Detection of Alcohol Intoxication in Airports using Pose Estimation

Final Report - Group Design Project

Group 1

Danuka Theja Wickramasinghe (S371881), Deepak Kovaichelvan (S336570), Priyanka Sarkar (S364601), Saurabh Bharadwaj (S348574), Xavier Ikuenobe (C099409), Yousuf Shaikh (S371550)

Date: 11th April 2022

Contents

1. Introduction

Air travel is often a stressful time for passengers, as they are likely to experience a range of stressors before boarding (e.g., tiring commutes, security, boarding procedures etc) and while in-flight (e.g., cramped seating, physical discomfort etc). When combined with individual factors such as anxiety or intoxication, these stressors manifest in the form of disruptive behaviour and endanger safety – a phenomenon known as Disruptive Airline Passenger Behaviour (DAPB) [1].

DAPB encompasses incidents that occur in-flight as well as in airports. Some of the commonly reported DAPBs include: refusing to comply with safety instructions, verbal/physical/sexual abuse, assault and damage to aircraft or airport property [1].

DAPB has increased rapidly in recent decades and is only recently showing signs of plateauing. DAPB is known to affect all surrounding people, including cabin staff, airport employees (e.g., poor job satisfaction, compromised health and safety) and fellow passengers (e.g., poor travel experience and increased vulnerability). DAPB is also expensive for airlines. For example, the re-routing of a transatlantic flight to Belfast is reported to cost an estimated £500,000, when accounted for the fuel dump and mandatory 24-hour delay [1].

Alcohol intoxication is one of the major contributors to DAPB. International Air Transport Association (IATA) data indicates that 31% of incidents in 2016 involved alcohol intoxication. A YouGov survey carried out in 2018 for the Institute for Alcohol Studies shows that this is a significant issue for passengers. The survey found that 60% of British adults had encountered drunk passengers and the majority of them believe that there is a serious problem with alcohol intoxication during air travel [1], [2].

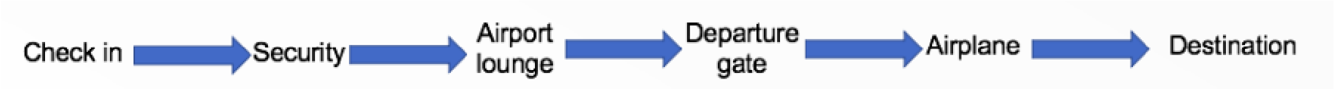


Figure 1: Typical journey through an airport

The drinking may begin on the flight itself, in the airport (bars, restaurants, duty-free) or even before arriving at the airport. Since the Licencing Act 2003 does not apply beyond security in airports in England and Wales, the control on the sale and consumption of alcohol in airports is relatively relaxed. Pre-flight drinking presents a major challenge for cabin crew as it makes it difficult to monitor their alcohol consumption due to the short time spent at each stage shown in Figure 1. The EU regulation requires that airlines shall take all reasonable measures to ensure that no person enters or is in an aircraft under the influence of alcohol to the extent that it endangers the safety of the aircraft or its occupants [2].

The alcohol intoxication issue is currently dealt with without the use of AI or technology. These methods are subject to human bias and low quality, as they use passenger profiling based on predictors such as race, ethnicity, age, appearance and gender. Therefore, there is a strong business case for a system to assist with the detection of alcohol intoxication at the airports, as the best means to prevent alcohol-related DAPB in-flight is to prevent the passengers from boarding in the first place.

The focus area of this report is to design and develop an AI system that can better identify alcohol-intoxication related DAPB using pose estimation. Pose estimation is a technique that uses the key points in the human body to make predictions. It is expected that pose estimation will be effective at reducing bias as the human pose is more neutral compared to the currently used predictors. The system will play an assisting role to the Police and airport authorities to better detect and mitigate such behaviours.

This project is intended to be a proof of concept. The intended beneficiaries/stakeholders of the system include the airline personnel, security personnel, restaurants, shops and passengers (including the disruptive passengers). The use cases for each of the stakeholders are described in section 3.

2. Literature Review

2.1. Current Approach in Airports

Literature shows that airports have taken two types of approaches to detect and monitor intoxicated passengers. Both of the approaches rely on humans for detection.

