Computational Intelligence Methods for Optimising Airport Security Process

A Thesis Submitted for the Degree of
Doctor of Philosophy

By

Mohamad Naji

in

School of Computer Sciences
UNIVERSITY OF TECHNOLOGY SYDNEY
AUSTRALIA

13/02/2022

CERTIFICATE OF ORIGINAL AUTHORSHIP

I, Mohamad Naji declare that this thesis, is submitted in fulfilment of the requirements for the award of Doctor of Philosophy, in the School of Computer Science/Faculty of engineering IT at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Australian Government Research Training Program.

Signature:

Date: 13/02/2022

UNIVERSITY OF TECHNOLOGY, SYDNEY
SCHOOL OF COMPUTER SCIENCES

The undersigned hereby certify that they have read this thesis entitled "**Computational Intelligence Methods for Optimising Airport Security Process** " by **Mohamad Naji** and that in their opinions it is fully adequate, in scope and in quality, as a thesis for the degree of **Doctor of Philosophy**.

Dated: 13/02/2022

Research Supervisors: ____________________
Dr. Madhu Goyal

____
Dr. Ahmed Al-Ani, Dr. Ali Anaissi

Acknowledgements

I am greatly indebted to my supervisor, Madhu Goyal for her continuous encouragement, advice, help and invaluable suggestions. She is such a nice, generous, helpful and kindhearted person. I feel really happy, comfortable and unconstrained with her during my PhD study. I owe my research achievements to her experienced supervision. Many thanks are also due to my co-supervisor, Ahmed Al-Ani for his valued suggestions and constant support, and for the numerous conversations with him. I gratefully acknowledge the useful discussions with Paul Kennedy, Ali Anaissi, and Ali Braytee. I appreciate the travel support for attending the international conferences which I received from the School of Computer Science. I would like to thank my wife, Maha and my children, for their understanding and assistance. This thesis could not have been completed without the support and encouragement of my parents-in-law, and my sisters.

To the soul of my father and mother who instilled in me the virtues of perseverance and commitment. To my wife Maha, whose sacrificial care for me and our children enabled the hours of research, contemplation, and writing necessary to complete this thesis. To my children Ali, Zahraa, Houssein, Fatima and Mahdi you are my greatest achievement, and you will always be my blessings from God. To my sisters thanks for your support as usual. All together we have managed to survive throughout this challenging journey. I love you all.

Table of Contents

List of Tables

List of Figures

Abstract

Airport security screening processes are essential to ensure the safety of both passengers and the aviation industry. Security at airports has improved noticeably in recent years through the utilisation of state-of-the-art technologies and highly trained security officers. However, maintaining a high level of security can be costly to operate and implement. It also lead to delays for passengers and airlines. Nowadays, research is focused to build efficient and effective systems to reduce the congestion caused by the security screening process while maintaining a high level of safety for passengers and the aviation industry. Two open security challenges motivates this thesis: optimize and design the security process at airport, and build an effective intelligent system to detect anomalies in X-ray images.

This thesis proposes a series of novel using queuing theory and machine learning models to handle the aforementioned challenges. Particularly, this thesis addresses the issues related to passengers congestion at the waiting area and improve the performance of the security detection system to ensure the safety of both passengers and the aviation industry.

There are four contributions in this thesis. Contribution 1 proposes queueing theory method to optimise the security screening process with multi-servers operating in parallel to serve a different number of passengers during different seasons, such as

Christmas, Easter and school holidays, and time of the day, as this strongly influences the number of passengers. Contribution 2 proposes a novel method based on queueing theory augmented with particle swarm optimisation (QT-PSO) to predict passenger waiting time in a security screening context and to determine the required number of servers and security officers. Contribution 3 propose a tensor-based learning approach to extract the informative latent features that will be used as an input to build a one-class model for anomaly detection. Contribution 4 proposes a federated learning (FL) approach for anomaly detection in X-ray security imaging using OCSVM. The innovative machine learning approach can train a centralized model on data generated and located on multiple airports without compromising the privacy and security of the collected data. The performance of all novel methods in this these is evaluated in the context of Sydney airport dataset, synthetic data, and public datasets for X-ray images. Further, all the results of the novel methods are compared to the state-of-the-art methods. The experimental results shows that our proposed methods in the contributions outperform the state-of-the-art and produce promising results.

Publications

Below is the list of journal and conference papers associated with my PhD research:

Naji, M., Anaissi A., Braytee A. & Goyal, M. (2021, April). Anomaly Detection in X-ray Security Imaging: a Tensor-Based Learning Approach. In 2021 International Joint Conference on Neural Networks (IJCNN) (pp. 1-8). IEEE

Anaissi, A., Suleiman, B., & Naji, M. (2021, April). Intelligent Structural Damage Detection: A Federated Learning Approach. In International Symposium on Intelligent Data Analysis (pp. 155-170). Springer, Cham.

Naji, M., Al-Ani, A, Braytee A., Anaissi A, Goyal M. & Kennedy, P. (2020, March). Design of Airport Security Screening Using Queueing Theory Augmented with Particle Swarm Optimisation. Journal of Service Oriented Computing and Applications.

Naji, M., Braytee A., Anaissi A., Sianaki, O. & Al-Ani, A, (2019, March). Optimizing the Waiting Time for Airport Security Screening using Multiple Queues and Servers. In CISIS 2019.

Naji, M., Al-Ani, A, Braytee A., Anaissi A.& Kennedy, P. (2019, January). Queue Formation Augmented with Particle Swarm Optimisation to Improve Waiting Time in Airport Security Screening. In AINA 2019.

Anaissi, A., Lee, Y., & Naji, M. (2018, December). Regularized Tensor Learning with Adaptive One-Class Support Vector Machines. In International Conference on Neural Information Processing (pp. 612-624). Springer, Cham.

Anaissi, A., Braytee, A., & Naji, M. (2018, July). Gaussian Kernel Parameter Optimization in One-Class Support Vector Machines. In 2018 International Joint Conference on Neural Networks (IJCNN) (pp. 1-8). IEEE.

Naji, M., Abdelhalim, S., Al-Ani, A. and Al-Kilidar, H., 2017. Airport Security Screening Process: A review. In CICTP 2017.

Chapter 1

Introduction

The aviation industry has been growing year after year and has become an essential mode for passenger transportation and to connect countries and cities around the world. It also plays a main role in supporting economies development and tourism. The number of passengers has been continuously increasing in almost all airports around the world. For example, the number of passengers that travelled through Sydney airport was 37 million in 2012, and this number is expected to reach 74 million by 2033. In fact, it is expected that the worldwide number of passengers travelling by air will grow between 4.2% and 4.7% by 2033 to reach 6 billion passengers. As for the United States, the number of passengers that travelled by air reached 1 billion in 2015 [65].

This important and complex industry has faced many security challenges that include aeroplane hijacking, and airport and aeroplane attacks [63]. The attacks of $11th$ of September in the U.S.A heightened the importance of aviation security and passenger safety to governments and aviation authorities. A number of legislation's have been issued because of that in many countries, where in the United States the Transportation Security Administration (TSA) was established to enhance the security level at airports [30]. After introducing security processes such as X-rays and explosive detection in airports, the number of attacks decreased but the number of fatalities rose, for example, there were 111 attacks in the 1970's causing 557 fatalities, whereas between 2000 and 2009 the number of attacks decreased to 21 attacks but the number of fatalities increased to 3032, including the 11th of September attack [33]. The increase in the number of passengers, and the threat of attacks on aircraft has made security screening of passengers before boarding essential, which has reduced comfort and passengers' satisfaction. Also, the screening instruments are very costly to acquire and maintain, hence having enough screening systems to avoid congestion has a huge economic impact on airlines, and airports [76].

The design of the security screening area is a crucial and complicated process that has big impact on the safety of the aviation industry and passengers. The security screening process consists of many processes, machines, instruments, and personnel. Therefore, managing the security screening process within acceptable budget and delay constrains is becoming increasingly difficult. This has inspired many researchers to study and optimise the congestion caused by the security screening process and strike a balance between reducing congestion and delays while maintaining a high level of safety for passengers and the aviation industry [76]. The propose techniques incorporate different methodologies, such as queueing theory [34, 69], fuzzy reasoning [76], lean management [4], Bayesian networks [84], simulation tools [20], machine learning [79], tensor analysis [66], and federated learning [38]

As the security process consisting of multi-stages and processes includes passengers arriving security area, joining a queue and scanning bags via X-ray machines.

In this sense, it is not possible to optimise a single process and neglect others. This study address the challenges that arise from security screening processes: (i) optimising the security process to reduce average waiting time, (ii) designing the airport security screening area to determine the number of servers required to serve different number of passengers, (iii) proposing a novel anomaly detection in X-ray security screening systems based on tensor analysis and one-class classification model. (iv) finally, developing a federated learning (FL) approach for anomaly detection in X-ray security imaging.

Different methods are used to optimise the security process, however, to the best of our knowledge, there are no studies that investigate the impact of queue formation, and the size of the security area on the average waiting time of passengers going through the security screening process. Most of the studies for airport security screening process, that used a queueing theory as a method, have implemented approaches based on single server, such as M/M/1 and M/G/1, where M stands for a system with Poisson arrivals, G generally identically distributed, while 1 is the number of servers [23], and then multiply the outcome by the total number of servers to determine the total average waiting time for the whole system. This assumption cannot be practically applied for many reasons, like differences in human experience, such as knowledge of the security screening process and the existence of special need passengers which will require more processing time due to passengers with slow motion and families with a greater number of passengers, as well as differences between machines, like (X-ray, metal detector, etc). For all these reasons, we consider these issues in our model during the implementation phase in order to determine the average waiting time for the whole system with multiple servers. We therefore considered Poisson input, exponential service time, multi-servers and the buffer or area size, which is known as M/M/S/K (where M/M stands for a system with Poisson arrivals and exponentially distributed service time, S number of servers, and K is the buffer size). To the best of our knowledge, this study represents the first attempt to explore the impact of queue formation on optimising the average waiting time for airport security screening process. Furthermore, it is considered as the first attempt to use the multi-parallel servers that are implemented in different scenarios to demonstrate real life settings.

Furthermore, a reliable security screening process system should handle input variability, such as the varying number of passengers at different times of the day, weekdays vs. weekends, and busy seasons (e.g., Christmas and Easter) vs. other times of the year. Some of the other challenges include handling different types of security machines, while the number of servers is dynamically adjusted according to queue length. The security officers must increase the speed of the process and make accurate decisions as to whether passengers or bags are carrying threatening items. Passenger variation should also be taken into consideration, such as individual passengers, frequent flying passengers, families, passengers with special needs and flight crew.

This study proposes a novel method based on queueing theory augmented with particle swarm optimisation (QT-PSO) to accurately predict passenger waiting time in a security screening context. This new method employs Lindley process as a queuing model with a new parameter named $W_{walking}$ to incorporate customers' walking time in our model. This parameter is often representing 30–60% of the service time; thus, it is profoundly affecting the passenger waiting time. However, the value of this parameter is unknown, and it is very hard to tune it manually based on the given data. Therefore, we used particle swarm optimisation (PSO) to optimise the walking time at different distances from the servers. The model also considers three types of lanes, namely, normal, slow, and express. The Lindley process formula is used here as our PSO fitness function.

This thesis also addresses the issue of automatic anomaly detection in X-ray security screen systems. It is well-known that X-ray machines play an important role in scanning passengers' bags to detect prohibited and dangerous items before a threat occurs. The rationale for using an X-ray scanner is to identify the contents of a bag without opening it to determine if a manual hand check is required. Baggage screening takes around 3-15 seconds depending on the type of machine, accuracy of the image, and the experience of the security officer [47].

To strike a balance between increasing security and reducing scanning time, and to address the issues of congestion and cost, different machine learning methodologies have been proposed for anomaly detection in X-ray security screening systems [79, 27, 1]. This study presents a novel method for anomaly detection in X-ray security screening systems based on tensor analysis and a one-class classification model. Our method initially performs data fusion of multi-angle scanned images in one tensor structure from where we extract the informative features, and further constructs a one-class support vector machine model using these features to detect anomalies. We evaluate this approach using two image-based datasets and one real X-ray security baggage data collected from Sydney airport. The results show that our tensor-based learning method outperforms other state-of-the-art approaches.

Finally, to further improve the anomaly detection capability in X-ray security imaging, we develop a federated learning (FL) approach using one-class support vector machine (OCSVM). FL allows the central machine learning model to build its learning from a broad range of data sets located at different locations. It aims to train a shared centralized anomaly detection model using datasets stored and distributed across multiple clients/airports. This innovative machine learning approach can train a centralized model on data generated and located on multiple airports without compromising the privacy and security of the collected data. Also, it does not require transmitting large amount of data which can be a major performance challenge especially for real-time applications. The FL approach can enable multiple airports to collaborate on the development of a central anomaly detection model by only sharing the model coefficients of each client/airport model rather than the whole data collected by all participating airports.

1.1 Motivations

This section discusses the impact of the security process on passengers and airports. Delays in the security process can reduce the satisfaction and comfort levels of passengers. Long processing times may cause flight delays and impact flight scheduling, i.e., affect the operation of airlines and airports.

1.1.1 Passenger Satisfaction

Security processes at the airport have undeniable costs for passengers that can make many passengers unsatisfied, and can be divided into four types: time delay, indirect financial costs, privacy, and inconvenience. Due to the time cost, passengers must arrive at the airport three hours before the departure of the plane, and sometimes passengers miss their flights due to the long delays at the screening process [5, 35].

Due to different reasons, some passengers think that the airport security process is useless and a waste of their time, leading to their dissatisfaction [65]. However, the level of satisfaction of passengers waiting in the queue can change if the passengers know the reason for any delay. Passenger satisfaction can vary based on age, gender, educational history, and previous knowledge of the dynamic of the process [35].

Due to their expenditure on security technology, most government and airport administrations have imposed a new tax, known as service cost or passenger screening cost, which varies from country to country and tends to increase annually. For example, in the U.S.A., the government first imposed this tax in 2001 and the cost per passenger one way was $2.50, however, this tax kept increasing and became $5.60 per passenger one way in 2014. Also, in 2014, a proposal to increase this tax 50 cents yearly until 2019 [33].

1.1.2 Economic Impact on Airlines and Airports

After explaining why airport security is absolutely indispensable, this section discusses the economic costs of the airport security processes. Long delays caused by security processes could cause delay to some flights, which would not only waist fuel but could also distribute departures and arrivals of other flights [52].

The cost of the security equipment, training security officers and their wages represent an additional expense deducted from the revenue of governments in general, and specifically the airport administration [44]. For example, the price of an explosive detector system (EDS) is $1 million and it will cost $1.5 million to $2 million to install and re-program it, according to [53]. It is estimated that the expenditure on aviation security of 18 European cities in 2011 reached 5.7 billion euros. In order to improve security to the United States airports, the government increased its funding to the transportation security administration to $8 billion in 2013 [33].

Due to the excessive tax and the impact of security processes on passengers, the number of air travellers decreased in some countries, for example, in Canada, the number of passengers dropped by 690,000 resulting from air transport security charges, in 2011 which translates to a loss of $227 million for the aviation industry and overall national economic loss of $2.2 billion [33].

1.2 Research Objectives and Aims

The objectives of this thesis are to optimize and design the security process at airport, and build an effective intelligent system to detect anomalies in X-ray images. In order to achieve these objectives, this thesis has the following aims:

1. Optimising and designing security screening area.

    - To develop a queues queueing theory (QQT) model for optimising the security screening process

    - To develop a queueing theory augmented with particle swarm optimisation (QT-PSO) model to: (i) designing the security screening process; (ii) forecasting the average waiting time based on number of servers and passengers;(iii) determining the number of servers required based on the number of passengers; (iV) model human and machine variations.

2. X-ray security screening images

    - To develop an anomaly detection model in multi-view learning setting.

    - To implement a data fusion technique for multi-view images

    - To develop central model using a federated learning network approach

1.3 Research Questions

Based on the above research aims, the research questions of this thesis are specified as follows:

- **RQ1**: How to optimise the security screening process to reduce passenger's average waiting time?

- **RQ2**: How to design the security screening process and forecast the average waiting time based on number of passengers and servers?

- **RQ3**: How to build an anomaly detection model in multi-view learning settings of X-ray security screening system?
- **RQ4**: How to implement a central model for Anomaly detection in X-ray security screening?

1.4 Contributions to Knowledge

This section presents the following contributions to knowledge based on the four research questions:

1. Optimizing the Waiting Time for Airport Security Screening using Multiple Queues and Servers-Framework.

   We build a framework consists of queueing theory and Lindley process for QQT (Queues Queueing Theory) model to optimise the security screening process with multi-servers in parallel to serve different number of passengers during different seasons.

2. Design of airport security screening using queueing theory augmented with particle swarm optimisation.

   We propose a novel model based on queueing theory augmented with particle swarm optimisation (QT-PSO) to predict passenger waiting times in a security screening context.

3. Anomaly Detection in X-ray Security Imaging.

We construct a tensor-based learning anomaly detection model in X-ray security imaging. Our method initially performs data fusion of multi-angle scanned images in one tensor data structure from where we extract the informative features, and further constructs a OCSVM model using these features to detect anomalies. Our tensor-based learning method also includes a novel algorithm called Edged Support Vector (ESV) for optimizing the Gaussian kernel parameter inherent in OCSVM and a regularised alternating least square (RALS) method for tensor decomposition.

4. A Federated Learning Anomaly Detection Approach for X-ray Security Imaging.

   We develop a federated learning anomaly detection model with a novel method of learning OCVSM model in FL settings and an efficient communication method for coefficient's aggregation.

1.5 Thesis Structure

This Thesis is presented as follows:

Chapter 2 presents background concepts and previous works related to the research topics of airport security screening area optimisation, anomaly detection in X-ray security image, and federated learning for X-ray security imaging.

Chapter 3 presents a relevant background of queueing theory, Lindley Process, particle swarm optimisation ,regression analysis, One-Class Support Vector Machine,Tensor Analysis.

Chapter 4 proposes QQT to optimise the waiting time for airport security screening using multiple queues and servers.

Chapter 5 presents QT-PSO model to design of airport security screening using queueing theory augmented with particle swarm optimisation.The proposed methods are evaluated and compared to the state-of-the-art and conclude with contribution to knowledge.

Chapter 6 presents a novel tensor based learning method for anomaly detection in X-ray security screening system based on tensor analysis and one-class classification model.The proposed methods are evaluated and compared to the state-of-the-art and conclude with contribution to knowledge.

Chapter 7 presents a federated learning (FL) approach for anomaly detection on X-ray security imaging using OCSVM. The proposed methods are evaluated and compared to the state-of-the-art and conclude with contribution to knowledge.

Chapter 8 concludes the thesis, putting the work into a broader context. It discusses the strengths and weaknesses of the proposed contributions, and describes future directions of research.

Chapter 2

Literature Review

This chapter provides the necessary background information for this thesis. Section 2.1 presents the background of security process. Section 2.2 explain the security procedure. Section 2.3 gives an overview of the time requires for the security process 2.4 presents the instruments used for security screening process . Section 2.5 Discusses the importance of security officers. Section 2.6 Presents the previous methods for optimising security processes at airports. Section 2.7 Discusses the recent methods proposed for anomaly detection in X-ray security imaging. Section 2.8 Discusses the concept of federated learning approach and presents the recent published methods in this area.

2.1 Background of Security Process

Technology and security operators play a very important role in improving the security of airports and the aviation industry. The aim of these scientific-technological instruments, such as surveillance cameras, X-ray machines, millimetre wave gates and explosive detectors is to scan passengers and their bags to detect any threats and avoid attacks, and hence maintain a safe and secure environment and flights [5],and to ensure that terrorists are not able to use aviation as a weapon to kill civilians and destroy buildings and assets in cities. Paul Benda, the former Director of the US Homeland Security Advanced Research Projects Agency in the US Department of Homeland Security, says technology has the power to improve the level of security as well as efficiency and to also provide a better air travel experience for passengers [76, 33].

The following subsections, explain the security procedure, the time required by passengers to go through the screening process, the role and functionality of surveillance cameras, X-ray machines, and millimetre wave and metal detectors. The role of security officers will also be discussed.

2.2 Security Procedure

The procedure of passenger check-in and moving through the security screening process differs from country to country, based on the governments' laws and airport administration. When the passengers have checked in, they then go through the security process, where they are only allowed to carry a small bag, known as a carry-on bag or hand luggage. This bag should not exceed a certain weight, as decided by the airline regulations and contain less than 100 ml of liquid.

The security area consists of several lanes of X-ray scanners, metal detectors, and millimetre wave detectors. In most airports, the passengers' hand luggage is first scanned through the X-ray, while electronic devices such as computers and mobile phones must to be scanned separately for better visualisation. If the security officer suspects the bag contains a prohibited item, then a manual inspection is carried out. After this, the passengers walk through a metal detector and millimetre wave gate for body image processing to ensure no prohibited or illegal items are being carried; a pat-down search will be carried out by the security officer if there is an alarm, otherwise the passenger collects their bag from the X-ray conveyor belt. Some passengers are selected randomly for additional screening while the others continue to the boarding gate [5, 72, 15]

2.3 Time Required for the Security Process

The time required for a single passenger to complete the security process varies between 15 and 60 seconds and takes 25 seconds on average. The ideal time for this process varies from 20 to 30 seconds for informed passengers, and between 60 to 120 seconds for inexperienced or uncooperative passengers. Also, an additional 20-40 % of time is required for passengers who have forgotten to remove prohibited items from their bags either intentionally or non-intentionally, 85-90 % of these items being harmless liquids, but they exceeded the 100 ml limit [44]

2.4 Security Instruments

The section below illustrates the instrument and machine has being used for the airport security purpose.

2.4.1 Surveillance Cameras

Surveillance vision systems are currently very popular and are being installed in both public and private places, such as streets, highways, hospitals, schools, universities, airports, and in many private companies to increase safety. Surveillance security is used in airports to quickly identify possible threats such as unattended bags and to also search for the person who dropped it. They also have the ability to locate a suspicious person in a crowd, and they also improve the throughput of the airport by monitoring travellers' movements. In 2012, a total of 8 million surveillance cameras were connected through the Internet world-wide and this number is expected to increase to 170 million by 2021 [44, 60]. Some of the advanced surveillance cameras are supported by internet protocols and provide digital high definition (HD) images to improve the image quality and processing [83].

2.4.2 X-Ray

Due to the increased terrorist attacks on aviation, governments introduced X-ray machines in airports to scan passengers' bags to increase safety and security by detecting prohibited and dangerous items before a threat occurs. The rationale for using an X-ray scanner is to identify the contents of a bag without opening it to determine if a manual hand check is required. Baggage screening takes around 3-10 seconds depending on the type of machine and the experience of the security officer [76, 47]. Different types of X-Rays have been developed, 2-D, 3-D, black and white and colour. This technology uses a pseudo colour technique to distinguish items of different colours. Several countries, such as some European countries, have installed the most advanced type of scanner known as CT machines which display the bag in 3-D and have the ability of 360-degree rotation [5]. According to a recent article published in the London Telegraph newspaper, passengers do not have to remove their electronic items and bottles of liquid from their hand luggage before screening by the CAT scanners [60].

2.4.3 Body Image Processing

Body Image Processing Body image scanners are used to scan the human body of a passenger and transform it into an image to reveal if the passenger conceals a prohibited or illegal item, such as explosive, knife or other metals.

There are two types of body image processing: ionizing radiation such as X-ray systems (active systems) and non-ionizing radiation systems such as millimetre wave and terahertz (passive systems). Active systems screen passengers by emitting radiation whereas passive systems receive radiation from passengers in order to scan their bodies. Some European countries use the non-ionizing scanners due to health and privacy issues, especially in relation to pregnant women and babies, while the U.S uses ionizing scanners. In relation to the safety of ionizing scanners, it has been proven that the amount of radiation that the passenger receives is less than 1 % of the radiation a person will be exposed to during a in high altitude flight [5].The advantage of ionizing scanners is that they are more accurate than non-ionizing scanners; however the disadvantage is the concern over exposure to radiation, even though it is a very low level. These instruments, such as X-ray, CT scanners, millimetre wave and metal detectors can sometimes give false responses, therefore it is important that security officers are sufficiently experienced to accurately interpret these images and operate the scanners.

2.5 Security Officers

The technological instruments are operated and controlled by security officers who are hired by the government or by the airport administration, depending on government policies. For example, Dubai airport employs locals to undertake security screening

[5], while in Europe and Australia, security officers are hired through private companies [33].The safety of the passengers and the aviation industry depends on the experience and skill of the security screening operators or ‘screeners’ [76] to detect potential threats.

Screeners should possess certain skills and knowledge. Firstly, screeners should have a full understanding of which items are prohibited, such as knives, guns, explosives, gas, liquids and gels [48]. In the USA, 1.6 million knives and 11.6 million lighters were intercepted by transportation Security officers in 2006 as prohibited items [54]. Secondly, screeners should be able to interpret the images from the scanners and distinguish between different items in a short period of time. Thirdly, they must be able to work quickly and avoid errors [76, 45, 54]. Fourthly, security officers should assume all passengers are equally likely to pose a threat [44]. Finally, it is important that the percentage of false alarms and false clears [14] are reduced and that the screening process time is kept to a minimum. Most of the screeners’ decisions are considered from a group context, but sometimes they are based on information from intelligence agencies and the police department [45, 59].

2.6 Optimising Security Processes at Airports

Due to the growth in the number of passengers around the world, and the delay caused by screening every passenger, the security process has become very slow and causes huge delays. However, if a decision was made to only select certain passengers to scan, this may result in a threat to the security of passengers, aviation, and airports [52]. Another option is to install extra equipment and hire more employees to deal with seasonal and peak hour air traffic, but this also has a limitation due to the size of certain airports and security areas [54]. To scan all passengers in the U.S.A. by EDS only would require the installation of over 6,000 EDS, which will have a total cost around $12 billion. The number of EDS would vary, based on the airport size and the growth in the number of passengers. The safest, cheapest and most applicable option is to optimise the security process by developing a scientific method to reduce processing time and increase the throughput [53].

