters, kiosks, online check-ins). Suppose a mean arrival rate λ_i and mean service rate μ_i for each check-in option i. We can use the Erlang-C formula to calculate the probability $P(W_i > 0)$ that a passenger has to wait for service at check-in option i. Notice that online check-ins do not require any servers, and we may assume it has an extremely high service rate, and imagine this process as a single queue.

$$P(W_i > 0) = \frac{\frac{(c_i\rho_i)^{c_i}}{c_i!\cdot(1-\rho_i)}}{\sum_{k=0}^{c_i-1}\frac{(c_i\rho_i)^k}{k!} + \frac{(c_i\rho_i)^{c_i}}{c_i!\cdot(1-\rho_i)}}$$

In this formula: c_i is the number of servers (check-in counters or kiosks) for check-in option i.

ρ_i is the traffic intensity for check-in option i, which is the ratio of the arrival rate.

λ_i to the product of the number of servers c_i and the service rate μ_i. It represents the proportion of time that the servers are busy. After obtaining the probability of waiting time greater than zero, we can obtain the wait time for each passenger:

$$W_{q_i} = \frac{P(W_i > 0)}{c_i\mu_i(1-\rho_i)} \times \frac{1}{\mu_i}$$

This formula calculates the average waiting time in the queue by dividing the multiplication of the probability of waiting, $P(W_i > 0)$, and the inverse of the service rate, $\frac{1}{\mu_i}$, which represents the average service time, by the product of the number of servers, the service rate, and the idle time (1 - traffic intensity). The formula takes into account the impact of the number of servers, the service rate, and the traffic intensity on the waiting time in the queue.

iii) **Security screening:** For the security screening process, we will still use the M/M/c queuing model. The model will incorporate the following factors:

- Arrival rate of passengers at the security checkpoint
- Service rate of each security lane (processing time per passenger)
- Number of security lanes (servers) available
- What type of security they need to get through (Normal or PreCheck)

3.4.2 Staff simulation

i) Staff movement: We plan to simulate staff movement. Each staff member will move according to whichever station is needed. Although it is assumed that the people at the check-in counters require a different skill set than the people at the security check. The staff will move between each of the check-in counters based on the amount of people and the wait times in each of the queues to minimize each of the wait times for passengers.

CHAPTER 4

SIMULATION MODEL

The given code is an implementation of a simulation model for predicting passenger arrival times at an airport based on various factors like airline, departure time, check-in option, and security option. The code can be divided into three main parts:

1. Data Preprocessing and Generation
2. Airport Simulation
3. Neural Network Model for Passenger Arrival Prediction

Data Preprocessing and Generation:

This part of the code involves preprocessing the input flight data and generating synthetic passenger data for each flight. It consists of the following steps:

- Extracting flights for the given month, day, and airport.
- Removing rows with missing departure times and calculating minutes since midnight.
- Initializing customer IDs and dictionaries to store customer information.
- Generating passenger data for each flight, including airline, departure time, check-in option, and security option.
- Creating binary indicator lists for airline, check-in, and security options.
- Combining the binary indicators to form input tensors for the neural network model.

Airport Simulation:

This part of the code simulates the airport processes, including check-in and security checks for passengers. The following elements are involved in the airport simulation:

- Initializing the simulation environment using SimPy.
- Creating check-in resources for each airline with the specified capacity.
- Creating security checkpoint resources with the specified capacity.
- Defining the Passenger class, which encapsulates the check-in and security check processes.
- Implementing the agent arrival process, which iterates through the list of customers, creates a Passenger instance with the customer data, and starts the check-in process for the passenger.
- Executing the simulation until the end of the day (2880 minutes).

Neural Network Model for Passenger Arrival Prediction:

This part of the code defines and trains a neural network model for predicting passenger arrival times at the airport. The neural network architecture consists of the following layers:

- Integer input layer and binary input layer for processing integer and binary input data.
- Hidden layers for processing combined input data.
- Output layer for generating the final prediction.