The Greater Manchester Police Department has launched a yellow card warning system to curb alcohol related problems at the Manchester International Airport. If a passenger is suspected of having consumed too much alcohol, they are handed a football-style yellow card by a Police officer, which outlines the legal restrictions on alcohol consumption. Information on the passenger's suspected condition is handed to the respective airline, who can make an informed decision on whether the passenger is fit to fly. In the most serious cases, the passengers who do not drink responsibly can be arrested or jailed for up to two years [3].

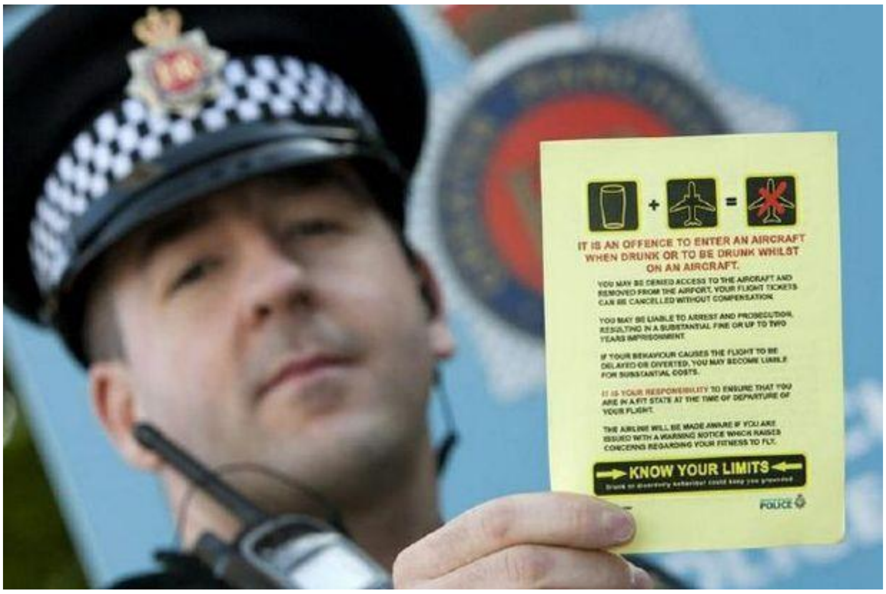

IT IS AN OFFENCE TO ENTER AN AIRCRAFT WHEN DRUNK OR TO BE DRUNK WHILST ON AN AIRCRAFT.
YOU MAY BE DENIED ACCESS TO THE AIRCRAFT AND REMOVED FROM THE AIRPORT. YOUR FLIGHT TICKETS CAN BE CANCELLED WITHOUT COMPENSATION.
YOU MAY BE LIABLE TO ARREST AND PROSECUTION, RESULTING IN A SUBSTANTIAL FINE OR UP TO TWO YEARS IMPRISONMENT.
IF YOUR BEHAVIOUR CAUSES THE FLIGHT TO BE DELAYED OR DIVERTED, YOU MAY BECOME LIABLE FOR SUBSTANTIAL COSTS.
IT IS YOUR RESPONSIBILITY TO ENSURE THAT YOU ARE IN A FIT STATE AT THE TIME OF DEPARTURE OF YOUR FLIGHT.
THE AIRLINE WILL BE MADE AWARE IF YOU ARE ISSUED WITH A WARNING NOTICE WHICH RAISES CONCERNS REGARDING YOUR FITNESS TO FLY.
KNOW YOUR LIMITS
POLICE

Figure 2: Yellow card warning issued at the Manchester International Airport [4]

The second approach, also deployed at the Manchester International Airport relies on airport staff to notify the police of potentially disruptive passengers. The staff working in airport bars, shops and gates use a live reporting system to provide alerts on potential troublesome passengers. If they suspect passengers of drunk or aggressive behaviour, they can subtly scan a QR code with a smartphone to create an incident report, which is shared across the airport network. The reporting system is used to prevent unruly passengers from boarding planes [5].

2.2. Computer Vision Based Approaches

Different algorithms have been explored in the existing literature to detect alcohol intoxication. However, there is no evidence of these systems deployed in airports. Airports present specific challenges such as occlusion due to crowded environments, which need to be considered for successful implementation.