Seasonal travel times, such as Easter and the Christmas holidays, the winter season, and also the growth in the number of passengers motivates the researchers to apply scientific knowledge, such as queueing theory, lean management, linear programming, fuzzy reasoning and some stochastic, statistical and different approaches to provide new models to optimise the security process and minimize the time passengers spend in queues.

In the following subsections, the background knowledge on fuzzy reasoning, Bayesian networks, lean management. and queueing theory will be explained, and their general application in airport check-in will be discussed, followed by an overview of how these methods are being used to optimise the security screening process.

Other authors have used fuzzy reasoning and linguistic variables to analyse human and technical factors to enhance the baggage screening process. For instance, [76] use fuzzy reasoning to improve the efficiency of a baggage screening system with respect to the uncertainty of machines (X-Ray) and human factors. Later, the authors developed a Fuzzy Passenger Security Control Assessment (FUPSCA) by using fuzzy set theory and fuzzy inference to assist security managers in deciding how to organise the security screening process [77]. Other work proposed by [41] uses Fuzzy Quality Function Development (QFD) which is based on multi-objective linear programming to improve airport service quality. The authors in [42] propose a prediction model based on linear regression to predict number of passengers arriving to the immigration and security areas and individual passengers connection time. The authors claimed that model produce very accurate results, however they did not apply it to predict the passenger's average waiting time in the security area. Similarly, Felkel et. al [29] also use a linear regression model to predict the passenger flow management at Frankfurt airport. On the other hand, Gaus et. al [32] propose a convolutional neural network for X-ray security imagery detection and classification of prohibited items. [44] discusses the cost of passengers behaviour on the security screening process, and others use the simulation method for optimising the security screening process [71]. These authors propose a model based on two concepts (AS-IS and To-Be) to determine the dynamic balance between staff and passenger requirements, such as passenger flow and also to determine the alternate future design. This work has been implemented at the Baltimore-Washington International Airport. [31] propose a convolutional neural network for X-ray security imagery detection and classification of prohibited items.

2.6.1 Queueing Theory Methods for Optimising Airport Security Screening Area

Several types of security machines are used in airports in the security screening area, such as several lanes of X-ray, computed tomography (CT) machines, metal detectors and millimetre wave gates to ensure that no prohibited or illegal items are being carried. The procedure of the security screening process differs from city to city and between countries, and this process process could be based on governmental laws or could be determined by airport administration. For example, at Sydney International

Airport, passengers go through the security screening process after they have completed the immigration procedure. It is permissible for passengers to take a small bag, known as hand luggage or carry-on, on to the aircraft. The permissible weight and size of this bag are determined by the airline, and there is a limit on the volume of liquids which can be carried on board. Bags are first checked by X-ray or CT machines, and then, passengers are checked by metal detectors or millimetre wave gate. A pat-down search is carried out by security officers if it is required or if there is an alarm [15].

One of the earliest studies to use a general queuing theory to optimise the screening process for a single server traffic uses intensity as a ratio of arrival rate to service rate [34]. The researcher stated that, in order to keep the system stable, the ratio of the intensity rate has to be equal to or less than 1, i.e., the service rate has to be greater than or equal to the arrival rate. The work in [52] proposes optimal static and dynamic policies, and the system is composed of a multilevel security system to optimise the security process.

The aim of the optimal static assignment policy is to reduce the screening process time, while the optimal dynamic policy is used to balance the time required for a passenger to finish the security process and reduce the false alarm rate. Due to the small amount of time required to scan the passengers and their bags, the service time cannot follow an exponential distribution as the author proposed.The work in [69, 54] propose similar models which follow the same concept of the multi-level single server M/M/1 model and the FIFO discipline, where a new parameter β known as rejection rate is introduced, and in [14], if $\beta = 1$, the bag will be rejected and will be scanned again. The total time required for the bags to be scanned and exit is

$T = T_p + T_s$ [54], where T_p and T_s are the primary and secondary inspection times and $T_p = \lambda_1/[\mu_1(\mu_1 - \lambda_1)]$, $T_s = \lambda_2/[\mu_2(\mu_2 - \lambda_2)]$.

The method developed by [69] is known as security speed and accuracy operating characteristic (SAOC), and is based on the same concept. If the bag is rejected in the first stage, it will be redirected to the second stage scanning system, where λ_1, λ_2, μ_1, and μ_2 are the arrival and service rate at stage 1 and stage 2, respectively. This approach has been applied in the busiest airport in the U.S. (Atlanta Hartsfield), where the number of passengers travelling through this airport yearly is around 90 million, which is 22 million passengers more than Heathrow airport.

The work in [68] uses a stochastic process to design, build and operate a security process, which is composed of two stages: the first stage is to determine the number of security devices to be installed; the second stage is to use the stochastic process to optimise the security process by screening the passengers through the available security devices. This screening is subject to assignment constraint of the passengers.

The work in [58] applies Parkinson's Law to minimize the service time as the queue length increases, i.e. speed up when the queue length is long, and slow down when the queue length is small. It is one of the proposed methods to reduce processing time without affecting the security. In [67], the author proposed a model to enhance the security process by assigning passengers who may represent a risk or threat to a selected lane, and also to improve the passenger checkpoint screening system by achieving a higher probability of a true alarm. The author used the basic concept of stochastic process and the theorem of total and conditional probability to propose his model.

Most of the reported work in the literature only use the traditional queueing theory to optimise the security screening process following the M/M/1 and M/G/1 approaches [23] where M represents a system with Poisson arrivals and exponentially distributed service time; G is the general identical distribution, which is independent of the arrival process and must be a non-negative random variable; and finally 1 is the number of servers. These approaches simply multiply the outcome of M/G/1 or M/M/1 by the total number of servers to determine the total passengers' average waiting time. However, this assumption cannot be practically applied and does not produce accurate results for various reasons, such as differences in staff and passenger experience, knowledge of the security screening process and the existence of special needs passengers who require more processing time. Furthermore, walking time is another important factor that needs to be considered since it has a great influence on the passengers' waiting time. For example, passengers with special needs and families require longer walking time, thus more processing time. Finally, the variation among different screening machines, such as X-ray machines and metal detectors also affects the total processing time [65].

For the aforementioned reasons, we propose a model that attempts to incorporate all these aspects in the implementation phase. So, an M/M/S queueing system with Poisson input, exponential service time, and multiple servers is used to determine the average waiting time for a system with multiple servers. As the model requires the average walking and processing times, we decided to optimize the walking time parameters using particle swarm optimization. However, the service time is tuned based on previous historical data and previous experience. To the best of our knowledge, this study is the first attempt to use multiple parallel servers that are implemented in different scenarios to replicate airport security screening real-life settings.

2.7 Anomaly Detection in X-ray Security

Different machine learning methods have been reported in the literature for anomaly detection and classification of prohibited items automatically for intelligent X-ray baggage security screening. Some researchers have used anomaly detection while others have used supervised learning techniques.

For instance, the authors in [79] state that instead of focusing on abnormal objects, human operators appear to do a better detection when focused on benign objects. In addition, experience of every day normal object inspection contributes to a much better result in detection. The same approach is used in anomaly detection principle, where the model is only trained with normal samples and evaluated on normal/abnormal examples. Moreover, according to [74] in the absence of label, unsupervised anomaly detection can be used to uncover rules that are able to separate normal and abnormal data. The most popular approach is to use One-Class SVM (OCSVM) to detect anomalies, which builds a smooth boundary around most of the probability mass of data.

The author in [1] propose a model composed of generative adversarial networks (GANs) and encoder-decoder encoder subnetworks which is known as (GANomaly). The aim of the model is to minimise the distance between real, generator images and their latent representation. While the author in [2] propose a Skip-GANomaly by using an encoder-decoder convolution neural network (CNN) with skip connection. The enhancement added to [1] is in the generator network to cope with higher resolution image. The work in [36] use a CNN to extract the features of the X-ray images for parcels, then trains a multivariate Gaussian model to capture the normal distribution of center for ‘Applied Science and Technology’ dataset. The work developed by [31]

introduces a dual automatic convolutional neural network design to detect the abnormal inside complex surveillance X-ray image. The authors use recent advantage in region-based (R-CNN) mask-based advances CNN (Mask R-CNN) and architectures of identification such as RetinaNet to provide variants of object localization for unique objects Interest classes.
While the author in [10] use CNN and decision-tree learning to develop a model to detect anomalies items in X-ray images inside cargo containers. Menon and Chawla [21] propose a one-class neural network (OC-NN) model to detect abnormal features or data in complex data sets. The work presented by An and Cho [6] use the reconstruction probability from the variational autoencoder, which is combination of variational inference with deep learning to propose an anomaly detection method known as (VAE). The author in [9] use a CNN based on decision-tree learning to propose an anomaly detection algorithm to detect a threat in X-ray cargo image. The work in [62] propose an end-to-end trainable consist of Convolutional Long Short-Term Memory (Conv-LSTM) networks which is known as (AnoGAN). The model is able to predict the evolution of video sequence from a limited number of input frames. The author in [86] develop an EGBAD model which is based on GAN at the same time to learn the encoder during training which is used for image anomaly detection.

The authors have used supervised machine learning such as a class-balanced hierarchical refinement (CHR) which use deep learning to model X-ray image detection. Authors use a dataset consist of 1 million X-ray images where only less than 1% have prohibited items proposed by [64]. The work in [85] proposed a model consist of GAN and CNN to enhance the GAN training to produce better X-ray images. The studies in [75] used a deep convolution neural network (CNN) to increase the depth by using an architecture with small ($3x3$) convolution filters to provide a ConvNet model. The author in [82] uses the Speed-Up Robust Features (SURF) with Support Vector Machine (SVM) to model Bag-of-Words (BoW) in order to detect a concealed firearms in baggage security Xray image. The work in [3] uses transfer learning by train SVM classification on CNN features to posed for object classification and detection within X-ray security baggage. In [73] use CNN to detect Small Metallic Threats (SMTs) hidden amongst legitimate items inside a cargo container by using dual-energy X-ray images for automated threat detection. The author in [80] combine inception architecture and residual connection which can accelerate the training of inception network significantly. This study has been applied for classification purpose.

2.8 Federated Learning

The concept of Federated Learning (FL) was initially proposed by Google for improving security and preventing data leakages in distributed environments [49]. FL allows the central machine learning model to build its learning from a broad range of data sets located at different locations. It aims to train a shared centralized machine learning model using datasets stored and distributed across multiple devices or sensors. FL has gained a lot of interest in recent years and as a result, it has attracted AI researchers as a new and promising machine learning approaches [40, 81].

This FL approach attracts several well-suited practical problems and application areas due to its intrinsic settings where data needs to be decentralized and privacy to be preserved. However, only a few studies, that have been reported in the literature, utilized the FL approach to construct a global model. For instance, Bonawitz *et* *al.*[39] employed FL model settings to develop a system that solves the problem of next-word prediction in mobile devices. On the other hand, other studies focused on addressing the training challenges of a central model to support all local data training especially when the distribution of data across clients is highly non-IID (independent and identically distributed).

McMahan *et al.*[61] proposed the first FL-based algorithm named *FedAvg*. It uses the local Stochastic Gradient Descent (SGD) updates to build a global model by taking average model coefficients from a subset of clients with non-IID data. This algorithm is controlled by three key parameters: C, the proportion of clients that are selected to perform computation on each round; E, the number of training passes each client makes over its local dataset on each round; and B, the local mini-batch size used for the client updates. Selected clients perform SGD locally for E epochs with mini-batch size B. Any clients which, at the start of the update round, have not completed E epochs (stragglers), will simply not be considered during aggregation. Subsequently, Li *et al.*[55] introduced the *FedProx* algorithm, which is similar to FedAvg. However, FedProx makes two simple yet critical modifications that demonstrated performance improvements. FedProx would still consider stragglers (clients which have not completed E epochs at aggregation time) and it adds a *proximal term* to the objective function to address the issue of statistical heterogeneity. Similarly, Manoj *et al.*[11] addressed the effects of statistical heterogeneity problem using a *personalization-based approach (FedPer)*. In their approach, a model is viewed as base besides penalization layers. The base layers will be aggregated as in the standard FL approach with any aggregation function, whereas the personalized layers will not be aggregated. Several other methods have been proposed to achieve personalization in FL. Recently, Smith *et al.*[78] proposed a new algorithm named *MOCHA-based multi-task learning (MTL)* framework to address the non-IID challenge in FL. Hanzel *et al.*[38] also proposed an L2GD algorithm that combines the optimization of the local and global models. Similarly, Deng *et al.*[26] developed an *adaptive personalized federated learning (APFL)* algorithm which mixes the user's local model with the global model.

Chapter 3

Preliminaries

In the following chapter, we present a relevant background of queueing theory, Lindley Process, particle swarm optimisation ,regression analysis, One-Class Support Vector Machine,Tensor Analysis.

3.1 Queueing Theory

Queueing theory was the first method used to optimise the security screening process in 1970. It was introduced in 1907 by Agner Kraup Erlang to describe the Copenhagen (Denmark) telephone exchange to deal with computer systems and call applications and aims to analyse and determine the length of queues, average waiting time, service and sojourn time. Recently, queueing theory has been used in other reallife application such as trains, hospital, banks, supermarkets and, most notably, airports [23].

Queueing theory is based on probability estimation and mathematical formalisation. Some parameters and values of queueing theory must be predetermined, such as service and arrival type, the number of servers or channels, buffer size, queue discipline, average service rate (μ) and average arrival rate (λ) [23]. A revised version of queueing theory was described in 1953 by Kendall, which is known as Kendall notation. The behaviour of this notation is denoted as A/B/C/D/E, where A is the inter-arrival time distribution, B is the service time distribution, C is the number of servers, D is the buffer size and E is the discipline, such as LIFO (last-in–first-out) or FIFO (first-in–first-out).

Models M/M/1, M/G/1 and M/D/1 are mainly used for security optimisation. In this research, an M/M/S queueing theory method assumption is used (where M represents a system with Poisson arrivals and exponentially distributed service time and S is the number of servers), to build the security screening area which is applicable for real-life airport security settings.

In general, according to [23], for M/M/S, the fraction of time a server is assumed to be busy can be expressed as:

$$C(s,a) = \frac{\frac{a^s}{s!(1-\frac{a}{s})}}{\sum_{k=0}^{s-1} \frac{a^k}{k!} + \frac{a^s}{s!(1-\frac{a}{s)}}} \tag{3.1.1}$$

The average waiting and response (sojourn) times, respectively, are:

$$E(W) = \frac{C(s,\alpha) \times \mu}{(1-\rho)S} \tag{3.1.2}$$

$$E(W) = E(W) + \mu = \frac{C(s,\alpha) \times \mu}{(1-\rho)S} + \mu \tag{3.1.3}$$

where c(s,α) is the fraction of time servers assumed to be busy, S is the number of servers, ρ is the utilisation or an average number of busy servers and α is the offered load or the number of passengers to arrive at the security area.

Several studies such as [69, 83, 19] have used M/M/1, M/D/1 and SM/M/1 as a queueing theory method to optimise the security screening process without considering variations in the service time and walking time factors.

3.2 Lindley Process

The Lindley equation is based on a discrete time stochastic process, which is used to describe the evolution of a queue length over time or to determine the average waiting time of passengers in a queue as expressed in equation 3.2.1. For example, the first passenger to arrive does not need to wait, so $W_n = 0$. The next passengers to arrive will need to wait if they arrive at a time before the previous passengers are served [12].

$$W_n + 1 = W_n + X_n - IA_n \tag{3.2.1}$$

where W_n is the waiting time of the previous passenger, X_n is the service time of passenger n , IA_n is the inter-arrival time, that is, the time between the previous and current arrival (when both passengers are in the service or waiting mode). Figure 3.1 illustrates the concept where Tn is the arrival time of passenger n and T_{n+1} is the arrival time of passenger $n + 1$.

This method has gained much popularity in this field due to its simplicity. For example, one equation is used to determine the average waiting and sojourn time instead of applying more than one formula as we observed in 3.1.1 and 3.1.2. Furthermore, Lindley's equation takes into consideration the time between the passenger's arrivals during the service time.

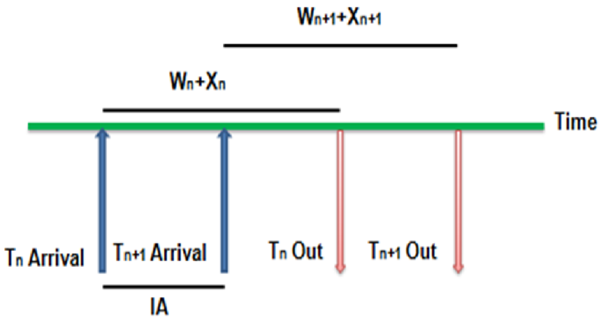


Figure 3.1: Lindley process principle

3.3 Particle swarm optimisation

Many methods have been used as an optimiser to find the optimal solution, such as a genetic algorithm, neural networks, evolutionary programming, evolution strategies, ant colony optimisation and particle swarm optimisation. PSO produces very accurate result for various parts of applications [17, 25]; therefore, we decided to incorporate PSO with the proposed queueing theory model, and hence, our proposed model is termed QT-PSO. Particle swarm optimisation is used in our model to optimise and find the optimal values for the walking time for the security screening process. Therefore, it is helpful to illustrate some of its fundamentals and to describe the process and parameters.

PSO is a simple mathematical and stochastic model used to describe the social behaviour of animals such as birds and fish to solve optimisation problems, which is based on swarm intelligence.Aswarm is a seemingly disorganised population of moving individuals that tend to cluster together, while each individual seems to be moving in a somewhat random direction. A PSO population member is called a particle.

Every particle is treated as a point in the D-dimensional space to be a possible solution to the problem. These particles have two vectors known as direction and velocity.

PSO had several advantages being efficient, effective on a variety of problems, easy to implement, a simple concept and a powerful algorithm which is based on communication and learning. It also contains a memory to memorise the optimal solution. The two patterns that implement the concept are known as personal and global best (P_best, G_best). P_best is a personal or individual best particle (personal best), and G_best is a global best for the best in the population (globalbest).

The PSO concept starts by initialising the population by assigning random velocities and positions, at every step changing the velocity for every particle, so the particles will cooperate to find the best location in the search space to find the optimal solution towards its P_best and G_best. Finally, the acceleration is weighted by a random term, with separate random numbers being generated for acceleration towards P_best, G_best. According to [25], if we have a particle (i), the update velocity and position at time ($t+1$) will be as follows:

$$V_i(t+1) = (wv_i(t) + r_1c_1(p_i(t) - x_i(t)) + r_2c_2(g(t) - x_i(t))) \tag{3.3.1}$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{3.3.2}$$

where $x_i(t)$ is the particle position for particle i, $v_i(t)$ is the particle velocity for particle i, r_1, r_2 are the uniform distribution random number between 0 and 1, c_1, c_2 are are acceleration coefficients ω is the inertia coefficient and the aim of introducing this coefficient is to control the impact of the previous velocity on the updated velocity, and $\omega v_i(t)$ is the inertia term.

3.4 Regression analysis

Regression analysis is a straightforward approach to mathematically model the relationship between two or more variables by using linear algebra. This approach uses a predictor variable (X) to forecast a quantitative response or dependent variable (Y).

$$Y = h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + ... + \theta_d x_d = \sum_{j=0}^{d} \theta_j x_j = \theta^T x \tag{3.4.1}$$

where $Y = h_\theta$ is the dependent or quantitative response, X_j, $j = 1...d$ are the independent variables. θ_0 is the constant or intercept and $\theta_j, j = 1...d$ are the coefficients. The aim of linear regression is to create a linear model that minimises the sum of the square of the residual error (SSE) which defines the cost function:

$$J(\theta) = \frac{1}{2n} \sum_{i=1}^{n} (h_\theta(x_{(i)}) - y_{(i)})^2 \tag{3.4.2}$$

In this model, we used a regression model that takes two features as input which is the number of passengers and the number of servers to predict the average waiting time.

The regression model is used in this research for comparison purposes to compare the actual state-of-the-art and the proposed method (QT-PSO).

3.5 One-Class Support Vector Machine

Given a set of data $X = \{x_i\}_{i=1}^{n}$, n is the number of training samples, OCSVM maps these samples into a high dimensional feature space using function ϕ through the kernel $K(x_i, x_j) = \phi(x_i)^T\phi(x_j)$. Then OCSVM tries to learn a decision boundary that maximally separates the training samples from the origin [74]. The primary objective of OCSVM is to optimize the following equation:

$$\min_{w,b,\xi,\rho} \frac{1}{2}\|w\|^2 - \rho + \frac{1}{\nu n}\sum_{i=1}^{n}\xi_i \tag{3.5.1}$$

$$s.t \qquad w.\phi(x_i) \geq \rho - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \ldots, n.$$

where ν is a user defined parameter used to control the rate of anomalies in the training data, ξ are the slack variables, w is a perpendicular vector to the decision boundary and ρ known as the bias term. This can also be expressed as minimization of the term $\frac{1}{2}\|w\|^2 - \rho$ appeared in Equation 3.5.1.

Considering the second term of the primary objective which is the minimization of the slack variables ξ for all points, the problem turns into a dual objective solved using the following quadratic programming formula .

$$\min_{\alpha} \frac{1}{2}\sum_{ij}^{n}\alpha_i\alpha_j K(x_i, x_j) \tag{3.5.2}$$

$$s.t \qquad 0 \leq \alpha_i \leq \frac{1}{\nu n}, \quad \sum_{i=1}^{n}\alpha_i = 1.$$

where $K(x_i, x_j)$ is the kernel matrix and α are the Lagrange multipliers. After we obtain the solution α of the dual optimization problem, the variable ρ is calculated to compute the anomaly score for a query sample using the following decision function:

$$g(x) = \sum_{i=1}^{n_s} \alpha_i K(x_i, x_{new}) - \rho. \tag{3.5.3}$$

Where n_s is the number of support vectors obtained from the constructed model. The OCSVM will use Equation 3.5.4 to identify whether a query point belongs to the positive class when returning a positive value and vice versa if it generates a negative value.

$$f(x) = sgn(g(x)) \tag{3.5.4}$$

3.6 Tensor Analysis

A tensor is a multi-way extension of a matrix to represent a series/set of matrices. It is often used when standard two way-data a.k.a matrices are not enough to capture the underlying structures inherited in multi-way data. Given a three-way tensor $X \in \Re^{I \times J \times K}$, we can decompose X into three matrices $A \in \Re^{I \times R}$, $B \in \Re^{J \times R}$and $C \in \Re^{K \times R}$, where R is the latent factors.

$$X_{(ijk)} \approx \sum_{r=1}^{R} A_{ir} \circ B_{jr} \circ C_{kr} \tag{3.6.1}$$

where "$\circ$" is a vector outer product. R is the latent element, A_{ir}, B_{jr} and C_{kr} are r-th columns of component matrices $A \in \Re^{I \times R}$, $B \in \Re^{J \times R}$and $C \in \Re^{K \times R}$.

The CANDECOMP/ PARAFAC (CP) method is often used for tensor factorization/decomposition. The main goal of CP decomposition is to decrease the sum square error between the model and a given tensor X. Equation 6.3.3 shows our loss function to optimize.

$$loss = L = \min_{A,B,C} \|X - \sum_{r=1}^{R} A_{ir} \circ B_{jr} \circ C_{kr}\|_f^2, \tag{3.6.2}$$

where $\|X\|_f^2$ is the sum squares of X and the subscript f is the Frobenius norm. It seems at first that the function presented in Equation 6.3.3 is a non-convex problem since it aims to optimize the sum squares of three matrices. However, we can reduce it a linear square problem by alternatively solve one matrix and fixed the other two. The alternating least square (ALS) technique can be employed now which repeatedly solves each component matrix by locking all other components until it converges. The rational idea of the least square algorithm is to set the partial derivative of the loss function to zero with respect to the parameter we need to minimize. In this sense, we need to calculate the partial derivative of L with respect to A, B and C alternatively. Below we show the three equations we need to optimize.

$$\begin{aligned} \frac{\partial L}{\partial A} &= 2(X - A \times (C \circ B)) \times (C \circ B) \\ \frac{\partial L}{\partial B} &= 2(X - B \times (C \circ A)) \times (C \circ A) \\ \frac{\partial L}{\partial C} &= 2(X - C \times (B \circ A)) \times (B \circ A) \end{aligned} \tag{3.6.3}$$

Setting $\frac{\partial L}{\partial A} = 0$ for optimal solution and using matrixization explained in Section 6.3, we can obtain the following three equations, one per mode to find A, B and C.

$$A = X_{(1)} \times ((C \odot B)^T)^{\dagger} \tag{3.6.4}$$

$$B = X_{(2)} \times ((C \odot A)^T)^{\dagger} \tag{3.6.5}$$

$$C = X_{(3)} \times ((B \odot A)^T)^\dagger \tag{3.6.6}$$

where † represents the pseudo-inverse of a matrix. The complete ALS algorithm is presented in Algorithm 1 .

Algorithm 1: Alternating Least Squares for CP

```
**Input**: Tensor $X \in \Re^{I\times J\times K}$
**Output**: Matrices $A \in \Re^{I\times R}$, $B \in \Re^{J\times R}$ and $C \in \Re^{K\times R}$
    Initialize $A, B, C$
    Repeat
      $A = \arg\min_{A} \frac{1}{2}\|X_{(1)} - A(C \odot B)^T\|^2$
      $B = \arg\min_{B} \frac{1}{2}\|X_{(2)} - B(C \odot A)^T\|^2$
      $C = \arg\min_{C} \frac{1}{2}\|X_{(3)} - C(B \odot A)^T\|^2$
        ($X_{(i)}$ is the unfolded matrix of $X$ in a current mode)
    until converged
```

3.6.1 Incremental Tensor:

Resolving the CP decomposition from scratch in online applications seems impractical in case of big training set of healthy samples. Therefore, there is an urgent need for incremental learning of tensor in online applications to update its components matrices when addition training data arrived. Similar to the ALS approach described in Algorithm 1, we fix the two components B and C then update the temporal mode A, and sequentially update the non-temporal modes B and C, by fixing the other two.