The neural network is trained using Stochastic Gradient Descent (SGD) optimizer and Mean Squared Error (MSE) loss. The training loop consists of the following steps:

1. Generating input data for the neural network and customer data for the simulation.

2. Passing input data through the neural network to obtain predictions.
3. Updating the customer data with the predicted arrival times.
4. Running the airport simulation with the updated customer data.
5. Processing the simulation results and updating the labels for the neural network.
6. Calculating the loss for the current epoch.
7. Performing backpropagation and optimizing the model.
8. Saving the updated model.

After the training, the model can be used to predict passenger arrival times at the airport based on the given input features.

CHAPTER 5

VERIFICATION AND VALIDATION

To verify and validate our simulator, we conducted experiments that aimed to evaluate its accuracy and reliability in predicting passenger arrival times at an airport. The experimental procedure, validation process, input modeling, and output analysis are described below.

5.1 Experimental Procedure

1. We selected a specific test day and set the number of check-in and security personnel to ensure adequate resources for the simulation.
2. We generated input data for the neural network model and customer data for the simulation based on the given test day, month, and airport.
3. We passed the input data through the neural network to obtain predicted arrival times for passengers.
4. We updated the customer data with the predicted arrival times and sorted the passengers by their arrival times.
5. We ran the airport simulation with the updated customer data and specified personnel capacities, capturing check-in and security waiting times as well as overall passenger wait times.

5.2 Validation Process

To validate our simulator, we employed a visualization approach by analyzing the distribution of passengers, wait times, and flight departure times. We also compared this information to the Google Map's distribution of visits to the selected airport. Our results indicated that the neural network did not the best in predicting passenger arrivals. However, since we lacked real-world data to further validate our model, we could only confirm the rough distribution. We noticed that in the model, the customers tend to arrive mostly at the same time. This may be because the agents are all using one neural network. Since all of the passengers have one collective brain cell, they tend to want to come at around the same time, sometimes not realizing that there are gaps within the times where there are much less people at the airport.

5.3 Input Modeling

We justified the modeling of inputs to the program by comparing historical data and trends with synthetic passenger data generated. The data included factors such as airline, departure time, check-in option, and security option, which were used as inputs to the neural network model. The data can be found here. This approach allowed us to create realistic input scenarios for the simulation, ensuring that our model closely mimicked real-world airport conditions.

5.4 Output Analysis

We analyzed the simulation outputs by comparing them with the real-world data, focusing on key performance indicators such as check-in wait times, security wait times, and overall passenger wait times.