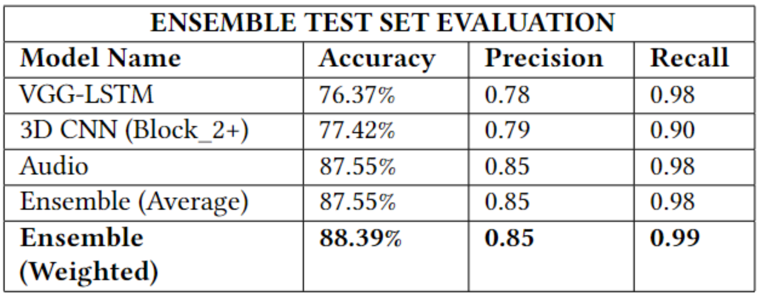
2.2.1. Facial Recognition

V Mehta et al [6] propose an alcohol detection system using facial recognition. The detection is performed using facial appearance, based on facial texture and its variation over time. Therefore, video input of the facial features is required for this approach. The underlying rationale for using facial recognition is that there is evidence of changes in facial expressions and eye movement patterns under the influence of alcohol. Drowsiness and fatigue are also visible after alcohol consumption [6].

Deep-learning based architectures were implemented. The first model uses a CNN-RNN architecture to extract spatio-temporal features from videos. The CNN layers extract facial features, while the RNN (LSTM) layers encode temporal change across frames. The second model employs 3D CNN to learn from the spatio-temporal changes in the videos [6].

Additionally, the sample's audio is processed using an Audio DNN and Audio LSTM. Finally, a decision ensemble approach is used to integrate the two modalities (video and audio) to improve performance. The results of the model are shown in Table 1, which shows that the ensemble is able to make reasonably good predictions [6].

Table 1: Results from the individual modalities and the ensemble [6]



ENSEMBLE TEST SET EVALUATION			
Model Name	**Accuracy**	**Precision**	**Recall**
VGG-LSTM	76.37%	0.78	0.98
3D CNN (Block_2+)	77.42%	0.79	0.90
Audio	87.55%	0.85	0.98
Ensemble (Average)	87.55%	0.85	0.98
Ensemble (Weighted)	**88.39%**	**0.85**	**0.99**

However, the model was trained and tested on a relatively small dataset. Further research also revealed the limitations in facial recognition algorithms. The algorithms boast high classification accuracy, but the results are not universal across demographic groups. This also applies to those developed by major technology companies, including Microsoft, IBM and Amazon. This has been attributed to the imbalances in the training and test datasets, which highlights that facial recognition is not neutral in nature unless careful consideration is made about the datasets used [7].

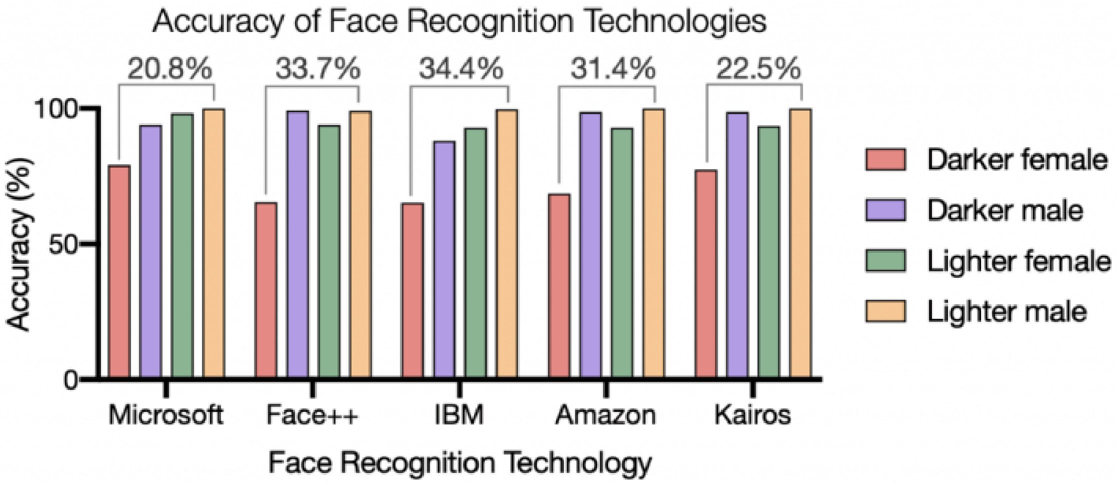


Figure 3: Variation in facial recognition accuracy across demographic groups [7]

2.2.2. Rule-Based Detection

Making a person walk in a straight line is one of the field impairments tests that are performed to determine whether a driver of a car is suffering from diminished psychomotor abilities. This test however requires a complex set of instructions that the subject would perform while walking in a straight line, ensuring at each step that the heel of the forward foot touches the toe of the back foot. This test also requires a certified human agent as an evaluator [8]. It was decided to replicate such a test, but without being intrusive or requiring a human agent as an evaluator, thus minimising the effect of racial, social and ethnic biases on decision making.