Update temporal mode C:

$$A = \arg\min_{A} \frac{1}{2}\|X^{(1)} - A(C \odot B)^T\|^2 = \arg\min_{C} \frac{1}{2}\left\|\begin{bmatrix} X^{(1)}_{old} - A_{old}(C \odot B)^T \\ X^{(1)}_{new} - A_{new}(C \odot B)^T \end{bmatrix}\right\|^2$$

The new time mode A_{new} can be by projecting the new arrived training sample $X_{new(1)}$ into the old matrices B and C. The new component A is then updated as follows

$$A = \begin{bmatrix} A_{old} \\ A_{new} \end{bmatrix} = \begin{bmatrix} A_{old} \\ X^{(1)}_{new}((C \odot B)^T)^{\dagger} \end{bmatrix} \tag{3.6.7}$$

where † represents the pseudo-inverse of a matrix

Update non-temporal modes B and C: The optimization functions for B and C can be written as $\frac{1}{2}\|X^{(2)} - B(C \odot A)^T\|^2$ and $\frac{1}{2}\|X^{(3)} - C(B \odot A)^T\|^2$, respectively. The resultant derivatives of these two functions w.r.t B and C and setting them to zero are:

$$B = \frac{\overbrace{X^{(2)} - (C \odot A)}^{P}}{\underbrace{(C \odot A)^T(C \odot A)}_{Q}} \quad (3.6.8) \qquad \text{and} \qquad C = \frac{\overbrace{X^{(3)} - (B \odot A)}^{U}}{\underbrace{(B \odot A)^T(B \odot A)}_{V}} \quad (3.6.9)$$

The computational time of $(C \odot A)$ and $(B \odot A)$ is costly since the resultant matrix size is very large. Therefore the simplified version of this equation can be estimated based on the old and new information of $X_{(i)^3_2}$ and A as follows:

$$P = P_{old} + X_{new}^{(2)}(A_{new} \odot B) \tag{3.6.10}$$

$$Q = Q_{old} + A_{new}^T A_{new} \circ B^T B \tag{3.6.11}$$

$$U = U_{old} + X_{new(2)}(A_{new} \odot C) \tag{3.6.12}$$

$$V = V_{old} + A_{new}^T A_{new} \circ C^T C \tag{3.6.13}$$

Chapter 4

A Framework for Optimizing the Waiting Time for Airport Security Screening using Multiple Queues and Servers

This chapter presents a novel queue formation method based on a queueing theory model and Lindley process known as QQT(Queues Queueing Theory) to optimise the security screening process with multi-servers in parallel to serve different number of passengers during different seasons, such as Christmas, Easter and school holidays, and time of the day, as this strongly influences the number of passengers, in order to improve the average waiting time in airport security areas.

Most of the reported work in chapter 2 section 2.6.1 only use the traditional queueing theory to optimise the security screening process following the M/M/1 and M/G/1 approaches [76] where M represents a system with Poisson arrivals and exponentially distributed service time; G is the general identical distribution, which is independent of the arrival process and must be a non-negative random variable; and finally 1 is the number of servers. These approaches simply multiply the outcome of

M/G/1 or M/M/1 by the total number of servers to determine the total passengers' average waiting time.

However, this assumption cannot be practically applied and does not produce accurate results for various reasons, such as differences in staff and passenger experience, knowledge of the security screening process and the existence of special needs passengers who require more processing time. Furthermore, walking time is another important factor that needs to be considered since it has a great influence on the passengers' waiting time. For example, passengers with special needs and families require longer walking time, thus more processing time. Finally, the variation among different screening machines, such as X-ray machines and metal detectors also affects the total processing time [65].
To the best of our knowledge, this study is the first attempt to use multiple parallel servers that are implemented in different scenarios to replicate airport security screening real-life settings.

The contributions of this chapter are summarised as follows:

- Construct a new queue formation model named QQT based on the Lindley process (queueing theory) to improve the average waiting time for multi-servers operating in parallel for security screening areas.
- Incorporate a new "walking time" parameter in our QQT model to improve prediction accuracy.
- Implement M/M/S/K for airport security screening system.

The rest of this chapter is organised as follows. Section 4.1 Security Screening procedure. Section 4.2 Queue Formation QQT and Module Description. Section 4.3 Experimental Setting. Section 4.4 Results, followed by a summary of this chapter in Section 4.5

4.1 Security Screening Procedure

Passenger security screening process procedures vary from country to country, and may be determined by the airport administration or based on government laws. For example, in Sydney airport when passengers have checked in and completed the immigration process, they go through the security process, where they are allowed to carry only a small bag known as a carry-on bag or hand luggage. This bag should not exceed a certain weight, as decided by the airline regulations and should contain less than 100 ml of liquid. Different types of machine are used in various airports to ensure that no prohibited or illegal items are being carried. These include traditional X-ray machines, CT (computed tomography) machines which can display bags in 3-D and allow 360-degree rotations, and metal detector devices that are used to scan passengers' bodies [13, 5].

4.2 Queue Formation QQT and Module Description

In general, most airports use a single passenger queue for the security screening process. However, in this study we consider three queuing scenarios: single, two and

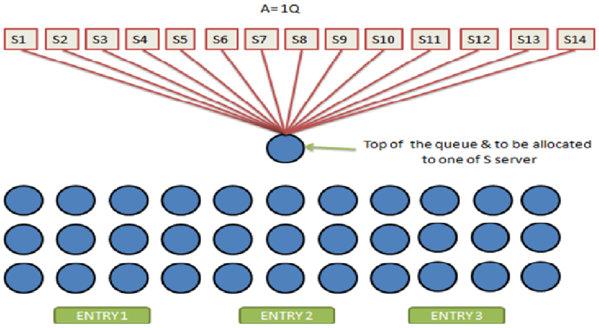


Figure 4.1: One Queue Scenario

three queues i.e. multiple server in parallel with different service distribution, i.e. M/M/S queueing theory assumption, to be applicable to real-world airport security settings. Also, we investigate their impact on passenger waiting time. Initially we assume we have fourteen available allocated to the security screening area.

The numbers of servers allocated for each queue according to scenarios of the proposed queue formation are presented as follows:

(i) One-queue scenario: all fourteen servers are allocated to one queue area and the passenger at the top of the queue is allocated to one of the S parallel servers, as shown in Figure 4.1.

(ii) Two-queue scenario: The total area A is divided into two equal queues or buffers and when there is an even number of servers, each area has the same number, whereas when there is an odd number of servers, the size of each buffer is based on the number of servers allocated to it. For example, in the case of fourteen servers, seven servers are allocated to each queue, as shown in Figure 4.2.

(iii) Three-queue scenario: The total area is divided into three queues. The size of each buffer is based on the number of servers allocated to it. For example, if there

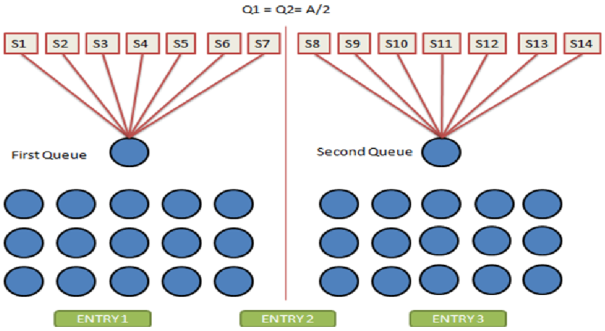


Figure 4.2: Two Queues Scenario

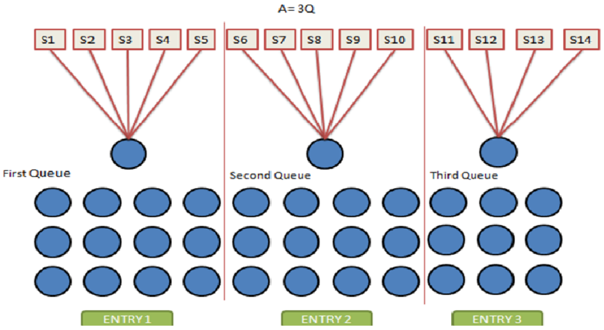


Figure 4.3: Three Queues Scenario

are fourteen servers, the first two queues are allocated five servers each and the third queue is allocated four servers, as shown in Figure 4.3.

Depending on the queue formation, a passenger is assigned by the security officers to join a certain queue (1, 2 or 3), based on the available free space and based on the type of passengers, such as normal passengers (aka economy passengers), crew, business passengers, families or passengers with special needs. Passengers are expected to be served by one of the S servers according to an exponential distribution with a mean service rate of μ and service completion rate of $1/\mu$ while the capacity of the area is assumed to be finite.

If N is the total number of passengers in the security area waiting to be served by S servers, then P_N is the steady state probability of N passengers in the system.

In general, $P_N Q_1 = \sum_{i=1}^{S} Q_1 S_i$ is the total number of passengers served by S servers in the first queue, while the total number of passengers served in the second queue is $P_N Q_2$, and the total number of passengers served in the system (all queues) is:

$$P_N Q_{total} = P_N Q_1 + P_N Q_2 + ... + P_N Q_N \tag{4.2.1}$$

The probability of blocking is defined as the chance that a customer will lose service due to high demand and lack of resources. For example, a probability of blocking of 0.01 means that 1% of customers will be denied or will lose service.

The probability of blocking for different numbers of queues for M/M/S/Q (where M represents a system with Poisson arrivals and exponentially distributed service time, S stands for the number of servers and Q is the number of queues) is:

$$P_Q = \sum_{q=1}^{Q} \sum_{r=1}^{s} \frac{\left(1 - \frac{\lambda\mu}{r}\right)\left(\frac{\lambda\mu}{r}\right)^q}{\left(1 - \frac{\lambda\mu}{r}\right)^{q+1}} \tag{4.2.2}$$

Equation 4.2.2 illustrates that the probability of blocking decreases when the number of queues increases.

In this model, queueing theory is used to build a system that follows M/M/S/K concept. A modified Lindley equation 4.2.3 is use to determine the average waiting time of N passengers.

The lindley equation 3.2.1 or Lindley process is based on dis create time stochastic process, which can be used to describe the evolution of a queue length over time or to determine the average waiting time of passengers in a queue. For example, the the first arriving passenger does not need to wait, so $W_n = 0$. Subsequent passengers will need to wait if they arrive at a time before the previous passengers are served [12]. The Lindley equation 3.2.1 is used in this work due to its simplicity and suitability for the security area queue formation. The simplicity is manifested by the applicability of a single equation for determining the waiting and sojourn time of passengers. Also, equations 3.1.1 and 3.1.2 do not take into consideration the difference in arrival times IA and the execution time of the previous passenger in determining the waiting time of passengers.

To be more applicable to queueing processes such as those at an airport security screening area and to produce accurate results, another parameter such as walking time must be incorporated in Lindley equation 3.2.1. Thus, the modified formula is:

$$W_n + 1 = W_n + X_n + W_walking - IA_n \tag{4.2.3}$$

where $W_{walking}$ is the time required by a passenger to walk when the passenger comes to the top of the queue and is the first to be served.

A modified Lindley equation 4.2.3 is used to determine the average waiting time of N passengers.

4.3 Experimental Settings

As in most implementations, some parameters must be predetermined or initialised before the simulation. These include service time (μ), number of arrivals, time between arrivals ($1/\lambda$), size of the security area, number of servers available (S) and finally walking time.

According to [44] the time required for a passenger to complete the security screening process varies between 15 and 60 seconds and is 25 seconds on average, so a time of 25 seconds is chosen as the mean average of the exponentially distributed service time. The number of passengers is initially considered to start at 500 and then sequentially increased by 500 passengers to reach a maximum of 8000. The inter-arrival rate between two consecutive passengers is exponentially distributed at the rate of 1 second. The inter-arrival rate between two consecutive passengers is exponentially distributed at the rate of 1 second. Small, medium and large security areas are considered, i.e., A=50, 75 and 100 m^2, and the total number of servers S is assumed to be 14 and they are available. Finally, the walking speed of a passenger with a carry-on bag who is at the top of the queue and the first to be served is assumed to be 1.5 seconds per metre. The three different sized areas are applied in each case to study the impact of this implementation on the average waiting time.

4.4 Results

We first consider the small area size of 50 m^2 and consider all three queuing scenarios of 1, 2 and 3 queues. Figure 4.4 shows the average waiting time when the number of passengers is varied between 500 and 8000. It also shows that the average waiting time per passenger for the three-queue formation is less than that of the two-queue, which in turn is less than that of the one-queue formation. This indicates that the average waiting time per passenger decreases when the number of queues increases, but the enhancement in waiting time between Q2 and Q3 is less than that between Q1 and Q2.

Figures 4.4 show that when the number of passengers increases from 500 to 8000 and the area size is 50 to 75 then 100 m^2, the average waiting time per passenger increases respectively. But if the size of the area is doubled from 50 to 100 m^2,

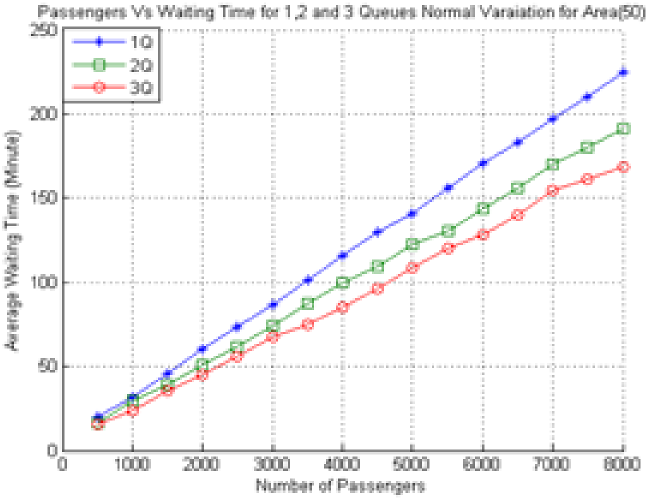


Figure 4.4: Normal variation - Area 50 m^2

the average waiting increases for the larger value only at the beginning, until the number of passengers reaches the maximum capacity of the area (saturation value), and then the increase becomes less for all scenarios. For example, when the area size is 50 m^2 and the passengers numbers 500, 5000 and 8000, the values of the average waiting time per passenger for the one-queue formation (1Q) are 19.9, 140.8 and 224.4 respectively, whereas if the size of the area changes to 100 m^2, the values are 34.4, 145.2 and 228.3 as shown in table 4.1. This increment in average waiting time looks very obvious at the beginning but subsequently becomes slower. This finding is similar in all the different cases. The impact of area size on average waiting time is due to the following:

Firstly, the waiting time of n+1 passenger is the cumulative sum of the previous n passengers, according to the Lindley equation or process 4.2.3.

Secondly, larger security areas can accommodate more passengers.

Thirdly, the walking time required for a passenger to reach the top of the queue is greater.

The results obtained from Tables 4.1 indicate that the three-queue formation is less sensitive to the increment in number of passengers than the one-queue formation. This finding is applicable to all scenarios, adding another advantage to the queue formation model. As shown in Figures 4.4, there is a noticeable difference between the graph of one queue and that of two and three queues. This could be justified by the walking time that passengers need to spend to reach the service lanes from the top of the queue. This time could be noticeably higher in the case of one queue, as the distance to the furthest service lanes is much greater than in the case of two queues. However, the differences in distance (and consequently in walking time) begin to decrease as the number of queues increases, which justifies the smaller difference between the waiting times of two queues compared to three queues. From this result, we believe that there will be no noticeable improvement if we increase the number of queues from three to four, and this also depending on the size of the security area and the number of servers.

Table 4.1: Waiting time per passenger vs. number of Queues

Waiting time per passenger (minutes) for 1, 2 and 3 queues normal variation for area (50 m^2)												
Number of pax	**500**	**1000**	**2000**	**2500**	**3500**	**4000**	**5000**	**6000**	**6500**	**7000**	**7500**	**8000**
1 Q	19.9	32	60.4	73.7	101.2	115.5	140.8	170.5	183.2	197	210	224.4
2 Q	16.6	29.5	50.7	61.6	87	99.9	122.7	143.3	155.7	169.8	179.9	191.2
3 Q	15.4	23.5	45.1	56.1	75.2	85.3	108.2	128.1	140.1	154.3	160.8	168.6
Waiting time per passenger (minutes) for 1, 2 and 3 queues normal variation for area (75 m^2)												
1 Q	24.7	35.4	63	75.8	104.4	115.3	144.5	172.3	184.6	199.4	213.8	224.4
2 Q	21.1	29.8	52.6	65	86	98.3	121.9	145.6	158.1	171.8	177.9	192.9
3 Q	18.7	26.8	47.9	58.5	77.2	89.2	109.9	131.7	138.2	152	164.3	173.5
Waiting time per passenger (minutes) for 1, 2 and 3 queues normal variation for area (100 m^2)												
1 Q	**34.4**	**39**	**65**	**79.8**	**105.6**	**118**	**145.2**	**175.2**	**185.9**	**202.3**	**214.2**	**228.3**
2 Q	**29.5**	**34.6**	**54.8**	**66.4**	**90.2**	**102**	**126**	**148**	**158.3**	**173.4**	**186.8**	**193.4**
3 Q	**25.1**	**29.9**	**49.7**	**58.5**	**79.7**	**91**	**110.7**	**132.3**	**143.3**	**152.1**	**162.9**	**176.9**

It is worth mentioning that we have chosen the model parameters that include service time, number of passengers and the time difference between arrivals to be as close as possible to real-life values. The service time formula will be verified when we collect real airport data and it will be implemented and discussed in the next chapter or contribution.

4.5 Summary

Airport security screening is essential to provide safety for the aviation industry and passengers travelling by air. Experienced officers and advanced technological instruments help to enhance security. However, the delay caused by security screening remains a concern. Technical methods have been proposed to model and optimise this process. Despite the achieved improvements, delays still present a major concern. In this chapter queueing theory M/M/S is used to build a system with multi-lane parallel servers to study the impact of queue formation and area size on the average waiting time. Unlike existing methods which uses M/M/1 and M/G/1 and do not consider the impact of multiple queues and area size. The result outperforms the proposed model compared to existing models, which addresses Contributions 1, 2 and 3 of this thesis. This implementation will be evaluated in the next chapter with real data collected from Sydney International Airport.

The second chapter proposes a new model QT-PSO to predict the average waiting times for multi-servers operating in parallel, which taking into consideration human and machine factors (variations). Further, the new model identifies the optimal walking time parameter to determine the optimal numbers of servers and security officers to maintain low average waiting time.

Chapter 5

Design of airport security screening using queueing theory augmented with particle swarm optimisation

Designing an efficient and reliable airport security screening system is a critical and challenging task. It is an essential element of airline and passenger safety which aims to provide the expected level of confidence and to ensure the safety of passengers and the aviation industry. In recent years, security at airports has gone through noticeable improvements with the utilisation of advanced technology and highly trained security officers. However, for many airports, it is important to find the best compromise between the capacity of the security area, the number of passengers, and the number of screening machines and officers to maintain a high level of security and to ensure that the cost and waiting times for passengers and airlines are at acceptable levels.

This Chapter proposes a novel method based on queueing theory augmented with particle swarm optimisation (QT-PSO) to predict passenger waiting times in a security screening context. This model consists of multiple servers operating in parallel and takes into consideration the complete scenario such as normal, slow and express lanes. Such an approach has the potential to be a reliable model that is able to assimilate variations in the number of passengers, security officers and security machines on the service time.

To evaluate our proposed method, we collected real-world security screening data from Sydney international airport from December to March for the two consecutive years of 2016 and 2017. The results show that our proposed QT-PSO method is superior to predict the average waiting time of passengers compared to the-state-of-the-art.

The contributions of this chapter are summarised as follows:

- Constructing a new model named QT-PSO based on the Lindley process (queueing theory) to predict the average waiting times for multi-servers operating in parallel in a security screening area.
- Optimising the new parameter was introduced to Lindley process in previous chapter named "walking time" which optimised by PSO.
- Modelling variations of servers and passengers to be compatible with real-world applications.
- Evaluating the performance of our model on real data collected from Sydney International Airport to demonstrate the effectiveness of the proposed method.

The rest of this chapter is organised as follows. Section 5.1 QT-PSO Method. Section 5.2 Data Collection. Section 5.3 Experimental results and discussion, followed by a summary of this chapter in section 5.4 .

5.1 QT-PSO Method

In this Chapter, we propose a QT-PSO method to model a security screening process. The main idea of QT-PSO is to implement the M/M/S queueing theory augmented with PSO to optimise the walking time parameter. Our QT-PSO method also considers variations in service time for different passengers and security officers. As shown in Figure 5.1, the QT-PSO framework is composed of multiple steps as follows

Step 1: Collect the data from the airport and divide it into two sets as explained in section 5.2. The first set is used to train the PSO to optimise $W_{walking}$. The second set is used to test the QT-PSO model to determine the average waiting time.
Step 2: Particle swarm optimisation is considered a general-purpose search process for optimization problems. It aims to optimise an objective function called a fitness function. In this setting, we followed a wrapper approach based on PSO to seek the global best walking time to minimize our fitness function defined in Equation (9) as illustrated in Figure 5.2.

PSO iteratively tries to select the best walking time which generates a similar result to the real data. In every iteration, PSO generates a new set of population walking time. This walking time is evaluated based on our fitness function to choose the best value that minimises the fitness function. For each evaluation, PSO optimises a different number of walking time sets, based on the number of servers used during this time. Some parameters in PSO must be predetermined before optimisation and this includes the following:
var_{min} and var_{max} are the minimum and maximum time to walk when the passengers reach the head of the queue and are the first to be served. The minimum time should

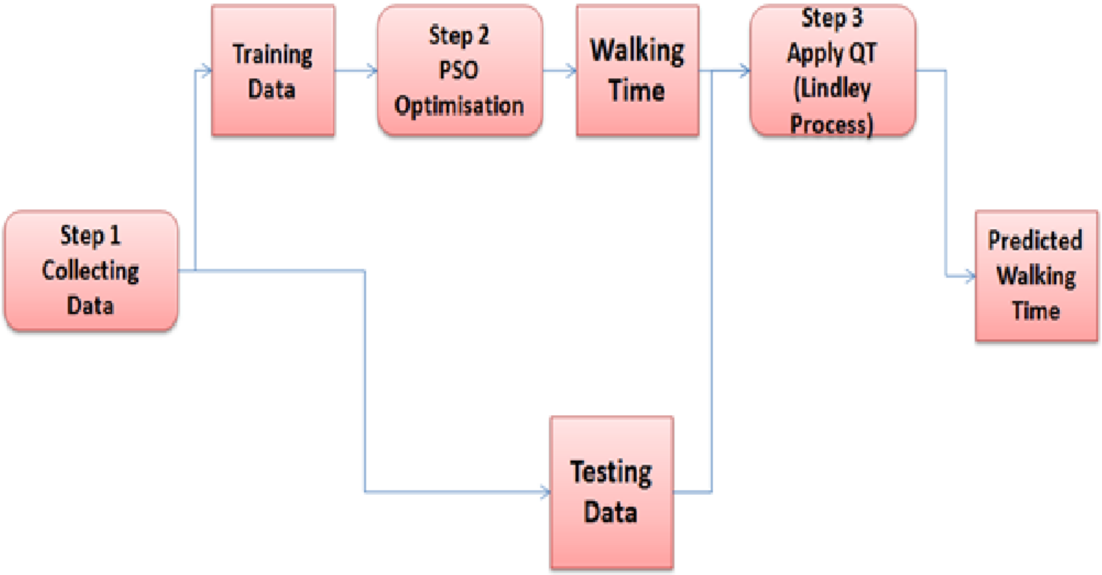


Figure 5.1: The QT-PSO Framework

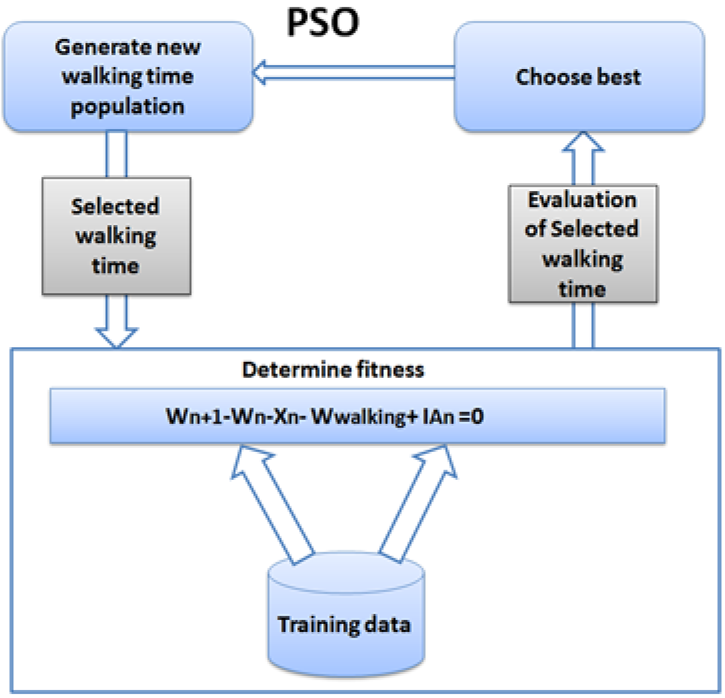

PSO
Generate new walking time population
Choose best
Selected walking time
Evaluation of Selected walking time
Determine fitness
Wn+1-Wn-Xn- Wwalking+ IAn =0
Training data

Figure 5.2: Weight-learning PSO

not be less than 1 sec to walk for any distance. However, the maximum time should not exceed 25 seconds, according to [72].

So, the lower bound of the decision variables $var_{min} = 1$ second and the upper bound of the decision variables $var_{max} = 25$ seconds.

The population size pop= 50, and the number of iterations= 100. Figure 5.3 shows the cumulative distribution frequency (CDF) of the optimised walking time. From the figure, we can see that the smallest value of the walking time is 4 sec which is more than the chosen var_{min}, while the maximum value of walking time is 10 seconds which is less than the selected var_{max}. This proves that the optimised values of walking time lie between the chosen boundaries of var_{min} and var_{max}.