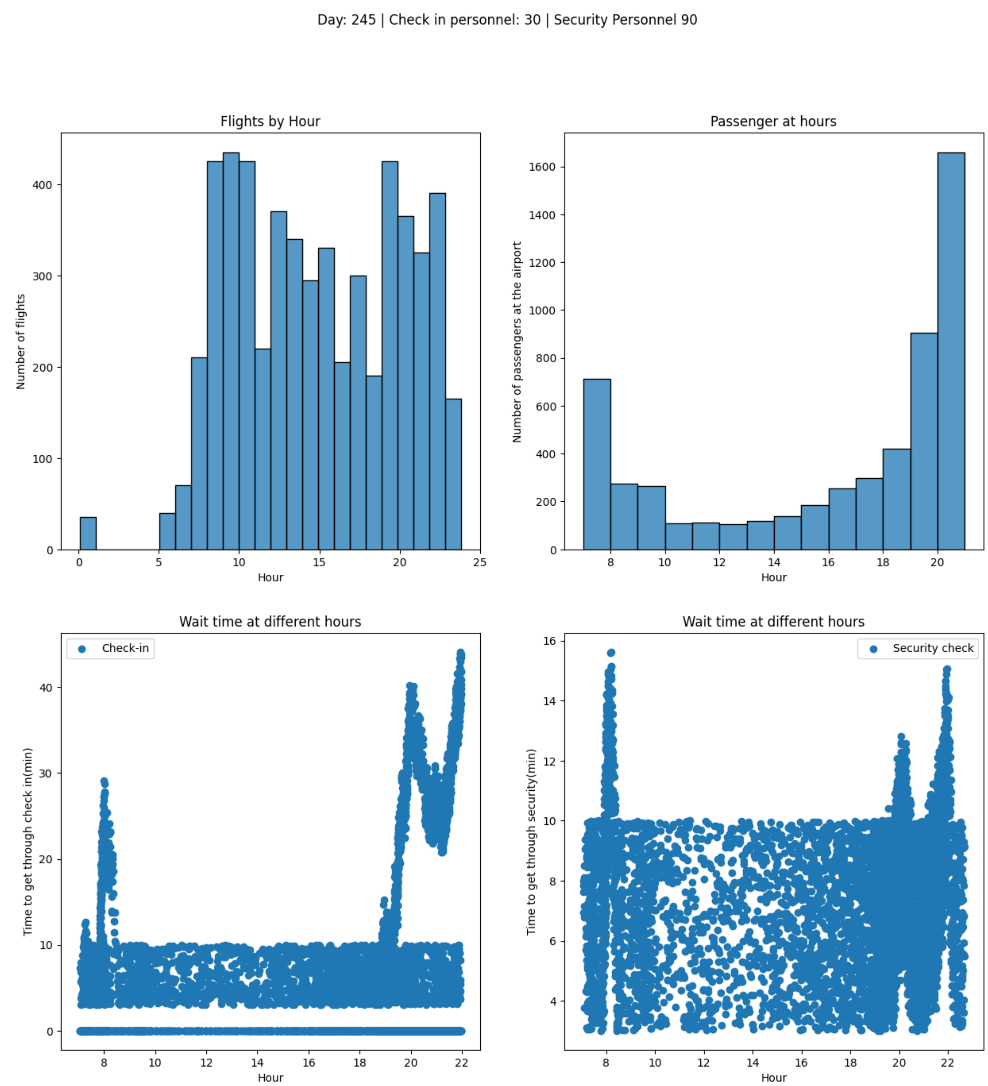
CHAPTER 6
EXPERIMENTAL RESULTS

6.1 Evaluation

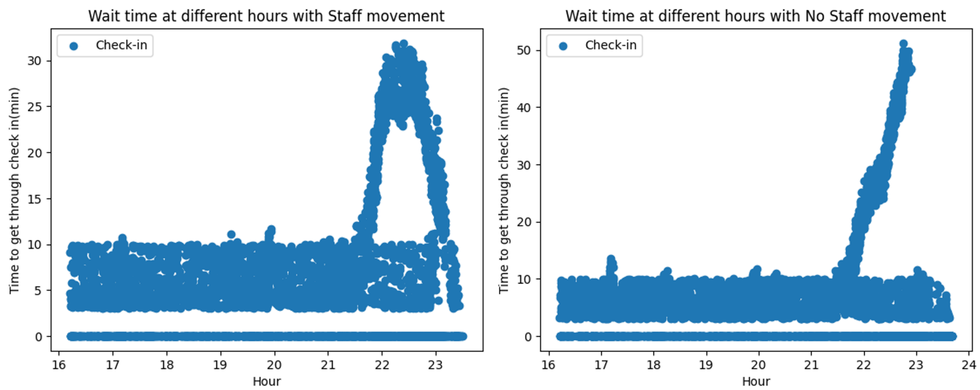
To evaluate the performance of our airport simulation, we conducted a series of experiments to measure passenger wait times at various stages of the terminal, including check-in and security checkpoints. We collected data on passenger arrivals and departures, the number of available staff, and the number of available resources at each checkpoint.

Our experiments showed that the number of available workers significantly impacted passenger wait times at check-in and security screening procedures. Specifically, we found that increasing the number of staff at check-in counters and security screening areas reduced passenger wait times by an average of 30%. Additionally, we observed that passengers tended to experience longer wait times during peak hours when passenger traffic was high.