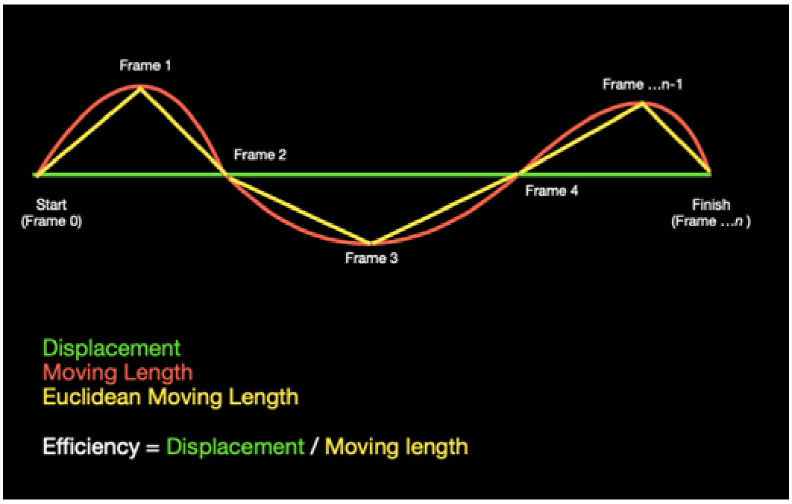


Figure 4: Calculating motion efficiency frame-by-frame

After further review of literature, a technique proposed by J Lee et al [9] was discovered to measure the efficiency and deviation of a person's trajectory while walking in a straight line. The metrics of interest in this paper were 1. Motion efficiency and 2. Motion deviation.

Motion efficiency compares the displacement from start to finish to the actual distance covered by a person's trajectory as shown in Figure 4 and the below equations [9].

$$Motion\ Efficiency = \frac{Euclidean\ displacement\ from\ start\ to\ finish}{Actual\ trajectory\ distance}$$

$$Motion\ Deviation = 1 - Motion\ Efficiency$$

The equations above demonstrate the zigzaggedness of a person walking in a straight line. A person with more psychomotor control would have a motion efficiency closer to 1 (motion deviation closer to zero), whereas a person with less psychomotor control would have a motion efficiency significantly less than 1 (motion deviation significantly greater than zero) [9].

One of the challenges with the above approach is the possibility of classifying a sober person as a drunk person, in case the sober person is zig-zagging for reasons other than intoxication e.g., avoiding an obstacle/people. The above issue was addressed by the paper with a cumulative averaging technique applied to deviation scores calculated at regular frame intervals.

The algorithm for cumulative averaging is as shown in Figure 5(a), (b) and its visualisation in Figure 5(c), where it can be noticed that the maximum and minimum deviation thresholds are used to decide whether to add deviation scores from subsequent frame samples or to reduce the motion deviation scores by a given constant in cases where the deviation is below the minimum deviation threshold at the time of observation. The deviation score is always initialised at a value of zero and the score is hardcoded to not go below zero, in case the person is walking within the minimum tolerances all the time [9]. This allows the technique to take a holistic approach towards motion deviation calculations along the whole trajectory. Moreover, it prevents misclassification of a sober person as drunk because of chance deviations while walking.

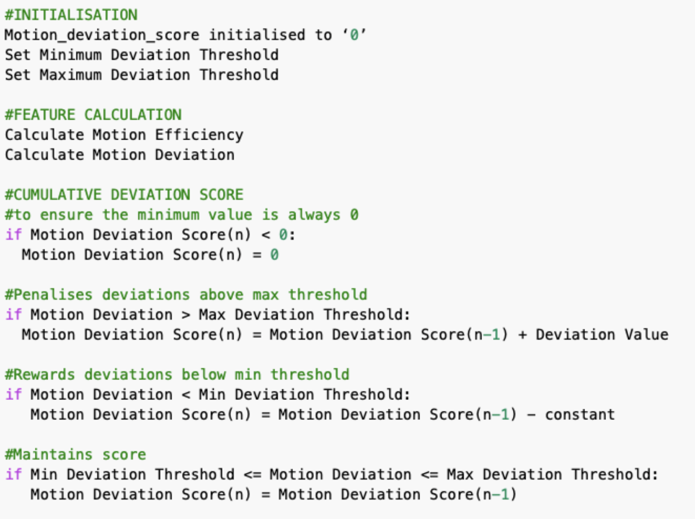
(a)