Step 3: Build a system known as QT-PSO to produce an average waiting time similar to the actual result. The aim of this work is to design a security screening process similar to Sydney International Airport, as shown in Figure 5.4. The security screening area comprises seventeen servers to serve the passengers according to a first-come-first-serve (FCFS) principle. The passenger's arrival is assumed to follow a Poisson distribution, with a mean arrival rate named λ The successive time between consecutive arrivals is defined by $1/\lambda$. A passenger at the head of the queue will be assigned to be served by one of the S_N, where N is the number of servers equal to seventeen servers based on the current number of operating servers. Passengers will be served according to an exponential distribution with mean service rate of μ and service completion rate of $1/\mu$. Finally, $P_N S_i$ is the total number of passengers being served by server S_i in the security area.

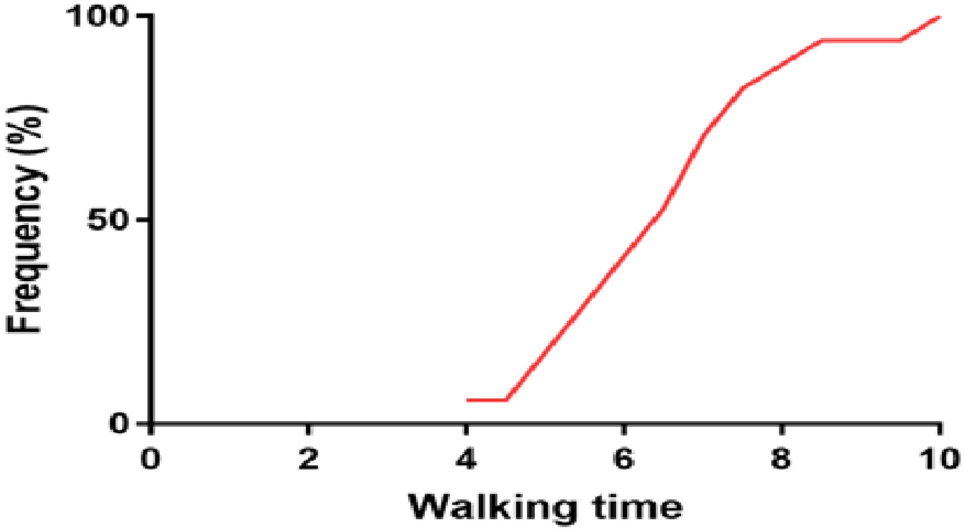


Figure 5.3: Walking time CDF

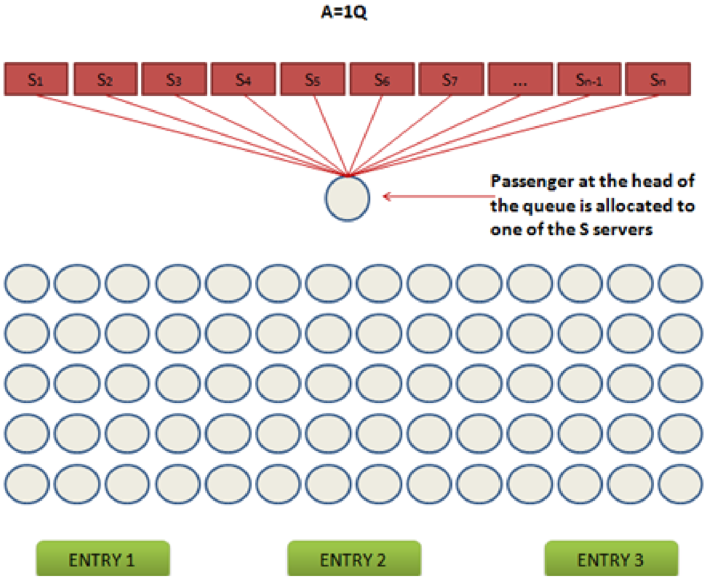

A=1Q
S1
S2
S3
S4
S5
S6
S7
...
Sn-1
Sn
Passenger at the head of the queue is allocated to one of the S servers
ENTRY 1
ENTRY 2
ENTRY 3

Figure 5.4: Security Area with Multi-servers in Parallel

Our proposed method is based on the Lindley equation which was defined in Equation 3.2.1.
We introduce a new walking time parameter to the Lindley process denoted by $W_{walking}$ in previous chapter (chapter 4) to consider the time required by a passenger to walk from the head of the queue to the target server. The modified formula is defined in Equation 4.2.3.
According to the experts and as proposed in [38], the time required for a passenger to complete the security screening process varies between 15 and 60 seconds for ordinary (Economy class) passengers. Business class passengers and crew require (aka express lanes) half the time (β) required by ordinary passengers whereas families and passengers with a special need (aka slow lanes) require three and half (α) more time than ordinary passengers. So, a time of 30 seconds is chosen as the mean average of the exponentially distributed service time.

Three different cases are considered while we optimise the walking time using PSO as follows:

1. Normal variation: this type of service is for ordinary passengers where the service time is assumed to be exponentially distributed with an average service mean and is then multiplied by a random number generator uniformly [1, 2]. The aim of this variation is to ensure variability among the passengers. The service time of passengers in the normal lane is expressed by the equation:

$$-\theta \ln(R) \tag{5.1.1}$$

where θ is the exponential distributed service time and (R) is random number [1, 2] generated uniformly.

2. The slow lanes are used to serve families and passengers with special needs. Such passengers may require α time, where α is equivalent to three and half times more the service time of ordinary passengers. The service time of passengers in the slow lane is expressed by the equation:

$$(-\theta \ln(R) \times \alpha) \tag{5.1.2}$$

3. The fast or express lane serves such as crew, business passengers and staff. This kind of passengers require half the service time of ordinary passengers. The service time of the express lane is calculated using the equation:

$$(-\theta \ln(R) \times \beta) \tag{5.1.3}$$

The variation in service time is applied in this model for several reasons. Firstly, according to [14], knowledge of the passenger screening process is different from one passenger to another and is influenced by previous travel experience which affects the service time. Secondly, efficacy can vary from one machine to another according to [5, 83, 59]. Thirdly, security officers vary in their experience, which could affect the time needed to screen luggage (service time). Screeners should be able to interpret images from scanners and distinguish between different items in a short period of time to be familiar with the content of the bag without opening it and to check whether the carry-on luggage contains any threat.

5.2 Data Collection

Our experiments are performed on real data collected from Sydney International Airport (T1). This data was collected from 17 lanes (aka servers). These servers are classified into three categories: express lanes, ordinary lanes, and slow lanes for special needs passengers and families. The data was collected at different periods of the year to ensure it incorporates all the seasonal variations including a busy period (e.g. Christmas and Easter) and a non-busy period which often occurs in January and February.

Moreover, the collected data also captures various types of passengers such as experienced passengers (who usually use the express lanes), non-experienced passengers (who usually use the ordinary lanes) and passengers with special needs or families who often use the slow lanes.

The data collected from Sydney International Airport was initially segmented in windows of 10 minutes. The resultant data windows contain 11036 samples for the year 2016 and 10453 samples for year 2017, where both data sets belong to the same time frame. Each has three features representing the number of passengers, number of servers and average waiting time. We built our training dataset using 2016 data to ensure it contains all the different periods of time and servers. The 2017 data set is used for testing to evaluate the performance of our proposed model. The allocation of time for training and testing data set contains both busy and non-busy periods.

5.3 Experiment results and discussion

For the sake of experiments and in order to have an objective comparison with the current implemented system in Sydney international airport, we conducted these experiments using one queue design.

Our initial experiment was to apply the PSO algorithm to optimise the walking time parameter introduced in Equation 4.2.3. The inter-arrival rate between two consecutive passengers is exponentially distributed at the rate of $1/\lambda$ seconds.

To evaluate the performance of our model, we used the mean square error (MSE) and standard error (s) defined in Equations 5.3.1 and 5.3.2, respectively. The paired t test was further conducted in these experiments to statistically compare the difference between our results obtained by QT-PSO and the actual airport data in addition to the other state-of-the-art methods.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(h_\theta(x(i) - y(i))^2 \tag{5.3.1}$$

$$S = \sqrt{MSE} \tag{5.3.2}$$

where $h_\theta(x(i))$ is the predicted average waiting time for our model, $y(i)$ is the actual average waiting time and n is the number of samples.

5.3.1 Results

In the following sections, we present the results of our QT-PSO model, and then we compare the QT-PSO model's results with the actual result (Airport), the state-of-the-art and the regression model. Finally, we draw a conclusion.

Generally speaking, the aims of this experimental study are twofold. One is to examine how closely the results of the proposed hybrid QT-PSO model are to the results obtained by the actual airport system. The second is to compare it with various existing systems, such as the state-of-the-art and the regression model.

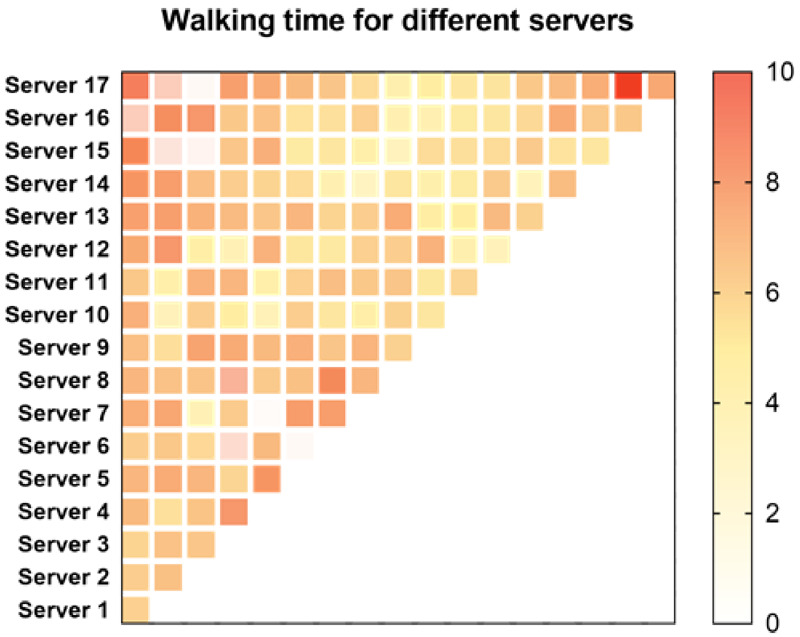


Figure 5.5: Walking Time (sec) Heat Map

5.3.2 PSO Result

Figure 5.5 shows the heat map of the optimised walking time values for a different number of servers. The x-axis represents time (seconds) and the y-axis represents the number of servers which varies from one to seventeen. The number of values of walking time is based on the number of servers in service at that time. From this figure, we can see that the smallest value for walking time is located in the middle. However, the largest values are on the far left and right sides. This is based on the location of the servers from the top of the queue and the first to be served. Also, we notice that when an extra server is added to increase the number of servers, the walking time value related to the new server is expected to increase too.

5.3.3 QT-PSO Result

The main aim of this experiment is to compare the standard error and similarity of the predicted average waiting time between the proposed model and the actual results from the airport.

From Table 5.1, although we take into consideration the variations of service time between passengers, security offices and machine, the proposed model able to produce an average standard error of 14.60. The percentage of similarity in the average waiting time between the actual and QT-PSO results is very high, which varies between 95% and 99% when the number of servers exceeds three. However, the proposed model does not perform well when the number of servers is less than four. For example, for server one, two and three, the similarity varies between 66%, 84% and 88% respectively, which is an accurate and similar result compared to the real model. Also, the total average waiting time similarity is approximately 94%. This shows that the accuracy of the proposed model for average waiting time exceeds 95% similarity for 14 servers. This reflects the robust and efficient nature of our model to meet varying airport requirements. The reason for the lower accuracy for servers one, two and three is because the proposed model has difficulty distinguishing between different types of passengers based on the process experience knowledge. However, in reality, when the number of servers exceeds 3, at least one server will be allocated to serve the express passengers, one to three servers for families and passengers with special needs and the remaining servers to serve ordinary passengers.

Further, we performed a paired t-test between the actual results and QT-PSO.

Table 5.1: Average waiting time and Standard Error (QT-PSO and Actual Result)

No of Servers	Average waiting Time (Second)		Similarity % QT-PSO_Actual	MSE (Second2)	Standard Error (Second)
	Actual Result	QT-PSO Model		(QT-PSO_Actual)	
17	389.002	397.406	97.885	48.14	6.938
16	174.557	180.167	96.886	198.966	14.105
15	170.535	175.397	97.227	160.71	12.677
14	256.651	253.112	98.621	205.557	14.337
13	201.054	196.695	97.832	245.742	15.676
12	161.045	165.578	97.262	164.089	12.809
11	160.1	159.545	99.653	163.206	12.775
10	159.125	162.772	97.759	57.98	7.614
9	179.37	177.626	99.027	442.227	21.029
8	150.931	148.6	98.455	216.483	14.713
7	146.261	140.907	96.339	586.994	24.227
6	120.291	116.852	97.141	285.481	16.896
5	98.314	93.915	95.525	44.275	6.653
4	135.742	129.267	95.229	889.174	29.819
3	84.228	74.721	88.712	499.668	22.353
2	77.983	65.972	84.597	122.123	11.05
1	26.285	39.775	66.084	20.549	4.533
Average	158.322	157.548	94.367	255.962	14.6

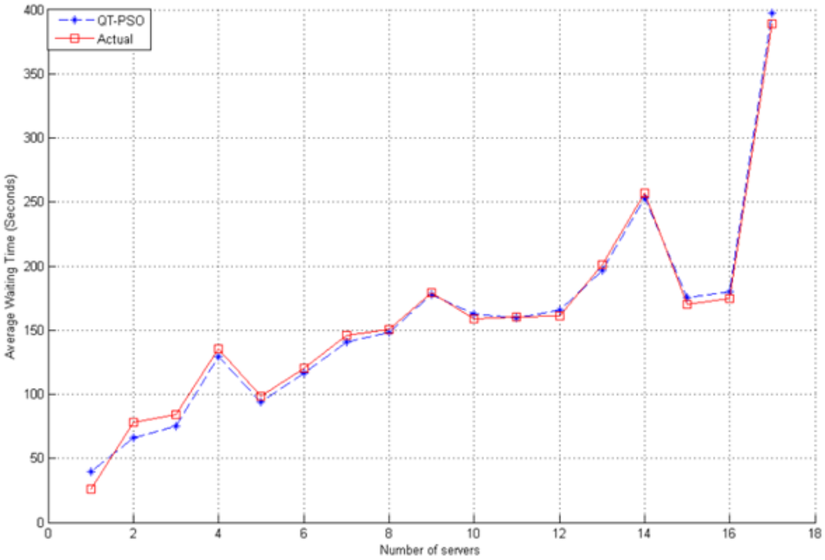


Figure 5.6: Average Waiting Time (QT-PSO-Actual Airport Result)

This test produces a p value of 0.637 which suggests that the difference is not significant. Thus, our prediction results are very similar to the airport results.

Figure 5.6 shows the average waiting for the proposed model's result compared to the actual result from the airport. From this figure, it becomes more obvious that the QT-PSO produces very accurate results compared to the actual results and it also follows the same trend. The only improvement needed is to enhance the prediction when the number of servers is below three.

5.3.4 Comparison with other methods

The following subsection compares the proposed QT-PSO with two other methods. One is the-state-of-the-art which is based on queueing theory proposed by [69], and the second method is the regression model proposed by [37, 29].

5.3.4.1 QT-PSO and State-of-the-Art Methods

Queueing theory was the first method used to optimise the security screening process in 1979. M/M/1, M/G/1 and S M/M/1 were the most preferred methods. So, the aim of this section is to compare QT-PSO with the state-of-the-art M/M/1.

During the implementation and testing of M/M/1, the following assumptions were considered:

Firstly, the system is designed to follow Poisson arrivals, exponentially distributed service time and only one server is in service at all times.

Secondly, to make an objective comparison between our proposed method and the M/M/1, the number of passengers assigned to both models is based on the number of passengers being served by S_N server from the real airport data, where N is the number of servers varying from one to seventeen.

Table 5.2 shows the difference in average waiting time between QT-PSO, M/M/1 and the actual results from the airport and Figure 5.7 illustrates the results.

From Table 5.2, we can observe the following: the M/M/1 model is able to produce a similar average waiting time compared to the actual result when the number of passengers is small. However, when the number of passengers increases to match the real number of passengers, the result of the M/M/1 model increases significantly to produce a higher result than the actual result. The S M/M/1 model produces a similar result for the first two servers and forecasts less than the actual time when the number of servers increases to more than 2. Furthermore, the overall average waiting time for both state-of-the-art models is very different from the actual, where the overall average results for M/M/1 is double the actual, the results of S M/M/1 is quarter of the actual however, the results of the proposed QT-PSO are similar to

Table 5.2: Average waiting time (QT-PSO, State of the Arts and Actual Airport Result)

	Average Waiting Time			
No of Servers	QT-PSO	State-of-the-Art M/M/1	Actual Result	State-of-the-Art S M/M/1
17	397.406	724.835	389.002	42.637
16	180.167	477.287	174.557	29.83
15	175.397	500.822	170.535	33.388
14	253.112	614.818	256.651	43.915
13	196.695	501.038	201.054	38.541
12	165.578	429.829	161.045	35.819
11	159.545	419.744	160.1	38.158
10	162.772	412.882	159.125	41.288
9	177.626	410.751	179.37	45.639
8	148.6	322.733	150.931	40.341
7	140.907	325.469	146.261	46.495
6	116.852	251.343	120.291	41.89
5	93.915	202.837	98.314	40.567
4	129.267	165.162	135.742	41.29
3	74.721	100.476	84.228	33.492
2	65.972	74.153	77.983	37.076
1	39.775	39.775	26.285	39.775
Overall Average waiting Time	157.548	351.409	158.322	39.42

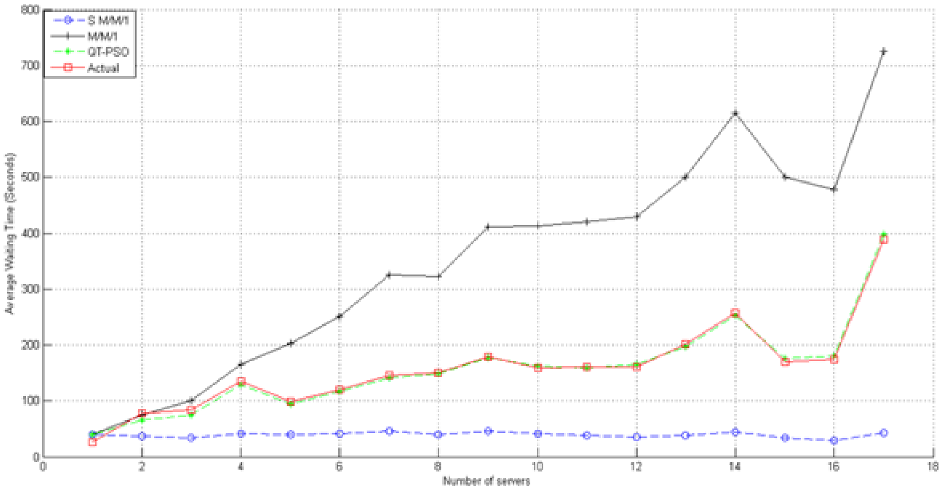


Figure 5.7: Average Waiting Time (QT-PSO, M/M/1, Actual Airport Result and M/M/1 S)

the actual result, as shown in Table 5.3.

Figure 5.7 illustrates the average waiting time for both state-of-the-art models compared to the actual and QT-PSO. From Figure 5.6, we note that the average waiting time for M/M/1 follows the pattern for the actual airport results and QT-PSO, but it is not close to the actual or QT-PSO, whereas the graph for the S M/M/1 model does not follow the pattern and also its forecast average waiting time is less than the actual.

Table 5.3: Standard Error(QTPSO- Actual, State-of-the-Art Methods- Actual)

	M/M/1- Actual	S M/M/1 - Actual	QT-PSO - Actual
Standard Error (Second)	193.537	120.488	14.6

Table 5.3 presents the standard error between the actual, the state-of-the-art (M/M/1, S M/M/1) methods and QT-PSO. The standard error for QT-PSO is 14.6 seconds whereas the standard error prediction for M/M/1 and S M/M/1 is 193.5 and 120.4 seconds, respectively. This means the interval error for QT-PSO is very small

compared with both state-of-the-art methods. This adds value to our proposed model which considers a system with multi-servers operates in parallel and with variations in scenarios.

[56] stated that the combination of S M/M/1 performs better and achieves good performance compared to M/M/S, whereas our experiment results show that our proposed QT-PSO model which is based on M/M/S produces much better and more accurate results compared to the real airport result.

5.3.4.2 QT-PSO and Regression Model

A regression model which is a part of machine learning can be used to predict the average waiting time. Our assumption is that there is a relationship between the number of servers and the number of passengers to predict the average waiting time. The regression model is used in this research as an additional comparison model.

The aim of this experiment is to compare the average waiting time and standard error of the predicted average waiting time between the regression model, the proposed model and the actual result from the airport.

From Table 5.4, we can see that the overall average waiting time for the actual result is 158.32 seconds. The regression model produced an overall average waiting time of 145.03 seconds, but the result of proposed model is 157.54 which is closer to the actual result. The standard error of our proposed model varies between 4.5 and 29.8 seconds for all servers with an average of 14.6 seconds. However, the standard error for the regression model varies between 2.6 and 101 seconds for all servers and 38.6 seconds on average.

From Table 5.4, we can also see that the regression model produces a close result to the actual, only when the number of servers is 1, 5, 8 and 10 whereas the proposed model produces a similar result to the actual when the number of servers exceeds three.

Table 5.4: Average Waiting Time and Standard Error (QT-PSO, Regression, and Actual)

	Average Waiting Time (second)			Standard Error (Second)	
No of Servers	Regression Model	QT-PSO Model	Actual Result	QTPSO-Actual	Regression-Actual
17	351.658	397.406	389.002	6.938	33.498
16	150.003	180.167	174.557	14.105	30.294
15	148.063	175.397	170.535	12.677	38.184
14	234.969	253.112	256.651	14.337	29.084
13	176.276	196.695	201.054	15.676	35.348
12	141.053	165.578	161.045	12.809	31.746
11	140.426	159.545	160.1	12.775	32.214
10	167.419	162.772	159.125	7.614	38.755
9	195.046	177.626	179.37	21.029	44.657
8	153.239	148.6	150.931	14.713	46.476
7	128.021	140.907	146.261	24.227	57.632
6	124.942	116.852	120.291	16.896	41.436
5	89.795	93.915	98.314	6.653	32.018
4	113.027	129.267	135.742	29.819	101.015
3	71.417	74.721	84.228	22.353	32.506
2	51.628	65.972	77.983	11.05	29.443
1	28.612	39.775	26.285	4.533	2.627
Overall Average Waiting Time	145.035	157.548	158.322	14.6	38.643

Figure 5.8 plots the average waiting time for the regression and the proposed model compared to the actual. From this figure, we can notice that the regression model produces a result which is less than the actual when the number of servers

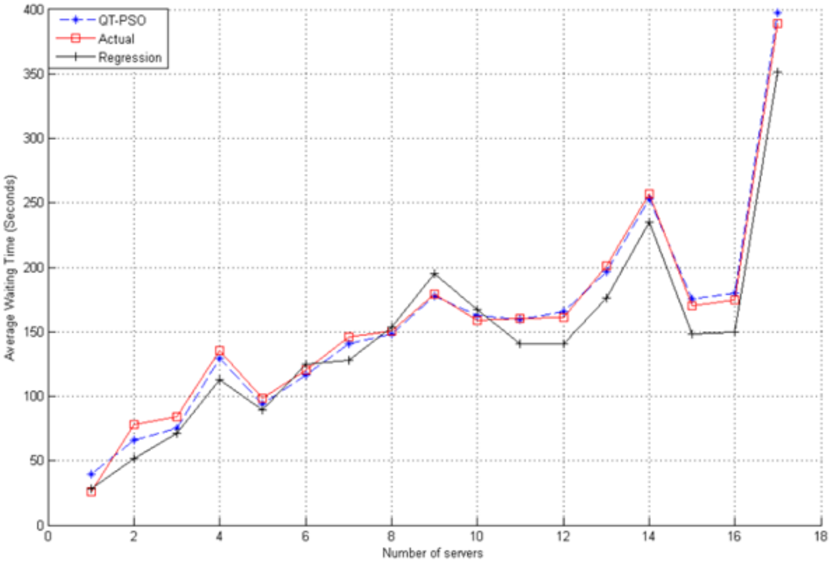


Figure 5.8: Average Waiting Time (QT-PSO, Actual Airport Result, and Regression)

exceeds 10 and is less than 5. Also, the plot of the average waiting time of the regression model does not follow the pattern of the actual, which makes it easy to see that QT-PSO produces very accurate results compared to the actual result and it also follows the same trend.

Further, we performed the paired $t-test$ between the actual result and QT-PSO, and between the regression model and the actual. This test produces a $p-value$ of 0.637 which suggests that the difference is not significant. However, the regression model produces a p-value of 0.002 at $p \leq 0.05$ which suggests that the difference is significant as shown in Table 5.5. Thus, our proposed model predicts results which are very similar to the airport result.