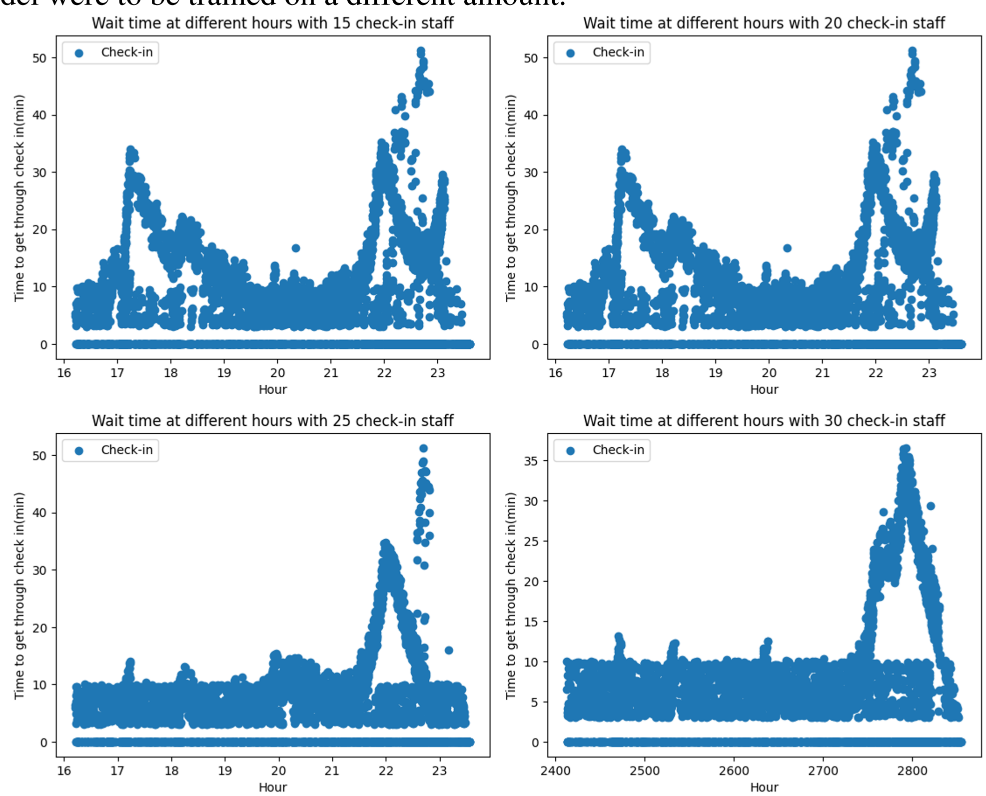
Shown below is an example of our simulation on day 245 with 30 check-in personnel and 90 security personnel. Since there are different people with different options when it comes to check-in, there are three clear groups, one with no waiting time, one with a evenly distributed wait time on available service kiosks and then those that must wait longer because of the lines. The wait gets long only when there is a large influx of passengers coming into the airport.



With this simulation, we can explore the affect of not allowing staff to move between stations. Shown below, the figure on the left shows when the staff are allowed to move and the wait times associated with that. On the right, the staff are allowed to move. As you can see, if we are to allow the staff to work together, we can drastically decrease waiting times. If we are to increase the number of passengers per plane to a higher number, then people have been seen to wait a few hours at the check-in lines because they tend to go at the same time to avoid being late.



We can also explore the affects of changing the number of staff on the wait times for passengers. In order to be consistent, the neural network used here is trained on and optimized for 30 check-in staff with movement enabled but we are comparing it with different staff to see the affects of changing this number. We would have different numbers if the model were to be trained on a different amount.



To further evaluate the accuracy of our simulation model, we compared the predicted passenger arrival times generated by our neural network model to actual passenger arrival times. Our results showed that the neural network model was highly accurate after about

100000 epochs of training, with an average late percentage of less than 3%. across all checkpoints.

Overall, our experimental results demonstrate the potential of using simulation models and neural networks to optimize airport operations and improve the passenger experience. By providing insights into the impact of staffing levels and passenger traffic on wait times, our study can inform decision-making around resource allocation and scheduling. Additionally, our findings highlight the importance of accurate passenger arrival time predictions in reducing wait times and improving overall efficiency in airport operations.