Figure 5.9 shows the box plot for the average waiting time, minimum and maximum values for the actual, QT-PSO and regression models per server. Overall, this

Table 5.5: Statistical T-test (Actual vs Regression, Actual vs QT-PSO)

Statistical Test	Regression-Actual	QT-PSO - Actual
t-value	3.704	0.479
p-value	0.002	0.637

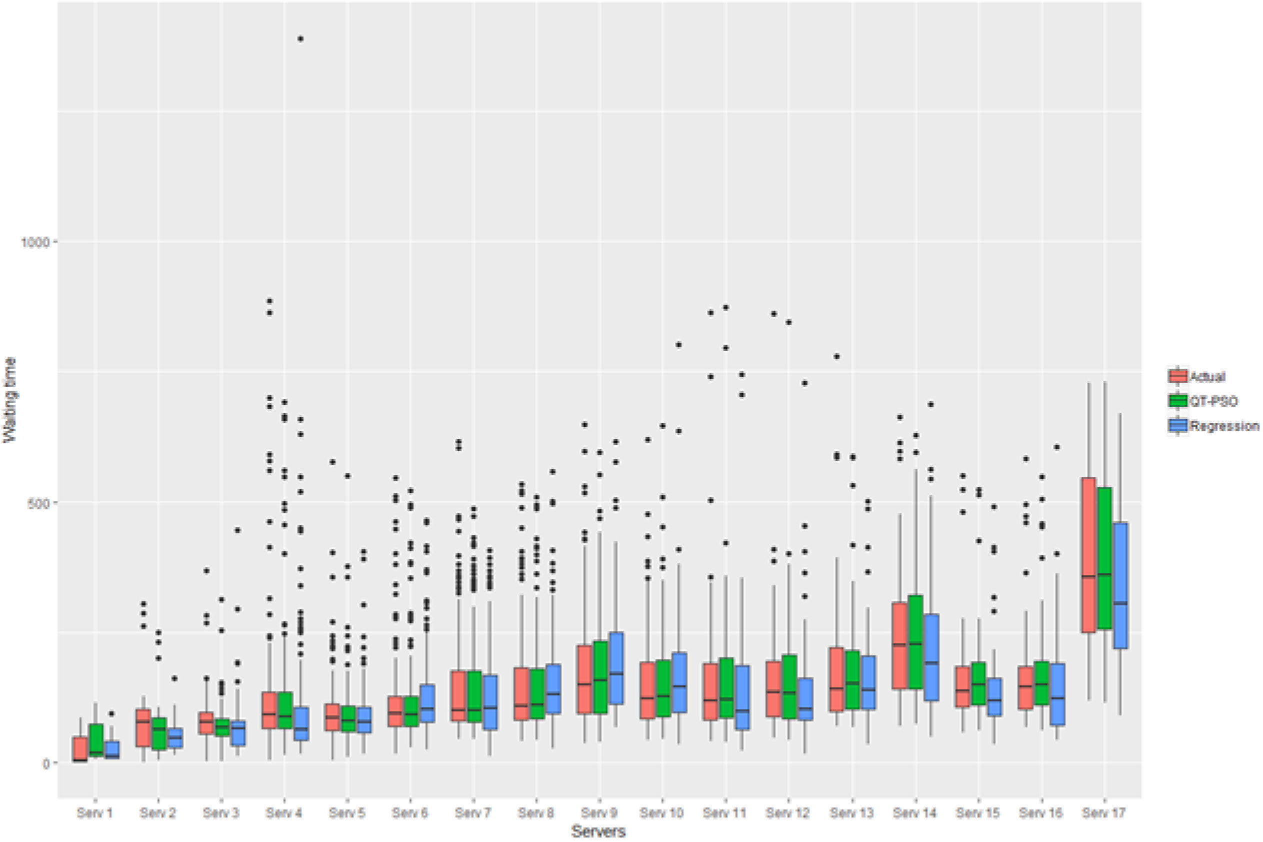


Figure 5.9: Average Waiting Time for All Servers (QT-PSO, Regression, and Actual)

figure shows that the average waiting time of QT-PSO is closer to the actual average than the regression model and per server when the number of servers is more than 3. Also, the minimum and maximum values per server for QT-PSO are approximately similar to the actual minimum and maximum. However, the regression model minimum and maximum values are not close to the actual minimum and maximum. We excluded the M/M/1 from this box plot as the average waiting time results were not close to the actual result.

From Figure 5.10 and the results that were discussed previously in sections 5.3.3, 5.3.4.1 and 5.3.4.2, we can arrive at the following conclusion. The regression model can produce an accurate average waiting time and a lower standard error than the state-of-the-art M/M/1 and S M/M/1 which means neither M/M/1 and S M/M/1 are applicable for a real case scenario for the airport security screening process and especially when multiple parallel lanes are used. However, QT-PSO produces a more accurate average waiting time than the regression model and the state-of-the-art methods. Also, the proposed model produces a lower standard error compared to the regression and state-of-the-art models. This study verifies that the QT-PSO which is based on the M/M/S system performs best in terms of efficiency in all cases and produces accurate results compared to the actual result provided by Sydney International Airport. This model gives an opportunity for different types of passengers to use the appropriate lanes to complete the security screening process. For example, experienced passengers, business class passengers and the crew can use the express or fast lanes, families and passengers with special needs use the slow lanes, and ordinary passengers use the normal lanes.

[56] stated that the combination of S M/M/1 performs better and achieves good

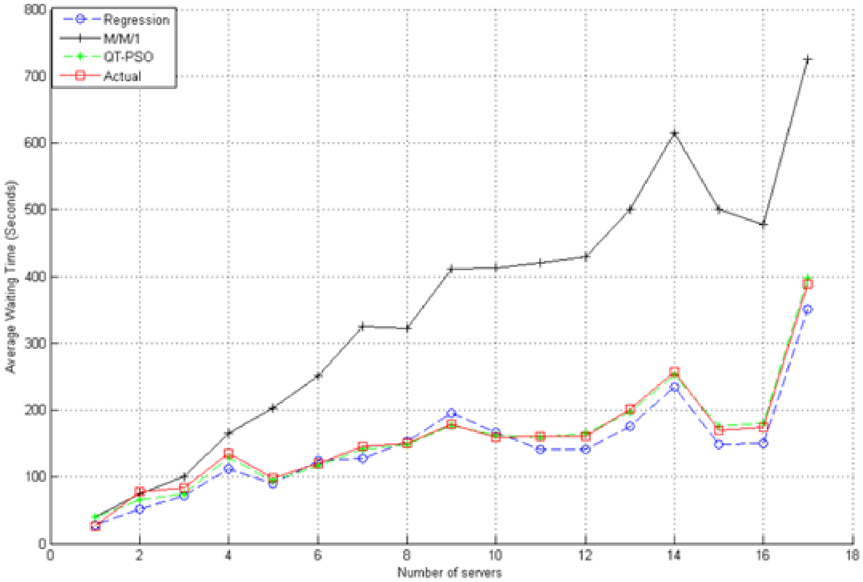


Figure 5.10: Average Waiting Time (QT-PSO, Actual Airport Result, M/M/1, and Regression)

performance compared to M/M/S, however our experiment results show that our proposed QT-PSO model which is based on M/M/S produces much better and more accurate results compared to the real airport result. However, [16] stated that M/M/S is much more efficient than the combination of S M/M/1.

The QT-PSO model recommends the number of servers required to serve a different number of passengers at different times to achieve an acceptable waiting time. However, the linear regression model does not consider or anticipate the different number of servers required to serve the different types of passengers at different seasons throughout the year, such as busy and non-busy periods. The regression model does not consider the variation in service time for passengers, security officers, and machines. Finally, the regression model does not consider the walking time factor and inter-arrival time.

5.4 Summary

Airport security screening is essential to ensure the safety of the aviation industry and passengers travelling by air. Different methods are used to determine the average waiting time for passengers to move through this complex system.

The main contributions of this chapter are as follows: firstly, various methods are considered, and a mathematical model is constructed. Secondly, all service variations for passengers, security officers, and machines are considered in our model. Thirdly, this paper presents a new novel hybrid model knows as QT-PSO based on the Lindley process (queueing theory) and particle swarm optimisation to predict passenger's average waiting time. Fourthly, a new parameter "walking time" is introduced in the Lindley formula and is incorporated and optimised in our QT-PSO model to improve prediction accuracy. Finally, various new findings are provided to show how the new proposed model is efficient for an airport security screening.

The results demonstrate that the QT-PSO method produces better average waiting time results and produce a lower standard error compared to the other state-of-the-art methods and the regression model. Finally, the results show that M/M/S is much more suitable and produces a more accurate result than M/M/1 and S M/M/1.

Extensive results were carried out in this paper to evaluate the performance of our model. These experiments used real datasets collected from the Sydney International airport. The results clearly indicated that our proposed QT-PSO model significantly produce average waiting time similar to the actual airport waiting time.

Furthermore, QT-PSO can anticipate the optimal number of servers required to obtain the best trade-off between the number of passengers and number of servers, which means it can send a notification to increase the number of servers if the number of the passengers increases. Therefore, it would be more beneficial for the security managers to adopt our proposed model. Our next chapter proposes a novel tensor-based learning method for anomaly detection in X-ray security screening systems based on tensor analysis augmented with a one-class classification model to enhance and fasten the security screening process.

Chapter 6

Anomaly Detection in X-ray Security Imaging

This chapter proposes a novel tensor based learning method for anomaly detection in X-ray security screening systems based on tensor analysis augmented with one-class classification model. Anomaly detection in X-ray security screening systems has earned a lot of interests in recent years and has attracted many researchers working in the area of machine learning. With the advances in computing technology, it is becoming more feasible to develop an approach for automated anomaly detection in security screening systems based on images collected via X-ray machines. Analyzing these X-ray images and constructing a detection model is considered as a challenging problem because of the lack or limited number of samples of anomalous objects. Furthermore, in the context of X-ray security screening system, the X-ray machine for scanning luggage generates a set of multi-view images for each event at a specific time. These collected images for each individual luggage are considered as multi-way data which are not only correlated with each other in time but also auto correlated in terms of angle views. Two-way matrix analysis which deals with each view-image as one event can not capture all of these correlation and relationships together. This approach may adds bias that could separate the distribution of the training data from different cameras. Thus, the extracted features will be separate images based on different view rather than anomalous and non-anomalous data. A naive approach would concatenate the multi-view images (aka multi-way data) for a certain event to form a single data instance at a specific time for anomaly detection in time dimension. However, unfolding the multi-way data and analyzing them using two-way methods may result in information loss and misinterpretation since it breaks the modular structure inherent in it. The contribution of this chapter is threefold:

- A tensor-based learning anomaly detection model in X-ray security imaging.
- A novel algorithm for optimizing the Gaussian kernel parameter inherent in OCSVM.
- A regularised alternating least square (RALS) method for tensor decomposition

The rest of this chapter paper is structured as follows. Section 6.1 presents our tensor-based learning approach for anomaly detection in X-ray imaging. Section 6.2 shows our data structure in X-ray machine settings. Sections 6.3 and 6.4 present our tensor analysis for data fusion and OCSVM for anomaly detection, respectively. Section 6.5 presents the experimental setup and Section 6.6 presents the results. Section 6.7 gives a summary of this chapter.

6.1 Tensor-Based Learning Approach for Anomaly Detection in X-ray Security Imaging

This thesis introduces a new novel method for anomaly detection in X-ray security image in multi-view settings. It performs data fusion of multi-view scanned images in a tensor data structure from where we extract the informative features. Further, it constructs a one-class support vector machine model using these features to detect anomalies. Specifically, our approach constructs first a three-way tensor image-node which contains: the multi-view images as one event, the number of pixels for each image, and the total number of events from the X-ray machine. We then use CANDECOMP/PARAFAC (CP) to decompose the tensor image-node into three subspace matrices which represents the latent information of each mode. in this sense, we propose a new regularised alternating least squares method to optimize the factor matrices in decomposition by adopting the L_1 regularization norm to achieve smooth representation of the factors. We further propose a new adaptive OCSVM model constructed based on the subspace matrix in time mode, extracted the from the three-way tensor, which contains events of positive instances (normal data).

6.2 Data Structure

Consider a set of s camera nodes located at different positions in the X-ray machine for scanning baggage's carried on a conveyor belt. The collected colour-mapped X-ray images are assumed to be a vector as $X_i = \left[x_1, x_2, \ldots, x_n\right]$; where $i = 1, \ldots, s$ are the camera nodes, and n is the total number of images at each camera node. Due to the lack of available data of X-ray scans with prohibited items, the training is achieved with the data corresponding to non-anomalous images (positive samples). Each positive training image sample $\{x_i\}_{i=1}^{n} \in \{X_i\}_{i=1}^{n}$ is an m-dimensional feature vector $P^p = p^1, p^2, \ldots, p^j m$, where $p = 1, \ldots, m$ are the number of pixels. The total number of pixels m depends on the image size, and the total number of data points (n) in X_i depends on the number of collected baggage's images we call it events. In our approach, we employ tensor learning as a data fusion method to merge images from the multi-angle image nodes into one multi-way image structure known as tensor $\mathcal{T}$. In the training phase, we extract the informative features from the $\mathcal{T}$ to construct a one-class model using the positive data. This model will be used later to classify a new raw X-ray image as either anomalous or non-anomalous.

6.3 Tensor Data Fusion

Our proposed approach employs tensor learning method as a data fusion step which merges data from a set of s image nodes into one data structure unit know as tensor. As we mentioned before, a naive approach would simply concatenate the pixels obtained from the multi-view images related to one particular event into one single two-dimensional image. However, unfolding the multi-view images data and analyzing them using two-way methods could leads to redundancy in reporting the events, and information loss since it breaks the modular structure inherent in the data. Therefore, a tensor data fusion approach will allow us to arranges the data from a set of multi-view images as one single image node $\mathcal{T}$ which we call it a tensor image-node. This tensor image-node $\mathcal{T}$ has data in a form of a three-way tensor $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$ where I represents the number of multi-angle images, J represents the number of pixels in each image, and K is the total number of events scanned by the X-ray machine. The structure of this tensor is shown in Figure 6.1

The aim now is to extract the latent features related to the multi-angle images which will be used later to construct a one class model for anomaly detention. In this paper we decompose our tensor using (CANDECOMP/PARAFAC decomposition) CP method due to its ease of interpretation compared with the Tucker method [46].

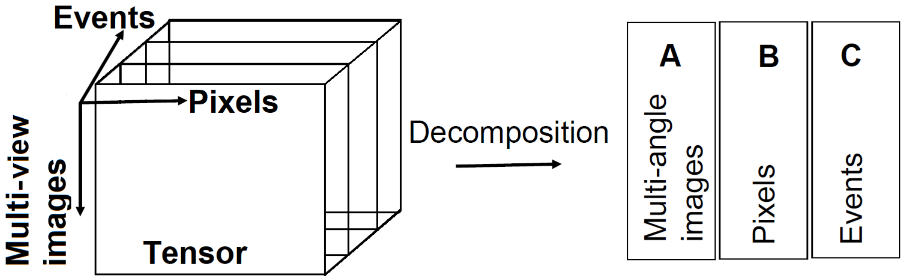


Figure 6.1: Multi-angle images fused in a tensor.

The tensor $\mathcal{X} \in \mathbb{R}^{I\times J\times K}$ is decomposed by CP into three matrices $A \in \mathbb{R}^{I\times R}$, $B \in \Re^{J\times R}$ and $C \in \Re^{K\times R}$ where R is the latent factors. Each matrix represents latent information for each mode or dimension. It can be written as follows:

$$\mathcal{X}_{(ijk)} \approx \sum_{r=1}^{R} A_{ir} \circ B_{jr} \circ C_{kr} \tag{6.3.1}$$

where "$\circ$" is a vector outer product.

We formulate the problem as follows:

$$\min_{A,B,C} \|\mathcal{X} - \sum_{r=1}^{R} \lambda_r \; A_r \circ B_r \circ C_r\|_f^2, \tag{6.3.2}$$

where $\|\mathcal{X}\|_f^2$ is the norm value which is the sum squares of all elements of $\mathcal{X}$, and the subscript f denotes the Frobenius norm. $A_r, B_r$and C_r are r-th columns of component matrices $A \in \mathbb{R}^{I\times R}$, $B \in \mathbb{R}^{J\times R}$ and $C \in \mathbb{R}^{K\times R}$.

6.3.1 L_1 Regularization for Learning Tensor:

Given a tensor $X \in \Re^{I\times J\times K}$, CP method decomposes it into three matrices A, B and C as shown in Fig. 6.1. Matrix A represents the view mode, B represents pixels mode and C represents time mode. The matrix $C \in \Re^{K\times R}$, which is associated with the event mode will be used later for constructing the anomaly detection model. This matrix has K rows, each of which represents one event data instance aggregated from all the multi-view images at a specific time.

The main goal of CP decomposition is to decrease the sum square error between the model and a given tensor X:

$$\min_{A,B,C} \|X - \sum_{r=1}^{R} \lambda_r \; A_r \circ B_r \circ C_r\|_f^2, \tag{6.3.3}$$

where $\|X\|_f^2$ is the sum squares of X, and the subscript f is the Frobenius norm. It seems at first that the function presented in Equation 6.3.3 is a non-convex problem since it aims to optimize the sum squares of three matrices. However, the problem can be reduced to a linear least squares problem by fixing two of the factor matrices, and solve only the third one. Following this approach, the ALS technique can be employed here which repeatedly solves each component matrix by locking all other components until it converges [70].

We remark that ALS can lead to sensitive solutions and it is not in general robust and hence motivates the need to incorporate the notion of penalty and regularization. The incorporation of regularization and penalization parameters into the L_1 norms make it possible to achieve smooth representations of the outcome and thus bypassing the perturbation surrounding the local minimum problem. The algorithm for CP decomposition using regularized ALS (RALS) is described in Algorithm 1. The L_1 penalty terms $||X||_{L_1} = \sum_{.} |x_.|$ enforces the intensity of sparsity in X.

6.3.2 Incremental Tensor:

Resolving the CP decomposition from scratch in online applications seems impractical in case of big training set of healthy samples. Therefore, there is an urgent need for incremental learning of tensor in online applications to update its components matrices when addition training data arrived. Similar to the RALS approach described

Algorithm 2: Regularized ting Least Squares for CP

```
Input: Tensor $X \in \Re^{I\times J\times K}$
Output: Matrices $A \in \Re^{I\times R}$, $B \in \Re^{J\times R}$, $C \in \Re^{K\times R}$, and $\lambda$
  1: Initialize $A, B, C$
  2: Repeat
   3: $A = \arg\min_{A} \frac{1}{2}\|X_{(1)} - A(C \odot B)^T\|^2 + \gamma_{X_A}||X_{(1)}||_{L_1}$
   4: $B = \arg\min_{B} \frac{1}{2}\|X_{(2)} - B(C \odot A)^T\|^2 + \gamma_{X_B}||X_{(2)}||_{L_1}$
   5: $C = \arg\min_{C} \frac{1}{2}\|X_{(3)} - C(B \odot A)^T\|^2 + \gamma_{X_C}||X_{(3)}||_{L_1}$
      ($X_{(i)}$ is the unfolded matrix of $X$ in a current mode)
  6: until converged
```

in Algorithm 1 and as proposed by [88], we fix the two components A and B then update the temporal mode C, and sequentially update the non-temporal modes A and B, by fixing the other two.

6.3.2.0.1 Update temporal mode C:

$$C = \arg\min_{C} \frac{1}{2}\|X_{(1)} - C(B \odot A)^T\|^2 = \arg\min_{C} \frac{1}{2}\left\| \begin{bmatrix} X_{old(3)} - C_{old}(B \odot A)^T \\ X_{new(3)} - C_{new}(B \odot A)^T \end{bmatrix} \right\|^2$$

The new time mode C_{new} can be estimated by projecting the new arrived training sample $X_{new(3)}$ into the old matrices A and B. The new component C is then updated as follows

$$C = \begin{bmatrix} C_{old} \\ C_{new} \end{bmatrix} = \begin{bmatrix} C_{old} \\ X_{new(3)}((B \odot A)^T)^{\dagger} \end{bmatrix} \tag{6.3.4}$$

where † represents the pseudo-inverse of a matrix

6.3.2.0.2 Update non-temporal modes A and B: The optimization functions for A and B can be written as $\frac{1}{2}\|X_{(1)} - A(C \odot B)^T\|^2$ and $\frac{1}{2}\|X_{(2)} - C(B \odot A)^T\|^2$, respectively. The resultant derivatives of these two functions w.r.t A and B and setting them to zero are:

$$A = \frac{\overbrace{X_{(1)} - (C \odot B)}^{P}}{\underbrace{(C \odot B)^T (C \odot B)}_{Q}} \quad (6.3.5) \qquad \text{and} \qquad B = \frac{\overbrace{X_{(1)} - (C \odot A)}^{U}}{\underbrace{(C \odot A)^T (C \odot A)}_{V}} \quad (6.3.6)$$

The computational time of $(C \odot B)$ and $(C \odot A)$ is costly since the resultant matrix size is very large. Therefore the simplified version of this equation can be estimated based on the old and new information of $X_{(i)_1^2}$ and C.

$$P = P_{old} + X_{new(1)}(C_{new} \odot B) \qquad (6.3.7)$$

$$Q = Q_{old} + C_{new}^T C_{new} \circ B^T B \qquad (6.3.8)$$

$$U = U_{old} + X_{new(2)}(C_{new} \odot A) \qquad (6.3.9)$$

$$V = V_{old} + C_{new}^T C_{new} \circ A^T A \qquad (6.3.10)$$

6.4 Adaptive One-Class Support Vector Machine Based Spatial Distance Algorithm

The one-class support vector machines (OCSVM) [74] has been widely applied in several application domains, such as medical [87], geology [43] and structural [8, 24] for anomaly detection purposes. It is an extension of the traditional two class support vector machines (TCSVM) method to handle the unsupervised learning problem when only samples from one class are available and the samples from other classes are very few or do not exists.

While the TCSVM learns to distinguish between two classes in a given data set by fitting a hyperplane that maximally divides both classes, the OCSVM learns a hyperplane that maximally separates the one-class data from the origin. In both cases, kernel functions such as Radial Basis function (aka Gaussian) are often employed when the data points are not easily separable in its original dimensional space. The kernel function aims to take the original positive data samples and increases their dimensionality to make them separable by a hyperplane.

The choice of kernel function in OCSVM represents a critical component for the success of this algorithm [28]. The Gaussian kernel function defined in Equation 1 is a popular and powerful kernel used in machine learning and it has turned out to be an appropriate kernel choice for OCSVM.

This kernel requires tuning for the proper value of Gaussian kernel parameter denoted by σ in Equation 6.4.1. This parameter has a great influence on the construction of a classification model as it controls how loosely or tightly the decision boundary fits the training data.

$$K(x_i, x_j) = \exp(-\frac{\|x_i - x_j\|^2}{2\sigma^2}) \tag{6.4.1}$$

As $\sigma \to 0$, the model will suffer from over-fitting since $K(x_i, x_j) \to 0$ will increase the number of support vectors. This will generate a tight decision boundary that yields to a complex over-fitted classifier lacking for the generalization. When $\sigma \to \infty$, OCSVM will suffer from under-fitting since $K(x_i, x_j) \to 1$ causes all the samples to be coincide at one point in the high dimensional space. This will generate a loose decision boundary that results in a simple trivial model which classifies all new samples as positive class.

This parameter can be simply optimized in TCSVM by stepping through a range of values for σ $[\sigma_{min}, \sigma_{max}]$ seeking for optimal performance of a model (e.g F1-score measure) with positive/negative instances in the training data. However, this problem is much more challenging in OCSVM where training validation is difficult due to the lack of negative instances.

In order to visualize the effect of parameter σ on the decision boundary in OCSVM, we used a synthetic two-dimensional Ring-shaped data set which allows us to visually observe the constructed OCSVM model. We applied OCSVM on the that dataset using different set of values for the parameter σ, and then we plotted the resultant decision boundary of OCSVM at each value of σ as shown in Figure 6.2.

It can be observed from this demonstration that the decision boundary starts tightly (as shown in Figure 6.2 (a)) when the value of σ was very small thus produces many interior and edge support vectors denoted by blue and red dots, respectively. This situation improved in Figure 6.2(b) and exceptional in Figure 6.2 (c) when all the support vectors in the training data exclusively located on the boundary of the training data i.e. edge samples. The value of σ was further increased in Figure 6.2(d) but it produces a loose decision boundary which may lead the model to miss-classify anomaly data points. Accordingly, the optimal value of σ should be selected once the spatial location of the support vectors exclusively exist on the boundary of the training data. In other words, the best value of σ is the one for which all the support vectors are edged support vectors. In this context, the problem is converted into inspection the spatial location of the support vectors in the training data rather than the classification performance (such as F1-score measure) which can't be calculated in one-class learning algorithms.

6.4.1 The Edged Support Vector Method

This thesis presents a novel algorithm called Edged Support Vector (ESV) for optimizing the Gaussian kernel parameter inherent in OCSVM. The idea of the ESV method is based on the spatial locations of the support vectors in OCSVM. Previously in Figure 6.2, we illustrated the relation between the geometric location of the support vectors with the appropriateness of the decision boundary. We observed that to generate a model converges on an optimal solution for the decision boundary which is judged neither too loose (under-fitting) or too tight (over-fitting), its support vectors must reside on the surface of the training data. On the other hand, the model may suffers the over-fitting when most of its support vectors located at the interior of the sample distribution form (denoted as interior support vectors from hereafter). Consequently, the first goal of ESV method is how to identify whether a support vector is sitting on the surface of the training data or not.

It is intuitive that an edge support vector x_e will have all or most of its neighbours sitting on one side of the hyperplane passing through x_e. Therefore, our edge pattern selection method uses a hard margin linear OCSVM to construct a hyperplane for each selected support vector x_s with its k-nearest neighbours data points. The method initially selects the k-nearest data points to each x_s, and then centralized them around x_s by computing the unit vectors $x_i - x_s$. This x_s stands now as an origin for its neighbours. Then we can apply a hard margin linear OCSVM to construct a hyperplane trying to separate these normalized unit vectors from their origin. If all or most of the vectors are separable (sitting on one side of the hyperplane), we consider x_s as an edge support vector x_e, otherwise it is consider as interior support vector.

In order to illustrate this theory of identifying the edge support vectors, we used a two-dimensional ring-shaped data set and we manually picked two edge samples from the concave and convex region as shown in purple colour in Figs 6.3(a) and 6.4(a). For each selected sample we computed its k-nearest neighbours points (shown in blue colour) and we centred them around their corresponding sample as shown in Figs 6.3(b) and 6.4(b). Then we constructed a hard margin linear OCSVM to separate the centred neighbours from the origin. As can be seen from Figs 6.3(b) and 6.4(b), all the nearest neighbour samples for the selected edge pattern were successfully separated from the origin and are sitting on one side of the plane. On the other hand and as we expected, the constructed hyperplane in Figure 6.5 was not able to separate from the origin all the k-nearest neighbours of an interior sample.

The algorithm of selecting the edge samples is described as follows: given a set of support vectors $x_s(i = 1, \ldots, n_s)$, the (normalize to unit length) of the difference of a certain support vector x_s with its k closest points x_j is computed as follows:

$$u_j^s = (x_{i_j} - x_s)/norm(x_{i_j} - x_s) \tag{6.4.2}$$

where $j = 1, \ldots, k$, $s = 1, \ldots, n_s$, and $x_{i_1}, \ldots, x_{i_k}$ are the k nearest neighbours of x_s.

Then we employ a hard margin linear OCSVM to separate v_j^k, the closest points to x_s, from the origin by solving the OCSVM optimization problem of the obtained unit vectors in Equation (6.4.2). Once we get the optimal solution $\alpha_j, j = 1, \ldots, l$ and calculating ρ, we estimate the value of the decision function using,

$$g(v_j) = \sum_{i=1}^{n_s} \alpha_i v_i . v_j - \rho. \tag{6.4.3}$$

The next step is to evaluate the optimization of the constructed linear OCSVM using the following equation

$$s = \frac{1}{k} \sum_{j=1}^{k} f(g(v_j)) > 0 \tag{6.4.4}$$

where s represents the success accuracy rate of the model. If all the closest points v_j^k are successfully separated from the origin (decision value is positive), then we count x_s as an edge support vector. In order to be lenient in identifying the edge support vectors, we used a threshold $1 - \gamma$ (γ is a small positive parameter), to control the number of edge support vectors by setting up a percentage for the acceptable success rate for each sample x_s. For $\gamma = 0.05$, if 95% of the closest points to a sample x_s are successfully separated from the origin, then the sample x_s can be still considered as an edge support vector, otherwise x_s is considered as an interior support vector. The pseudo code of the ESV method is presented in Algorithm 3.

The algorithm starts with the entire set of positive samples. Two parameters are used in this algorithm; k, the number of the nearest neighbours which has been thoroughly studied by [57] and they set $k = 5ln(n)$, and γ is a small positive number takes a value in the range of [0, 0.1].

The proposed method for ESV is described as follows: given a dataset of $x_i(i = 1, \ldots, n)$, the first step in the algorithm generates a candidate set for parameter σ. Empirically, σ values should be in the form of $[\ln d_{min}^2, \ln d_{max}^2]$ where the d_{min} and d_{max} are the minimal and maximal distances between the training samples, respectively.

Algorithm 3: The Edged Support Vector.

```
Input: A set of $n$ positive samples $x = \{x_i\}_{i=1}^{n}$
Generate a candidate set $\sigma_i$ $(i = 1, \dots, m.)$.
For each $\sigma_i$.
    • Generate the OCSVM model $(\mathbf{M})$
    • Set the value of the over-fitting factor $\mathcal{O} = 0$.
    • Get the set of support vectors $\{x_s\}_{s=1}^{n_s}$.
    • For each $x_s(s = 1, \dots, n_s)$.
        * Find the $k$ closest points to $x_s$: $x_j, j = 1, \dots, k$.
        * Calculate the unit vectors $v_j^k$ of $x_s$ according to (6.4.2).
        * Separate $v_j^k$ from the origin using a hard margin linear OCSVM.
        * Calculate the decision values of $v_j^k$ according to (6.4.3).
        * Calculate the success factor $s_i$ according to (6.4.4).
        * If $s_s \leq 1 - \gamma$ then
              $\mathcal{O} = \mathcal{O} + 1$ ($x_s$ is an interior support vector).
    • $\mathcal{O}_{\%} = \mathcal{O}/n_s$
    • Terminate when $\mathcal{O}$ remains 0;
Output: $\sigma_i$ and $(\mathbf{M})$.
```

This interval is then divided equally into m values and the exponential of these values comprise the candidate set of $\sigma_i(i = 1, \ldots, m)$ which represents the upper and lower bound of that parameter. Once we defined the candidate set of σ, we solve the optimization problem of OCSVM at each value of σ_i (starting from the minimum value) to identify the set of support vectors $x_s(i = 1, \ldots, n_s)$ in the training dataset. For each obtained solution, we inspect the spatial location of each support vector to see whether it is an interior sample or it is sample sitting on the surface of the training data. Then we estimate the percentage over-fitting factor $\mathcal{O}_{\%}$ with respect to the total number of the support vector n_s. Since we are starting with the minimum value of σ, we expect to have a large value for the factor $\mathcal{O}_{\%}$ at the beginning which tends to decrease as we approaching the maximum value of σ. The algorithm terminates when we reach the first zero value for $\mathcal{O}_{\%}$ because afterwards we actually start under-fitting the model as illustrated in Figure 6.2(d).

Figure 6.6 shows the optimization process of the ESV algorithm using the ring-shaped dataset. As can be seen, the over-fitting percentage factor $\mathcal{O}_{\%}$ was very high in the first few iterations before it is dropped to zero when the value of σ became equal to 0.04. This $\mathcal{O}_{\%}$ remains zero after that but the model starts to suffer the under-fitting problem as we shown that in Figure 6.2(d).

6.5 Experimental Setup

We conduct three different experiments using three types of datasets in the context of anomaly detection and X-ray security screening to illustrate and evaluate the performance of our proposed tensor-based learning model. The first dataset is the MNIST handwritten digit database [51], and the second one is the CIFAR dataset [50]. The

MNIST has a total of 70,000 images each with a size of 28×28 separated evenly into 10 classes. The CIFAR data has a total of 60,000 images each with a size of 32×32 which are also separated evenly into 10 classes. These two datasets are publicly available, and we have used them in the same way as in [86] and [1] to replicate their results and to be objective in comparison analysis. In this sense, for each of these two datasets we treat one class out of the ten classes as anomaly, while the remaining classes are considered as normal samples. This process results in ten sets of data from each datasets of which one class is considered as anomaly.

The third one is the Airport Baggage (AB) Dataset comprises 7,025 X-ray security screening events collected from Sydney airport. This data is naturally made up of two classes. The first class is the anomalous data related to objects such as knife, scissors and saw, etc. This class contains 417 events in contrast to the normal class which contains 6,608 events related to benign objects such as laptop, smartphone, brush, etc (see Figure 6.7). Each event comprises eight multi-angle X-ray images scanned from multiple perspectives. Each X-ray image has a size of 64×64 extracted using an overlap sliding window approach

For all datasets, we randomly select 80% of the normal data for training, and the remaining 20% are used for testing in addition to the anomalous data. For all experiments, the core consistency diagnostic technique (CORCONDIA) technique described in [18] is used to determine the number of rank-one tensors R when it is decomposed using the CP method. The ESV method proposed in [7] is used to tune the Gaussian kernel parameter σ in OCSVM. All the reported accuracy values were obtained using the area under the curve (AUC) of the Receiver Operating Characteristic (ROC) and F-Score (FS), defined as $F\text{-}score = 2 \cdot \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$, $Precision = \frac{\text{TP}}{\text{TP} + \text{FP}}$ and

$Recall = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$, where the number of true positive is abbreviated by (TP), false positive (FP), and false negative (FN), respectively.

6.6 Experimental Results and Discussions

6.6.1 The MNIST Dataset

We start the experiments by validating our tensor-based learning approach using MNIST dataset described in Section 6.5. This data doesn't have any multi-angle images, but for the sake of our approach, we deal with each eight images from the same class as one event, and then we arrange them in a three-way tensor $\mathcal{X}$. The resultant three-way training and test tensor data has a structure of $\mathcal{X}_{train} \in \Re^{8\times784\times6300}$ and $\mathcal{X}_{test} \in \Re^{8\times784\times2450}$, respectively.

The ALS method described in Algorithm 2 is initially applied to decompose the training tensor $\mathcal{X}_{train}$ into three matrices A, B and C, and the CORCONDIA method [18] selects $R = 10$ in ALS alogrithm. The resultant matrix $C \in R^{6300\times10}$ represents events data in time mode. To illustrate the performance of our tensor based learning approach, we plot the resultant matrix C in two dimensional spaces to see whether the tensor based features are able to distinguish between the ten different classes given in our data. It can be clearly observed from Figure 6.9(a) how only the first two features in matrix C can efficiently separate the 10 digits. These data are then used to construct an anomaly detection model using OCSVM with the ESV method for tuning the Gaussian kernel parameter. Figure 6.8(a) shows how the decision boundary of OCSVM with a tuned σ can precisely describe the shape of the training data, neither over-fitted nor under-fitted. On the other hand, the default value of σ in OCSVM leads to a loose decision boundary as can be seen in Figure 6.8(b).

Table 6.1: *F-score* of various methods.

	TOCSVM	GANomaly	EGBAD	AnoGAN	SGAN
MNIST	0.92±0.02	0.80±0.01	0.51±0.05	0.45±0.06	0.35±0.06
CIFAR	0.79±0.03	0.60±0.02	0.46±0.05	0.44±0.06	0.73 ±0.03
AB	0.95±0.02	0.66±0.01	-	-	0.92±0.06

For each new incoming X_{test} datum, we use Equation 6.3.4 to calculate C_{new} which represents the tensor-based features. The decision function defined in Equation 3.5.4 is then used to generate the anomaly score for C_{new} which specifies whether this new event is normal or abnormal. The average AUC accuracy for all models constructed at each digit designated as anomalous class is recorded at 0.92. The results of our proposed tensor-OCSVM (TOCSVM) method outperforms the other state-of-art methods proposed in [1, 86, 62, 6] which are described in Section 2.7 (see Figure 6.10(a) and Table 6.1). This is what we anticipated discovering from the OCSVM model which is trained based on latent features extracted from data fused in a three-way tensor.

6.6.2 The CIFAR Dataset

We apply the same process again here on CIFAR dataset described in Section 6.5. We fuse eight images from the same class in a three-way tensor $\mathcal{X}$ as one single event. The resultant three-way training and test tensor data has a structure of $\mathcal{X}_{train} \in \Re^{8\times1024\times5400}$ and $\mathcal{X}_{test} \in \Re^{8\times1024\times2100}$, respectively.

We decompose the training tensor $\mathcal{X}_{train}$ into three matrices A, B and C, and the CORCONDIA method [18] selects $R = 4$ in ALS alogrithm. The matrix $C \in R^{5400\times4}$ represents event data in time mode. We again plot here the resultant C matrix to illustrate the performance of our tensor based learning approach and to see whether the tensor based features are able to distinguish the ten different classes given in CIFAR data.

We can observe from Figure 6.9(b) that some classes were difficult to separate in this dataset such as samples belong to the classes of "cat", "bird" and "deer". On the other hand, tensor data analysis was able to capture meaningful features and to successfully distinguish classes of "truck", "dog", "car" and "plane". The constructed OCSVM model using C_{train} generates comparable results to the methods of [1, 86, 62, 2] (see Figure 6.10 (b) and Table 6.1) with an average AUC accuracy of 0.79.

6.6.3 The AB Dataset

The last experiments we conduct are based on our main AB Dataset collected from Sydney airport. Each event in this data is naturally comprises 8 multi-angle images. The three-way training and test tensor data has a structure of $\mathcal{X}_{train} \in \Re^{8\times784\times5286}$

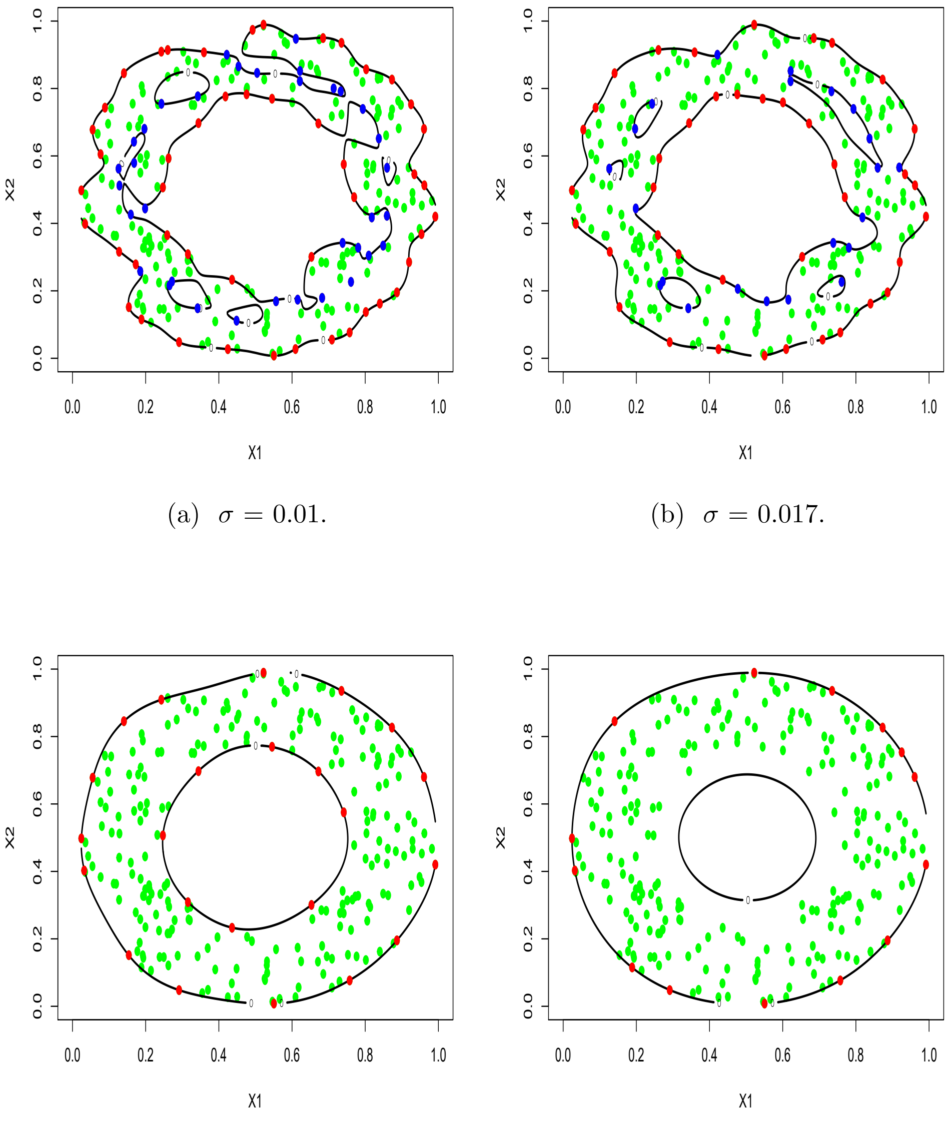
and $\mathcal{X}_{test} \in \Re^{8\times784\times1739}$, respectively. We initially apply ALS to decompose the training tensor $\mathcal{X}_{train}$ into three matrices A, B and C at $R = 16$ set by CORCONDIA. Then we construct our OCSVM model using $C \in R^{5286\times16}$ to classify each new incoming C_{test} datum whether its normal or abnormal. In this experiment we use the 10-fold cross validation methods to generate 10 different sets of training and test data. Our model generates an average AUC accuracy of 0.95 with a low false alarm rate. We compare these results to other related methods reported in the literature i.e GANomaly and SGAN. The resulted accuracies are shown in Figure 6.11 and Table 6.1 which demonstrates that our TOCSVM consistently outperforms the other approaches at each fold except for "fold 8". We didn't compare our experimental results of AB dataset to the results of EGBAD and AnoGAN since the source code of these two methods are not available.

These results justify our arguments on how tensor data analysis is able to fuse multi-way data in one structure from where we can extract the most informative and latent features which we can use to detect anomalous data. In fact, there is also another factor leads to these promising results which is the ESV method we used to tune the Gaussian kernel parameter σ in OCSVM.

6.7 Summary

This chapter presents an anomaly detection approach augmented with data fusion for X-ray security imaging in screening applications in airports. Our approach employs OCSVM based on latent features extracted from multiple images fused in a tensor via ALS technique. This approach performed successfully in three case studies based on two publicly available datasets and real X-ray baggage data collected from the

Sydney airport. The results show the importance of tensor analysis in extracting anomalous sensitive features as well as reducing the dimensionality of the data. In this approach, the OCSVM model was built using a self-tuning method of Gaussian kernel parameter for constructing an optimal decision boundary without suffering from the over-fitting nor the under-fitting problems. Our next chapter proposes the implementation this approach in a federated learning setting to build a central model learnt from data distributed at different airports.



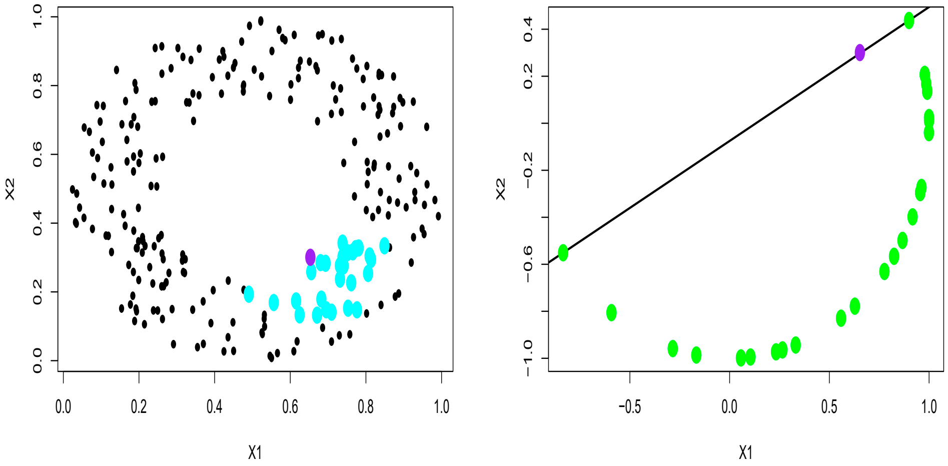
(a) $\sigma = 0.01$.

(b) $\sigma = 0.017$.

(c) $\sigma = 0.04$

(d) $\sigma = 0.16$

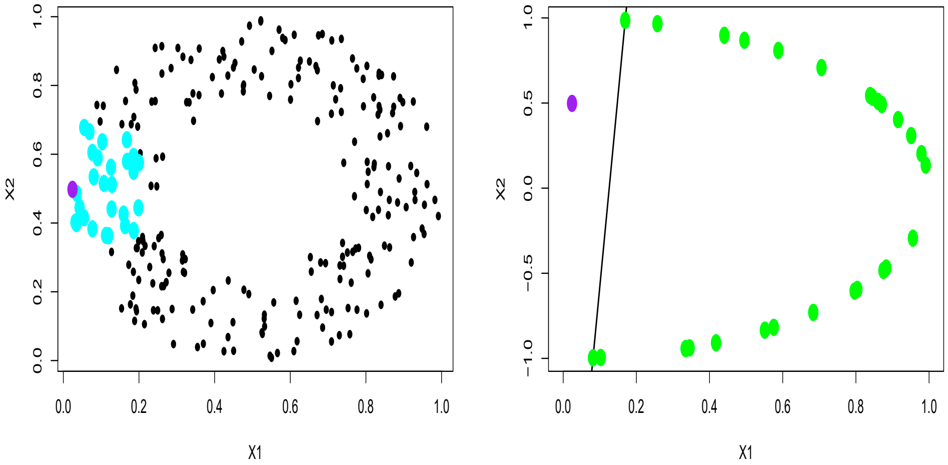
Figure 6.2: Experimental results using a ring-shaped dataset.



(a) $\sigma = 0.01$.

(b) $\sigma = 0.017$.

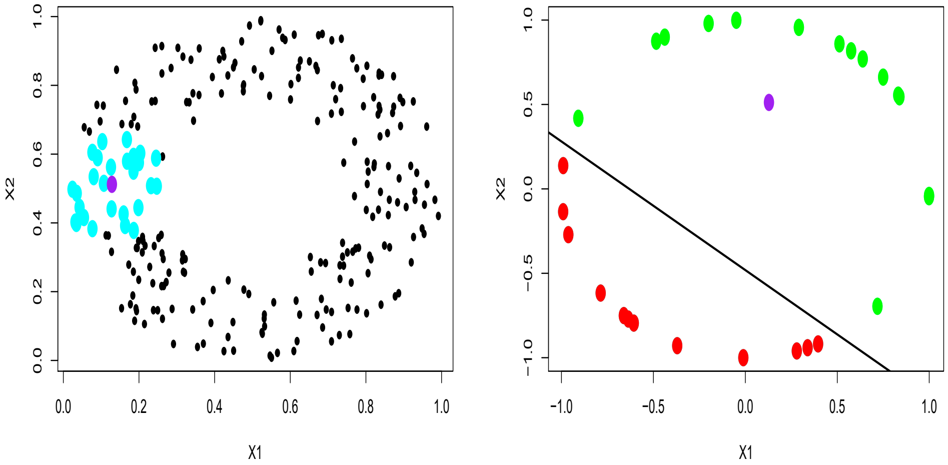
Figure 6.3: Concave edge sample selection.



(a) $\sigma = 0.04$

(b) $\sigma = 0.16$

Figure 6.4: Convex edge sample selection .



(a) $\sigma = 0.04$ (b) $\sigma = 0.16$

Figure 6.5: Interior sample selection.

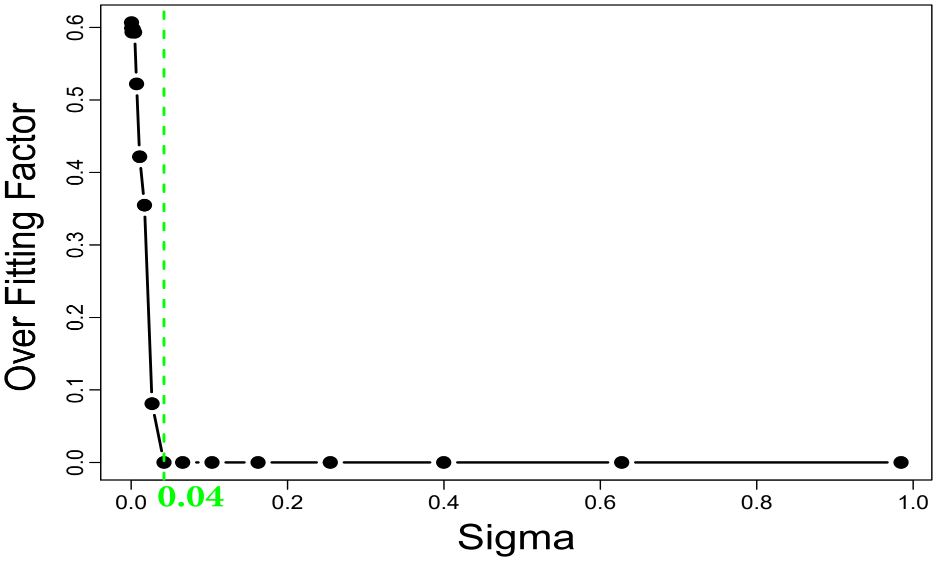


Figure 6.6

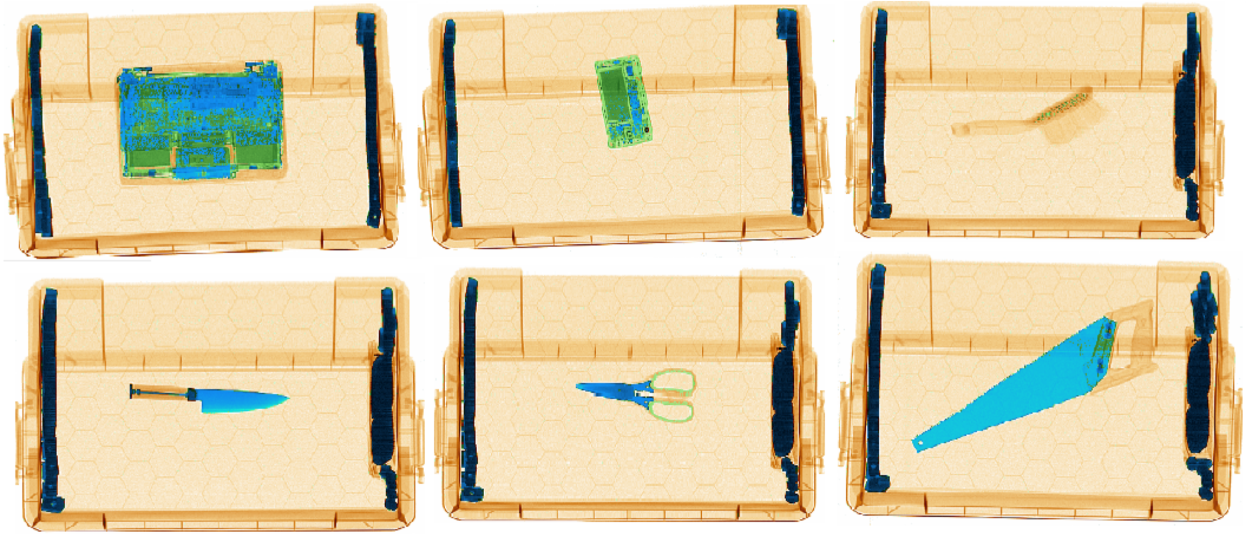
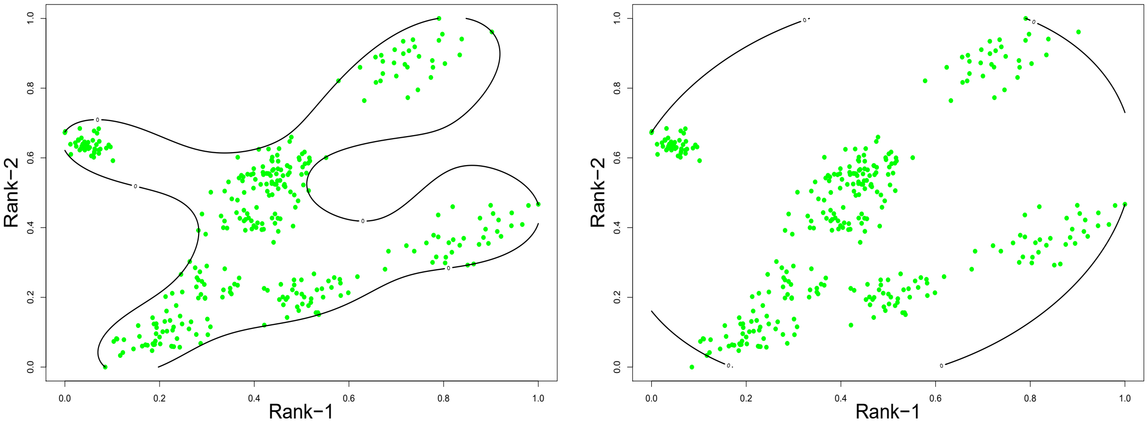


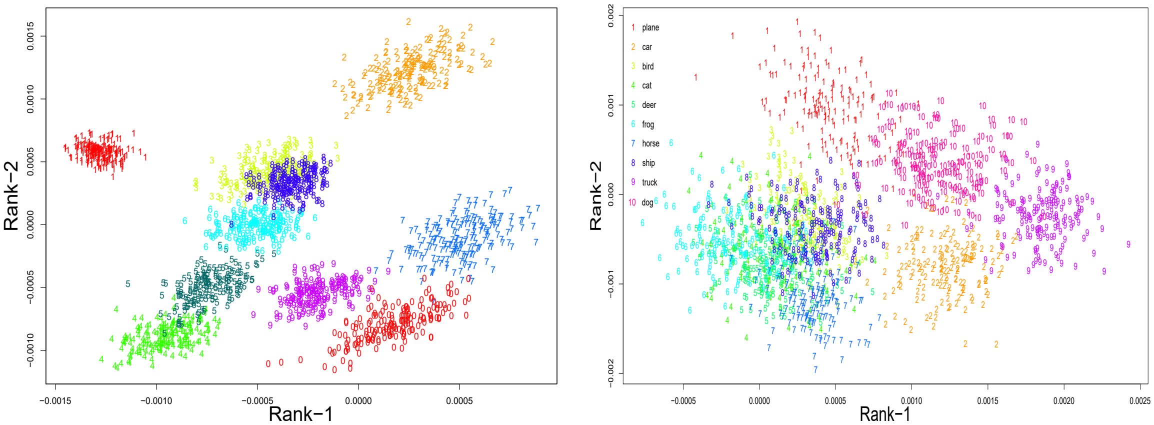
Figure 6.7: Exemplary X-ray images for the normal (row 1) and anomaly (row 2) classes in AB dataset.