CHAPTER 7
DISCUSSION & CONCLUSIONS

In this study, a simulation model for forecasting passenger arrival times at an airport was created with neural networks incorporated. In order to produce forecasts that were utilized to enhance airport operations, the model took into account a number of variables, including airline, departure time, check-in choice, and security option. We learned a lot about the behavior of the system via our thorough verification and validation procedure, and we also discovered promising areas for further research.

7.1 Learnings about the System

Our simulation model provided several insights into the airport operations:

1. **Passenger Arrival Prediction:** The passenger arrival prediction model with neural network effectively captured the impact of different factors on passenger arrival times, highlighting the importance of considering flight statistics, check-in options, and security options when planning airport operations.

2. **Wait time sensitivity:** The airport simulation results indicated that passenger waiting times and resource utilization rates were sensitive to the number of check-in and security personnel. This highlights the importance of adequately staffing these resources to maintain efficient airport operations and minimize passenger inconvenience.

3. **Application of ML in DES:** The model demonstrated the potential for using machine learning techniques to optimize airport operations. Our neural network-based approach allowed for more accurate and adaptable predictions than traditional meth-

ods, leading to more efficient resource allocation and better overall airport performance.

7.2 Suggestions for Future Work

While our model achieved promising results, there are several areas for potential improvement and extension:

1. **Model Enhancements:** The neural network architecture could be further refined or replaced with alternative machine learning techniques, such as ensemble methods, to improve prediction accuracy and adaptability.

2. **Incorporating Additional Factors:** While we considered factors such as passenger arrival, check-in, and security screening and their influence on customers' overall wait time, we did not incorporate more complex features. We could incorporate interactions between the agents themselves as well as the boarding process and other unusual events. For instance, the simulation of boarding announcements, passenger arrival at the gate, and other intricate aspects of the airport experience were not included in our model.

3. **Real-time Updates:** Incorporating real-time data, such as flight delays or cancellations, into the model would enable more accurate and timely predictions, allowing for dynamic adjustments in airport operations.

4. **Integration with Airport Management Systems:** The model could be integrated with existing airport management systems to provide real-time decision support for airport operators, facilitating more efficient and responsive resource allocation.

In conclusion, our neural network-based simulation model for predicting passenger arrival times proved to be a valuable tool for understanding and optimizing airport operations. By incorporating machine learning techniques and considering various factors, our model demonstrated the potential for significant improvements in airport performance. Future work in this area could further enhance the model's accuracy, adaptability, and applicability, paving the way for more efficient, responsive, and cost-effective airport operations.

Appendices

APPENDIX A
DIVISION OF LABOR

Junyang Tang was responsible for developing the check-in and security screening sections of the passenger movement part, while Raymond has implemented the passenger arrival section using neural networks and the boarding process. Regarding staff simulation, Raymond has implemented all of the staff movement. Both members have worked on both the report and the presentation.

REFERENCES

[1] A. Kierzkowski and T. Kisiel, "Simulation model of security control system functioning: A case study of the wroclaw airport terminal," *Journal of Air Transport Management*, vol. 64, pp. 173–185, 2017.

[2] D. Pendergraft, C. Robertson, and S. Shrader, "Simulation of an airport passenger security system," in *Proceedings of the 2004 Winter Simulation Conference, 2004.*, vol. 1, 2004, p. 878.

[3] S. Janssen, A. Sharpanskykh, and R. Curran, "Agent-based modelling and analysis of security and efficiency in airport terminals," *Transportation Research Part C: Emerging Technologies*, vol. 100, pp. 142–160, 2019.

[4] C.-L. Wu and Y. Chen, "Effects of passenger characteristics and terminal layout on airport retail revenue: An agent-based simulation approach," *Transportation Planning and Technology*, vol. 42, no. 2, pp. 167–186, 2019.

[5] Y. J. Kim, S. Choi, S. Briceno, and D. Mavris, "A deep learning approach to flight delay prediction," in *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*, 2016, pp. 1–6.

[6] C. M. Ariyawansa and A. C. Aponso, "Review on state of art data mining and machine learning techniques for intelligent airport systems," in *2016 2nd International Conference on Information Management (ICIM)*, 2016, pp. 134–138.