(a) Tuned σ using ESV [7].

(b) Not tuned σ.

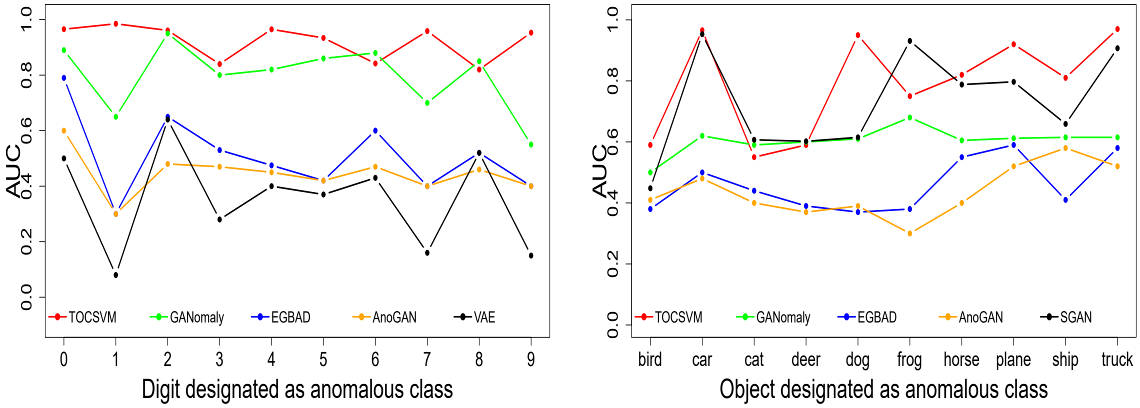
Figure 6.8: The resultant decision boundary of OCSVM.



(a) MNIST.

(b) CIFAR.

Figure 6.9: Two-dimensional plot of the resultant C matrix of tensor $\mathcal{X}$.



(a) MNIST.

(b) CIFAR.

Figure 6.10: Comparison of AUC accuracy on the ten classes.

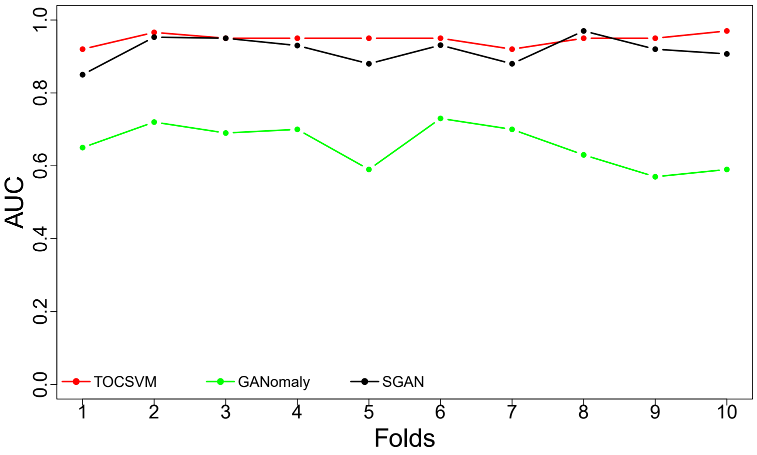


Figure 6.11: Comparison of AUC accuracy on the ten folds of AB dataset.

Chapter 7

A Federated Learning Anomaly Detection Approach for X-ray Security Imaging

This chapter proposes a federated learning (FL) approach for anomaly detection in X-ray security imaging using OCSVM. FL allows the central machine learning model to build its learning from a broad range of data sets located at different locations. It aims to train a shared centralized anomaly detection model using datasets stored and distributed across multiple clients/airports. This innovative machine learning approach can train a centralized model on data generated and located on multiple airports without compromising the privacy and security of the collected data. Moreover, it does not require transmitting large amount of data which can be a major performance challenge especially for real-time applications. FL approach can enable multiple airports to collaborate on the development of a central anomaly detection model by only sharing the model coefficients of each client/airport model rather than the whole data collected by all participating airports. The contribution of this chapter is twofold.

- A novel method of learning OCVSM model in FL settings

- An efficient communication method for coefficient's aggregation.

The rest of this chapter is structured as follows. Section 7.1 gives a brief review of problem formulation in federated learning setting. Section 7.2 presents our federated learning network with a novel method for Optimising coefficient's aggregation. Section 7.3 presents the experimental setup and the results. Section 7.4 gives a summary of this chapter.

7.1 Problem Formulation of OCSVM in Federated Learning

In FL setting, a set of S clients (airports) each of which has access to its local data, but they are connected to a central server to solve the following problem:

$$\underset{w \in \mathbb{R}^d}{min} f(w) := \frac{1}{C} \sum_{c=1}^{C} f_c(w_c) \tag{7.1.1}$$

where f_c is the loss function corresponding to a client c that is defined as follows:

$$f_c(w_c) := \mathbb{E}[\mathcal{L}_c(w_c; x_i)] \tag{7.1.2}$$

where $\mathcal{L}_c(w_c; x_i)$ measures the error of the model w_c (e.g. OCSVM) given the input x_i. The Sequential Minimal Optimization (SMO) is often used in the support vector machine. However, in the case of the nonlinear kernel model as in OCSVM, SMO does not suit the FL settings well. Therefore, we propose a new method for solving the OCSVM problem in FL setting using the SGD algorithm. The SGD method solves the above problem defined in Equation 7.1.2 by repeatedly updating w to minimize

$\mathcal{L}(w; x_i)$. It starts with some initial value of $w^{(t)}$ and then repeatedly performs the update as follows:

$$w^{(t+1)} := w^{(t)} + \eta \frac{\partial \mathcal{L}}{\partial w}(x_i^{(t)}, w^{(t)}) \tag{7.1.3}$$

Thus, we need now to formulate the cost function of OCSVM defined in Equation 7.1.4 to be optimized with SGD subject to the constraints $0 \leq \alpha_i \leq \frac{1}{\nu n}$

$$\max \mathcal{L}(\alpha) = \sum_{i}^{n} \alpha_i - \frac{1}{2} \sum_{i}^{n} \sum_{j}^{n} \alpha_i \alpha_j K(x_i, x_j) \tag{7.1.4}$$

$$s.t \qquad 0 \leq \alpha_i \leq \frac{1}{\nu n}, \quad \sum_{i=1}^{n} \alpha_i = 1,$$

where $K(x_i, x_j)$ is the kernel matrix and α are the Lagrange multipliers.

Let us assume $\mathcal{L}(\alpha)$ is given at the Lagrange multiplier α_k:

$$\max \mathcal{L}(\alpha_k) = \alpha_k - \frac{1}{2} \alpha_k^2 K(x_k, x_k) - \alpha_k \sum_{i=1, i \neq k}^{n} \alpha_i K(x_i, x_k)$$

The gradient of $\mathcal{L}(\alpha_k)$ at α_k is given as:

$$\nabla \mathcal{L}(\alpha_k) = 1 - \sum_{i=1}^{n} \alpha_i K(x_i, x_k) \tag{7.1.5}$$

Starting from an initial value of α, the gradient descent approach successively updates α as follows:

$$\alpha^{t+1} = \alpha^t + \eta \nabla J(\alpha^t) \tag{7.1.6}$$

Algorithm 4: Our OCSVM Using SGD

```
Require: : $X$, kernel function $\phi$ , $\eta$, $\epsilon$
  $K = \phi(x_i, x_j)_{i,j=1,\dots,n}$
Ensure: Initialize $\alpha$
  $t = 0$
  repeat
    $\alpha = \alpha_t$
    for $k \leftarrow 1$ to $n$ do
      $\alpha^k = \alpha^k + \eta(1 - \sum_{i=1}^{n} \alpha_i K(x_i, x_k))$
      if $\alpha^k \leq 0$ then $\alpha^k = 0$
      if $\alpha^k \geq \frac{1}{\nu n}$ then $\alpha^k = \frac{1}{\nu n}$
    end for
    $\alpha_{t+1} = \alpha$
    $t = t + 1$
  until $\|\alpha_t - \alpha_{t+1}\| \leq \epsilon$
  return $\alpha$
```

In SGD approach, the update rule for the kth component is given as:

$$\alpha^k = \alpha^k + \eta \nabla J(\alpha^k) \tag{7.1.7}$$

Recall that the optimization of α is subject to the constraints $0 \leq \alpha_i \leq \frac{1}{\nu n}$. Thus in the above update step, if $\alpha^k \leq 0$ we reset it so that $\alpha^k = 0$, and if $\alpha^k \geq \frac{1}{\nu n}$ we reset it so that $\alpha^k = \frac{1}{\nu n}$. Therefore, our OCSVM using SGD algorithm is presented in Algorithm 4.

In fact, the SGD algorithm in OCSVM focuses on optimizing the Lagrange multiplier α for all patterns x_i where $x_i : i \in [n], \alpha_i > 0$ are called support vectors. Thus, exchanging gradient updates in FL for averaging purposes is not applicable. Consequently, we modified the training process of SGD to share the coefficients of the features in the kernel space under the constraints of sharing an equal number of samples across each client C. In this sense, our SGD training process computes the kernel matrix $K = \phi(x_i, x_j)_{i,j=1,\dots,n}$ before looping through the samples. Then it computes the coefficients w after performing a number of epochs as follows:

$$w^{(t+1)} = \alpha K; \tag{7.1.8}$$

$$s.t \qquad \alpha = \alpha + \eta(1 - \sum_{i=1}^{n} w)$$

Each client performs a number of E epochs at each round to compute the gradient of the loss over its local data and to send the model parameters w^{t+1} to the central server S along with their local loss. The server then aggregates the gradients of the clients with a condition that a client should have generated a loss below the overall median loss, and applies the global model parameters update by computing the average value of all the selected clients model's parameters as follows:

$$w^{(t+1)} := \frac{1}{C} \sum_{i=1}^{C} w^{(t+1)}; \tag{7.1.9}$$

where C is the number of selected clients.

The server then share the $w^{(t+1)}$ to all selected clients in which each one performs another iteration to update $w^{(t+1)}$ but with setting $w_i^{(t)} = w^{(t+1)}$ as defined in the traditional FedAvg method.

7.2 Optimising coefficient's aggregation in Federated Learning

We also address the communication and aggregation problems that arise in the FedAvg method where several distributed models (clients) communicate with the central model to report its learning to the central model (the server). In fact, the performance of FedAvg is affected by the low-performing clients when their coefficients are shared are included in the aggregation at the central server. Therefore, we design a new method called Optimised-FedAvg (Opt-FedAvg) to mitigate this problem by selecting the optimal client's coefficients and filtering out the low-performing clients based on their local performance. The rationale behind this design is that at each update round, all the clients initially send their local loss values to the central model which in turn calculates the overall median loss. Then, only clients with their local loss value is lower than the overall median loss will be included in the aggregation at the central model, and the rest will be dropped until the next round. This would cater for different local clients, with varying progress, to perform more local computation. The completed algorithm of our Opt-FedAvg learning process is given in Algorithm 5

7.3 Experimental Results and Discussion

We conduct three different experiments using the same datasets we used in 6 i.e X-ray security screening, MNIST and CIFAR. For the sake of federated learning setting, we split each dataset into 5 subsets with 20% each to virtually represent five clients in our federated learning network. For all datasets, we randomly select 80% of the normal data for training, and the remaining 20% are used for testing in addition to the anomalous data. The same OCSVM setting is used here with ESV method described in Algorithm 3 for tuning the Gaussian kernel parameter σ. All the reported accuracy values were obtained using the area under the curve (AUC) of the Receiver Operating Characteristic (ROC) and F-Score (FS).

Algorithm 5: Optimised-FedAvg

```
    $K = \phi(x_i, x_j)_{i,j=1,\dots,n}$
    **Server executes:**
    Initialize $\alpha, w$
    **for** each client $c \in C$ **in parallel do**
        $w_c^{t+1}, loss_c^{t+1} = ClientUpdate(w^t)$
    **end for**
    find the median $\mathcal{M} = median(loss_c^{t+1})$
    select clients with $loss_c^{t+1} \leq \mathcal{M}$
    compute the average $w^{(t+1)} := \frac{1}{C}\sum_{c=1}^{C} w_c^{(t+1)}$
    send $w^{(t+1)}$ to the selected clients

    **ClientUpdate**$(w)$**:**
  **Require:** $X$, kernel function $\phi$ , $\eta$
  **Require:** $K = \phi(x_i, x_j)_{i,j=1,\dots,n}$
    **for** $k \leftarrow 1$ to $n$ **do**
        $\alpha^k = \alpha^k + \eta(1 - \sum_{i=1}^{n} w(k, i))$
        **if** $\alpha^k \leq 0$ **then** $\alpha^k = 0$
        **if** $\alpha^k \geq \frac{1}{\nu n}$ **then** $\alpha^k = \frac{1}{\nu n}$
    **end for**
    $w = \alpha K$
    compute $loss$
    **return** $w$ and $loss$ to server
```

7.3.1 Experiments on the AB Dataset

We initially study the effect of the number of local training epochs E on the performance of the four experimented federated learning methods as suggested in previous works [61, 22]. The candidate local epochs we consider are $E \in 5, 10, 20, 30, 40, 50$. For each of the candidate E, we run all the methods for 40 rounds and report the final f1-score accuracy generated by each method. The result is shown in Figure 7.1. We observe that conducting longer epochs on the clients improves the performance of Opt-FedAvg and FedPer, but it slightly deteriorates the performance of FedProx and FedAvg. The second experiment was to compare our Opt-FedAvg method to FedAvg, FedPer and FedProx in terms of accuracy and the number of communication rounds needed for the global model to achieve good performance on the test data. We set the total number of epochs E for Opt-FedAvg and FedPer to 50, and 30 for FedProx and FedAvg as determined by the first experimental study related to the local training epochs E. The results showed that Opt-FedAvg outperforms FedAvg, FedProx and FedPer in terms of local training models and performance accuracy. Table 7.1 shows the accuracy results of all experiments using *F-score*. Although no data from the anomalous classes has been employed to construct the central model, each local client model was able to identify the anomalous events with an average *F-score* accuracy of 0.85 ± 0.02.

7.3.2 Experiments on the MNIST Dataset

Similar to the experimental setup we did in Chapter 6, we treat one class out of the ten classes as anomaly, while the remaining classes are considered as normal samples. This process produces ten sets of data of which one class is considered as anomaly. The reported *F-score* accuracy is the overall average over the tens datasets.

We also study here effect of the number of local training epochs E on the performance of the four experimented FL methods. The results are summerised in Figure 7.2. Both Opt-FedAvg and FedPer benefit from doing longer epochs on clients in contrast to FedProx and FedAvg which are slightly deviated from the optimal convergence. The second experiment was to compare our method to FedAvg, FedPer and FedProx in terms of accuracy and the number of communication rounds needed for the global model to achieve good performance on the test data. We set the total number of epochs E for Opt-FedAvg and FedPer to 40, and 20 for FedProx and FedAvg as suggested from Figure 7.2(b). As shown, Opt-FedAvg outperforms FedAvg, FedProx and FedPer in terms of local training models and performance accuracy. Table 7.1 shows the accuracy results of all experiments using *F-score*.

7.3.3 Experiments on the CIFAR Dataset

As we did with MNIST dataset, we selected one class out of the ten classes to be our anomalous samples, and we considered the remaining classes as normal samples. We reported the overall average *F-score* accuracy over the obtained tens datasets. We also study here effect of the number of local training epochs E on the performance of the four experimented FL methods. The results are summerised in Figure 7.3. Similarly, both Opt-FedAvg and FedPer benefit from doing longer epochs on clients in contrast to FedProx and FedAvg which are slightly deviated from the optimal convergence. We run another experiment to compare our method to FedAvg, FedPer and FedProx in terms of accuracy and the number of communication rounds needed for the global model to achieve good performance on the test data. We set the total number of epochs E for Opt-FedAvg and FedPer to 40, and 20 for FedProx and FedAvg as suggested from Figure 7.3(b). Again here, Opt-FedAvg outperforms FedAvg, FedProx and FedPer in terms of local training models and performance accuracy. Table 7.1 shows the accuracy results of all experiments using *F-score*.

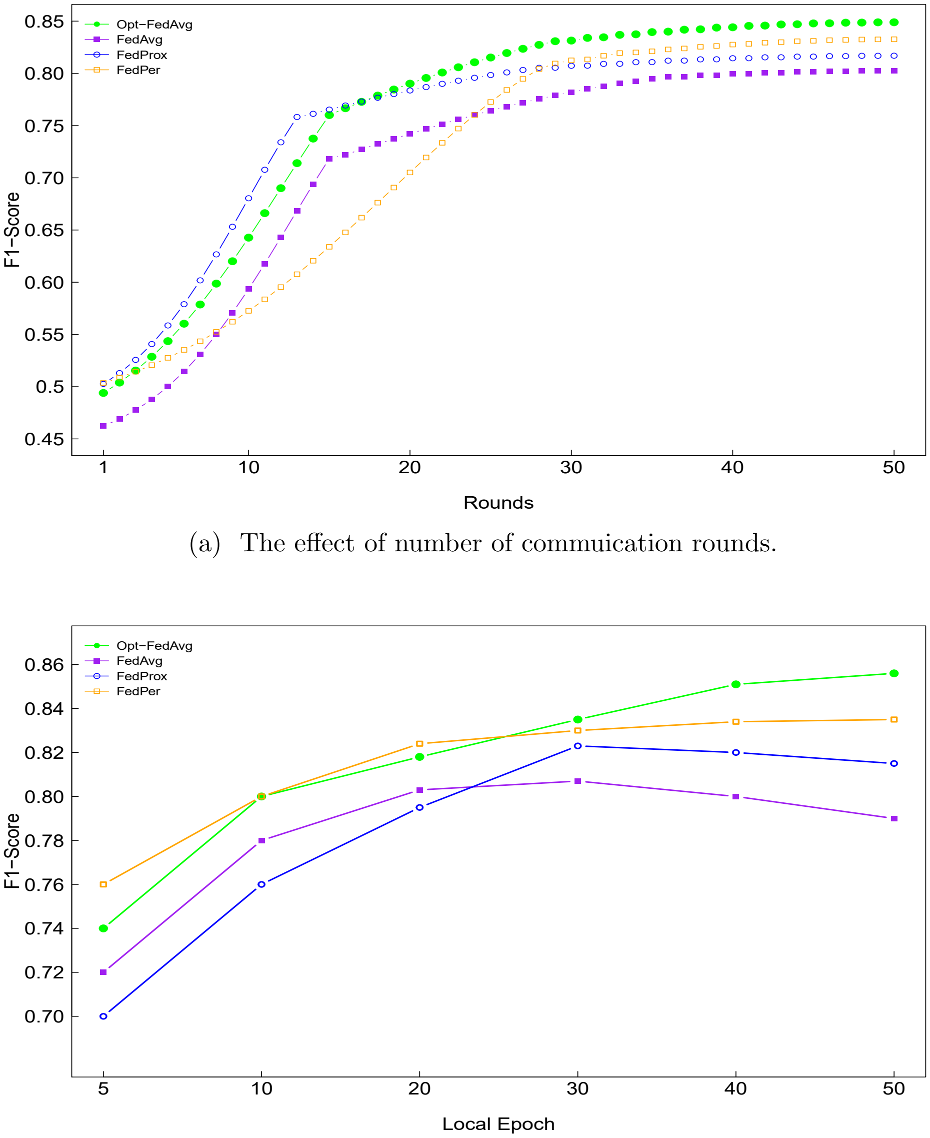
7.4 Summary

We present a novel machine learning approach for an effective and efficient anomaly detection model in such applications like anomaly detection in X-ray images which may require information derived from many spatially-distributed locations. Our method employs a Federated Learning (FL) approach to OCSVM as an anomaly

Table 7.1: *F-score* of various methods.

	Opt-FedAvg	FedProx	FedPer	FedAvg
AB	0.85±0.02	0.82±0.01	0.83±0.03	0.80±0.04
MNIST	0.87±0.03	0.85±0.02	0.82±0.05	0.83±0.06
CIFAR	0.75±0.03	0.74±0.02	0.73±0.05	0.71±0.06

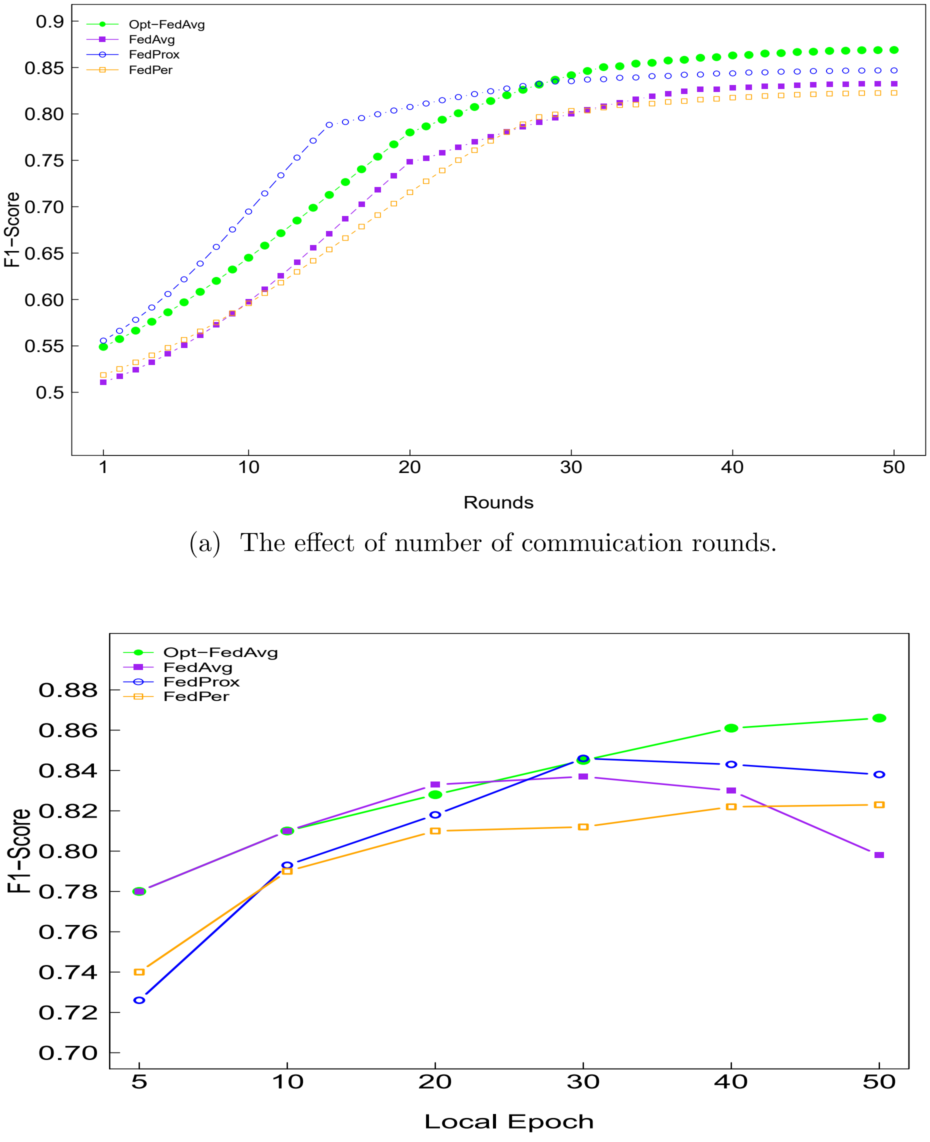
detection model augmented with a novel efficient communication method for coefficient's aggregation. Our experimental evaluation on AB, MNIST and CIFAR datasets showed promising anomaly detection accuracy compared to other state-of-the-art methods. In the "AB" dataset, our Opt-FedAvg method achieved an accuracy of 85%. In the MNIST and CIFAR datasets, our Opt-FedAvg method achieved 87% and 75% anomaly detection accuracy, respectively. The experimental results of these three datasets demonstrated the capability of our FL-based anomaly detection approach with the coefficient's aggregation method to add extra learning experience to each local model using data located on multiple airports without compromising the privacy and security of the collected data.



(a) The effect of number of commuication rounds.

(b) The effect of number of local training epochs.

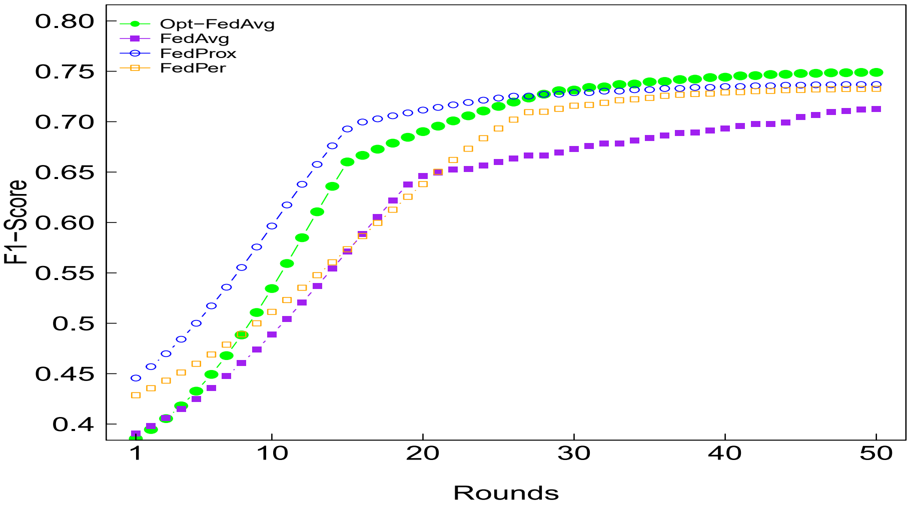
Figure 7.1: Convergence rates of various methods in federated learning applied on AB dataset with five clients.



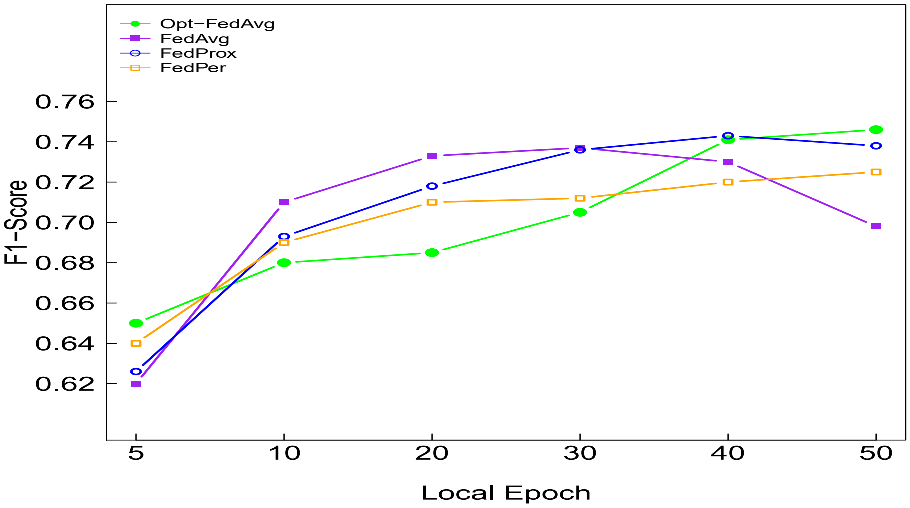
(a) The effect of number of commuication rounds.

(b) The effect of number of local training epochs.

Figure 7.2: Convergence rates of various methods in federated learning applied on MNIST dataset with five clients.



(a) The effect of number of commuication rounds.



(b) The effect of number of local training epochs.

Figure 7.3: Convergence rates of various methods in federated learning applied on CIFAR dataset with five clients.

Chapter 8

Conclusion

Airport security screening is essential to ensure the safety of the aviation industry and passengers travelling by air. Security processes at the airport have undeniable costs for passengers that can make many passengers unsatisfied and can be divided into four types: time delay, indirect financial costs, privacy, and inconvenience. However, with the increased number of passengers.

The goal of this research was to develop essential models: (i) to optimise the security screening process by reducing the average waiting time; (ii) to design security screening area by predicting the number of servers and security officers; (iii) to reduce time and enhance the X-ray security screening system by automating anomaly detection in cabin bags; and (iv) to build a federated learning model from a broad range of data sets located at different locations without data pooling. The main data used in this study was collected from Sydney International Airport.

Four main research questions have been addressed in this thesis: (i) How to optimise the security screening process to reduce passenger's average waiting time; (ii)

How to design the security screening process and forecast the average waiting time based on number of passengers and servers; (iii) How to build an Anomaly detection model in multi-view learning settings of X-ray security screening system; (iv) How to implement federated learning network for anomaly detection in X-ray security screening.

8.1 Contribution 1: Optimizing the Waiting Time for Airport Security Screening using Multiple Queues and Server

Section 4.2 presents the framework is a combination of queueing theory and Lindley process to propose QQT (Queues Queueing Theory) model to optimise the security screening process with multi-servers in parallel to serve different number of passengers during different seasons, such as Christmas, Easter and school holidays, and time of the day, as this strongly influences the number of passengers, in order to improve the average waiting time in airport security areas.

8.2 Contribution 2: Design of airport security screening using queueing theory augmented with particle swarm optimisation

This model proposes a novel method based on queueing theory augmented with particle swarm optimisation (QT-PSO) is presented in section 5.1 to address the second research question. This model is used to predict passenger waiting times in a security screening context. This model consists of multiple servers operating in parallel and takes into consideration the complete scenario such as normal, slow and express lanes. Such an approach has the potential to be a reliable model that is able to assimilate variations in the number of passengers, security officers and security machines on the service time.

8.3 Contribution 3: Anomaly Detection in X-ray Security Imaging

This model proposes a novel tensor-based learning method for anomaly detection in X-ray security screening systems based on tensor analysis augmented with one-class classification model, introduced in section 6.1 to address the third research question. Our method initially performs data fusion of multi-angle scanned images in one tensor structure from where we extract the informative features, and further constructs a one-class support vector machine model using these features to detect anomalies.

8.4 Contribution 4: A Federated Learning Anomaly Detection Approach for X-ray Security Imaging

This model proposes a federated learning (FL) approach for anomaly detection in X-ray security imaging using OCSVM, introduced in section 7.2 to address the fourth research question. This innovative machine learning approach can train a centralized model on data generated and located on multiple airports without compromising the privacy and security of the collected data. Moreover, it does not require transmitting large amount of data which can be a major performance challenge especially for real-time applications. FL approach can enable multiple airports to collaborate on the development of a central anomaly detection model by only sharing the model coefficients of each client/airport model rather than the whole data collected by all participating airports.

8.5 Future Work

The ultimate goal of this thesis is to build a complete model for airport security screening process which includes optimising and designing the security process at airport. Although our QQT and QT-PSO models were able to reduce passengers average waiting, predict passengers average waiting time and determine number of services, future work is required on this research project to handle the case when the number of servers are less than three.

Similarly, our federated learning network for anomaly detection based on tensor analysis and OCSVM was able to successfully detects anomalous objects in X-ray images with an average accuracy of 85%. However, it is important to improve this accuracy and address the case when the data across different airports are not independent and non-identically distributed (Non-IID). Another challenge to address in our future work is to incorporate deep learning methods such as Generative Adversarial Network (GAN) for efficient anomaly detection methods.

Bibliography

[1] Samet Akcay, Amir Atapour-Abarghouei, and Toby P Breckon, *Ganomaly: Semi-supervised anomaly detection via adversarial training*, Asian conference on computer vision, Springer, 2018, pp. 622–637.

[2] Samet Akçay, Amir Atapour-Abarghouei, and Toby P Breckon, *Skip-ganomaly: Skip connected and adversarially trained encoder-decoder anomaly detection*, 2019 International Joint Conference on Neural Networks (IJCNN), IEEE, 2019, pp. 1–8.

[3] Samet Akcay, Mikolaj E Kundegorski, Chris G Willcocks, and Toby P Breckon, *Using deep convolutional neural network architectures for object classification and detection within x-ray baggage security imagery*, IEEE transactions on information forensics and security **13** (2018), no. 9, 2203–2215.

[4] TM Al Muhareb and J Graham-Jones, *Using lean six-sigma in the improvement of service quality at aviation industry: Case study at the departure area in kkia*, International Journal of Social, Behavioral, Educational, Economic, Business and Industrial Engineering **8** (2014), no. 1, 145–151.

[5] S Almazroui, W Wang, and G Zhang, *Imaging technologies in aviation security*, Advances in Image and Video Processing **3** (2015), no. 4, 12–27.

[6] Jinwon An and Sungzoon Cho, *Variational autoencoder based anomaly detection using reconstruction probability*, Special Lecture on IE **2** (2015), no. 1, 1–18.

[7] A Anaissi, A Braytee, and M Naji, *Gaussian kernel parameter optimization in one-class support vector machines*, Proceedings of the International Joint Conference on Neural Networks, 2018.

[8] Ali Anaissi, Nguyen Lu Dang Khoa, Thierry Rakotoarivelo, Mehri Makki Alamdari, and Yang Wang, *Self-advised incremental one-class support vector machines: An application in structural health monitoring*, International Conference on Neural Information Processing, Springer, 2017, pp. 484–496.

[9] Jerone TA Andrews, Nicolas Jaccard, Thomas W Rogers, and Lewis D Griffin, *Representation-learning for anomaly detection in complex x-ray cargo imagery*, Anomaly Detection and Imaging with X-Rays (ADIX) II, vol. 10187, International Society for Optics and Photonics, 2017, p. 101870E.

[10] JTA Andrewsa, N Jaccarda, TW Rogersa, T Tanaya, and LD Griffina, *Anomaly detection for security imaging*, (2017).

[11] Manoj Ghuhan Arivazhagan, Vinay Aggarwal, Aaditya Kumar Singh, and Sunav Choudhary, *Federated learning with personalization layers*, arXiv preprint arXiv:1912.00818 (2019).

[12] Søren Asmussen, *Applied probability and queues*, vol. 51, Springer Science & Business Media, 2008.

[13] Benjamin Avi-Itzhak, Hanoch Levy, and David Raz, *Quantifying fairness in queueing systems: Principles and applications*, Preprint (2004).

[14] Vellara L Lazar Babu, Rajan Batta, and Li Lin, *Passenger grouping under constant threat probability in an airport security system*, European Journal of Operational Research **168** (2006), no. 2, 633–644.

[15] Garrick Blalock, Vrinda Kadiyali, and Daniel H Simon, *The impact of post-9/11 airport security measures on the demand for air travel*, The Journal of Law and Economics **50** (2007), no. 4, 731–755.

[16] Alexandr Borovkov, *Stochastic processes in queueing theory*, vol. 4, Springer Science & Business Media, 2012.

[17] Ali Braytee, Farookh Khadeer Hussain, Ali Anaissi, and Paul J Kennedy, *Abc-sampling for balancing imbalanced datasets based on artificial bee colony algorithm*, 2015 IEEE 14th international conference on machine learning and applications (ICMLA), IEEE, 2015, pp. 594–599.

[18] Rasmus Bro and Henk AL Kiers, *A new efficient method for determining the number of components in parafac models*, Journal of chemometrics **17** (2003), no. 5, 274–286.

[19] Maria Virginia Caccavale, Antonio Iovanella, Carlo Lancia, Guglielmo Lulli, and Benedetto Scoppola, *A model of inbound air traffic: The application to heathrow airport*, Journal of Air Transport Management **34** (2014), 116–122.

[20] Elizabeth Castaneda, Jamie Gonzalez, Shannon Harris, and Joon Kim, *Optimized airport security infrastructure system (oasis)*, 2007 IEEE Systems and Information Engineering Design Symposium, IEEE, 2007, pp. 1–6.

[21] Raghavendra Chalapathy, Aditya Krishna Menon, and Sanjay Chawla, *Anomaly detection using one-class neural networks*, arXiv preprint arXiv:1802.06360 (2018).

[22] Fei Chen, Mi Luo, Zhenhua Dong, Zhenguo Li, and Xiuqiang He, *Federated meta-learning with fast convergence and efficient communication*, arXiv preprint arXiv:1802.07876 (2018).

[23] Robert B Cooper, *Introduction to queueing theory, ch. 4*, Elsevier North Holland Inc **6** (1981), 18–136.

[24] Santanu Das, Ashok N Srivastava, and Aditi Chattopadhyay, *Classification of damage signatures in composite plates using one-class svms*, Aerospace Conference, 2007 IEEE, IEEE, 2007, pp. 1–19.

[25] Yamille Del Valle, Ganesh Kumar Venayagamoorthy, Salman Mohagheghi, Jean-Carlos Hernandez, and Ronald G Harley, *Particle swarm optimization: basic concepts, variants and applications in power systems*, IEEE Transactions on evolutionary computation **12** (2008), no. 2, 171–195.

[26] Yuyang Deng, Mohammad Mahdi Kamani, and Mehrdad Mahdavi, *Adaptive personalized federated learning*, arXiv preprint arXiv:2003.13461 (2020).

[27] Thibaut Durand, Taylor Mordan, Nicolas Thome, and Matthieu Cord, *Wildcat: Weakly supervised learning of deep convnets for image classification, pointwise localization and segmentation*, Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 642–651.

[28] Paul F Evangelista, Mark J Embrechts, and Boleslaw K Szymanski, *Some properties of the gaussian kernel for one class learning*, International Conference on Artificial Neural Networks, Springer, 2007, pp. 269–278.

[29] Rolf Felkel and Dirk Klann, *Comprehensive passenger flow management at frankfurt airport*, Journal of Airport Management **6** (2012), no. 2, 107–124.

[30] H George Frederickson and Todd R LaPorte, *Airport security, high reliability, and the problem of rationality*, Public Administration Review **62** (2002), 33–43.

[31] Yona Falinie A Gaus, Neelanjan Bhowmik, Samet Akçay, Paolo M Guillén-Garcia, Jack W Barker, and Toby P Breckon, *Evaluation of a dual convolutional neural network architecture for object-wise anomaly detection in cluttered*

x-ray security imagery, 2019 International Joint Conference on Neural Networks (IJCNN), IEEE, 2019, pp. 1–8.

[32] Yona Falinie A Gaus, Neelanjan Bhowmik, and Toby P Breckon, *On the use of deep learning for the detection of firearms in x-ray baggage security imagery*, 2019 IEEE International Symposium on Technologies for Homeland Security (HST), IEEE, 2019, pp. 1–7.

[33] David Gillen and William G Morrison, *Aviation security: Costing, pricing, finance and performance*, Journal of Air Transport Management **48** (2015), 1–12.

[34] Ronald R Gilliam, *An application of queueing theory to airport passenger security screening*, Interfaces **9** (1979), no. 4, 117–123.

[35] Konstantina Gkritza, Debbie Niemeier, and Fred Mannering, *Airport security screening and changing passenger satisfaction: An exploratory assessment*, Journal of Air Transport Management **12** (2006), 213–219.

[36] Lewis D Griffin, Matthew Caldwell, Jerone TA Andrews, and Helene Bohler, *"unexpected item in the bagging area": Anomaly detection in x-ray security images*, IEEE Transactions on Information Forensics and Security **14** (2018), no. 6, 1539–1553.

[37] 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 (2021).

[38] Filip Hanzely and Peter Richtárik, *Federated learning of a mixture of global and local models*, arXiv preprint arXiv:2002.05516 (2020).

[39] Andrew Hard, Kanishka Rao, Rajiv Mathews, Swaroop Ramaswamy, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, and Daniel

Ramage, *Federated learning for mobile keyboard prediction*, arXiv preprint arXiv:1811.03604 (2018).

[40] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Keith Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al., *Advances and open problems in federated learning*, arXiv preprint arXiv:1912.04977 (2019).

[41] Sema Kayapınar and Nihal Erginel, *Designing the airport service with fuzzy qfd based on servqual integrated with a fuzzy multi-objective decision model*, Total Quality Management & Business Excellence **30** (2019), no. 13-14, 1429–1448.

[42] Waqar Ahmed Khan, Sai-Ho Chung, Hoi-Lam Ma, Shi Qiang Liu, and Ching Yuen Chan, *A novel self-organizing constructive neural network for estimating aircraft trip fuel consumption*, Transportation Research Part E: Logistics and Transportation Review **132** (2019), 72–96.

[43] Safa Khazai, Saeid Homayouni, Abdolreza Safari, and Barat Mojaradi, *Anomaly detection in hyperspectral images based on an adaptive support vector method*, Geoscience and Remote Sensing Letters, IEEE **8** (2011), no. 4, 646–650.

[44] Alan Avi Kirschenbaum, *The cost of airport security: The passenger dilemma*, Journal of Air Transport Management **30** (2013), 39–45.

[45] Alan Avi Kirschenbaum, Michele Mariani, Coen Van Gulijk, Sharon Lubasz, Carmit Rapaport, and Hinke Andriessen, *Airport security: An ethnographic study*, Journal of air transport management **18** (2012), no. 1, 68–73.

[46] Tamara G Kolda and Brett W Bader, *Tensor decompositions and applications*, SIAM review **51** (2009), no. 3, 455–500.

[47] Saskia Koller, *Assessing x-ray image interpretation competency of airport security screeners*, Ph.D. thesis, 2006.

[48] Saskia M Koller, Colin G Drury, and Adrian Schwaninger, *Change of search time and non-search time in x-ray baggage screening due to training*, Ergonomics **52** (2009), no. 6, 644–656.

[49] Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon, *Federated learning: Strategies for improving communication efficiency*, arXiv preprint arXiv:1610.05492 (2016).

[50] Alex Krizhevsky, Geoffrey Hinton, et al., *Learning multiple layers of features from tiny images*, (2009).

[51] Yann LeCun, *Mnist handwritten digit database, yann lecun, corinna cortes and chris burges*, 2013.

[52] Adrian J Lee and Sheldon H Jacobson, *The impact of aviation checkpoint queues on optimizing security screening effectiveness*, Reliability Engineering & System Safety **96** (2011), no. 8, 900–911.

[53] Kelly Leone and Rongfang Rachel Liu, *The key design parameters of checked baggage security screening systems in airports*, Journal of Air Transport Management **11** (2005), no. 2, 69–78.

[54] Kelly Leone and Rongfang Rachel Liu, *Improving airport security screening checkpoint operations in the us via paced system design*, Journal of Air Transport Management **17** (2011), no. 2, 62–67.

[55] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith, *Federated optimization in heterogeneous networks*, arXiv preprint arXiv:1812.06127 (2018).

[56] Yongli Li, Xin Gao, Zhiwei Xu, and Xuanrui Zhou, *Network-based queuing model for simulating passenger throughput at an airport security checkpoint*, Journal of Air Transport Management **66** (2018), 13–24.

[57] Yuhua Li and Liam Maguire, *Selecting critical patterns based on local geometrical and statistical information*, IEEE Transactions on Pattern Analysis and Machine Intelligence **33** (2011), no. 6, 1189–1201.

[58] Clara V Marin, Colin G Drury, Rajan Batta, and Li Lin, *Human factors contributes to queuing theory: Parkinson's law and security screening*, Proceedings of the Human Factors and Ergonomics Society Annual Meeting, vol. 51, SAGE Publications Sage CA: Los Angeles, CA, 2007, pp. 602–606.

[59] Roberto Rendeiro Martín-Cejas, *Tourism service quality begins at the airport*, Tourism Management **27** (2006), no. 5, 874–877.

[60] Ivan Martinovic and Martin Strohmeier, *On the security of the automatic dependent surveillance- broadcast protocol*, IEEE Communications Surveys & Tutorials **17** (2015), no. 2.

[61] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas, *Communication-efficient learning of deep networks from decentralized data*, Artificial Intelligence and Statistics, PMLR, 2017, pp. 1273–1282.

[62] Jefferson Ryan Medel and Andreas Savakis, *Anomaly detection in video using predictive convolutional long short-term memory networks*, arXiv preprint arXiv:1612.00390 (2016).

[63] Ariel Merari, *Attacks on civil aviation: Trends and lessons*, Terrorism and Political Violence **10** (1998), no. 3, 9–26.

[64] Caijing Miao, Lingxi Xie, Fang Wan, Chi Su, Hongye Liu, Jianbin Jiao, and Qixiang Ye, *Sixray: A large-scale security inspection x-ray benchmark for prohibited item discovery in overlapping images*, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 2119–2128.

[65] Mohamad Naji, Sherif Abdelhalim, Ahmed Al-Ani, and Hiyam Al-Kilidar, *Airport security screening process: a review*, CICTP 2017: Transportation Reform and Change—Equity, Inclusiveness, Sharing, and Innovation (2018), 3978–3988.

[66] Mohamad Naji, Ali Anaissi, Ali Braytee, and Madhu Goyal, *Anomaly detection in x-ray security imaging: a tensor-based learning approach*, 2021 International Joint Conference on Neural Networks (IJCNN), IEEE, 2021, pp. 1–8.

[67] Xiaofeng Nie, Gautam Parab, Rajan Batta, and Li Lin, *Simulation-based selectee lane queueing design for passenger checkpoint screening*, European Journal of Operational Research **219** (2012), no. 1, 146–155.

[68] Alexander G Nikolaev, Sheldon H Jacobson, and Laura A McLay, *A sequential stochastic security system design problem for aviation security*, Transportation Science **41** (2007), no. 2, 182–194.

[69] Sun Olapiriyakul and Sanchoy Das, *Design and analysis of a two-stage security screening and inspection system*, Journal of Air Transport Management **13** (2007), no. 2, 67–74.

[70] Evangelos E Papalexakis, Christos Faloutsos, and Nicholas D Sidiropoulos, *Tensors for data mining and data fusion: Models, applications, and scalable algorithms*, ACM Transactions on Intelligent Systems and Technology (TIST) **8** (2016), no. 2, 16.

[71] David R Pendergraft, Craig V Robertson, and Shelly Shrader, *Simulation of an airport passenger security system*, Proceedings of the 2004 Winter Simulation Conference, 2004., vol. 1, IEEE, 2004.

[72] Vesna Popovic, Ben Kraal, and Phil Kirk, *Passenger experience in an airport: an activity-centred approach*, International Association of Societies of Design Research 2009 Proceedings: Design Rigor and Relevance (2009), 1–11.

[73] Thomas W Rogers, Nicolas Jaccard, and Lewis D Griffin, *A deep learning framework for the automated inspection of complex dual-energy x-ray cargo imagery*, Anomaly Detection and Imaging with X-Rays (ADIX) II, vol. 10187, International Society for Optics and Photonics, 2017, p. 101870L.

[74] Bernhard Schölkopf, John C Platt, John Shawe-Taylor, Alex J Smola, and Robert C Williamson, *Estimating the support of a high-dimensional distribution*, Neural computation **13** (2001), no. 7, 1443–1471.

[75] Karen Simonyan and Andrew Zisserman, *Very deep convolutional networks for large-scale image recognition*, arXiv preprint arXiv:1409.1556 (2014).

[76] Jacek Skorupski and Piotr Uchroński, *Fuzzy inference system for the efficiency assessment of hold baggage security control at the airport*, Safety science **79** (2015), 314–323.

[77] Jacek Skorupski and Piotr Uchronski, *Managing the process of passenger security control at an airport using the fuzzy inference system*, Expert Systems with Applications **54** (2016), 284–293.

[78] Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet Talwalkar, *Federated multi-task learning*, arXiv preprint arXiv:1705.10467 (2017).

[79] Yanik Sterchi, Nicole Hättenschwiler, Stefan Michel, and Adrian Schwaninger, *Relevance of visual inspection strategy and knowledge about everyday objects for x-ray baggage screening*, 2017 International Carnahan Conference on Security Technology (ICCST), IEEE, 2017, pp. 1–6.

[80] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi, *Inception-v4, inception-resnet and the impact of residual connections on learning*, Thirty-first AAAI conference on artificial intelligence, 2017.

[81] Canh T Dinh, Nguyen Tran, and Tuan Dung Nguyen, *Personalized federated learning with moreau envelopes*, Advances in Neural Information Processing Systems **33** (2020).

[82] Diana Turcsany, Andre Mouton, and Toby P Breckon, *Improving feature-based object recognition for x-ray baggage security screening using primed visualwords*, 2013 IEEE International conference on industrial technology (ICIT), IEEE, 2013, pp. 1140–1145.

[83] Xiaofang Wang and Jun Zhuang, *Balancing congestion and security in the presence of strategic applicants with private information*, European Journal of Operational Research **212** (2011), no. 1, 100–111.

[84] Paul Pao-Yen Wu, Jegar Pitchforth, and Kerrie Mengersen, *A hybrid queue-based bayesian network framework for passenger facilitation modelling*, Transportation Research Part C: Emerging Technologies **46** (2014), 247–260.

[85] Jinfeng Yang, Zihao Zhao, Haigang Zhang, and Yihua Shi, *Data augmentation for x-ray prohibited item images using generative adversarial networks*, IEEE Access **7** (2019), 28894–28902.

[86] Houssam Zenati, Chuan Sheng Foo, Bruno Lecouat, Gaurav Manek, and Vijay Ramaseshan Chandrasekhar, *Efficient gan-based anomaly detection*, arXiv preprint arXiv:1802.06222 (2018).

[87] Jianguo Zhang, Kai-Kuang Ma, Meng-Hwa Er, and Vincent Chong, *Tumor segmentation from magnetic resonance imaging by learning via one-class support vector machine*, International Workshop on Advanced Image Technology (IWAIT'04), 2004, pp. 207–211.

[88] Shuo Zhou, Nguyen Xuan Vinh, James Bailey, Yunzhe Jia, and Ian Davidson, *Accelerating online cp decompositions for higher order tensors*, Proceedings of

the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2016, pp. 1375–1384.