$$s_m(t) = \begin{cases} s_m(t-1) + \lambda_m(t) & \text{if } \lambda_m(t) \geq T_m^h \\ s_m(t-1) - \delta_m & \text{if } \lambda_m(t) < T_m^l \\ s_m(t-1) & \text{otherwise} \end{cases},$$

Where:

S_m: *Cumulative Motion Deviation*
t : *Time*
λ_m: Motion Deviation at a Particular Frame
T_m^h: *Upper Threshold for Motion Deviation* (0.2)
T_m^l: *Lower Threshold for Motion Deviation* (0.1)
δ_m: *Predefined Constant for Motion Deviation Reduction*

(b)



```
#INITIALISATION
Motion_deviation_score initialised to '0'
Set Minimum Deviation Threshold
Set Maximum Deviation Threshold

#FEATURE CALCULATION
Calculate Motion Efficiency
Calculate Motion Deviation

#CUMULATIVE DEVIATION SCORE
#to ensure the minimum value is always 0
if Motion Deviation Score(n) < 0:
  Motion Deviation Score(n) = 0

#Penalises deviations above max threshold
if Motion Deviation > Max Deviation Threshold:
  Motion Deviation Score(n) = Motion Deviation Score(n-1) + Deviation Value

#Rewards deviations below min threshold
if Motion Deviation < Min Deviation Threshold:
   Motion Deviation Score(n) = Motion Deviation Score(n-1) - constant

#Maintains score
if Min Deviation Threshold <= Motion Deviation <= Max Deviation Threshold:
   Motion Deviation Score(n) = Motion Deviation Score(n-1)
```

(c)

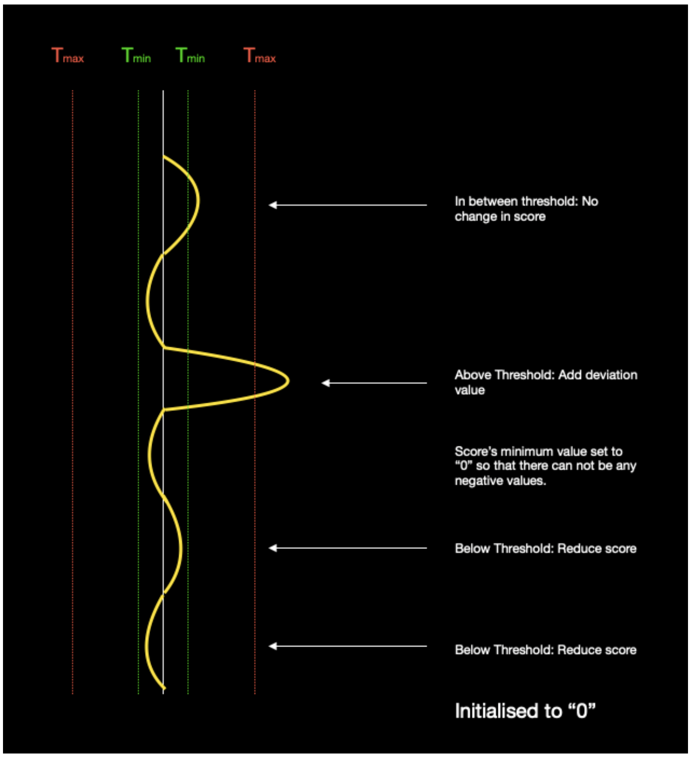

Tmax
Tmin
Tmin
Tmax
In between threshold: No change in score
Above Threshold: Add deviation value
Score's minimum value set to "0" so that there can not be any negative values.
Below Threshold: Reduce score
Below Threshold: Reduce score
Initialised to "0"

Figure 5: (a) Mathematical formulation for cumulative motion deviation [9]*, (b) Pseudo Code for cumulative addition of deviation scores, (c) Visual representation of cumulative addition for motion deviation*

Figure 6 shows the distributions of motion efficienies for normal and drunk people, with the red dashed lines showing how the motion deviaton thresholds in Figure 5(a) were determined.

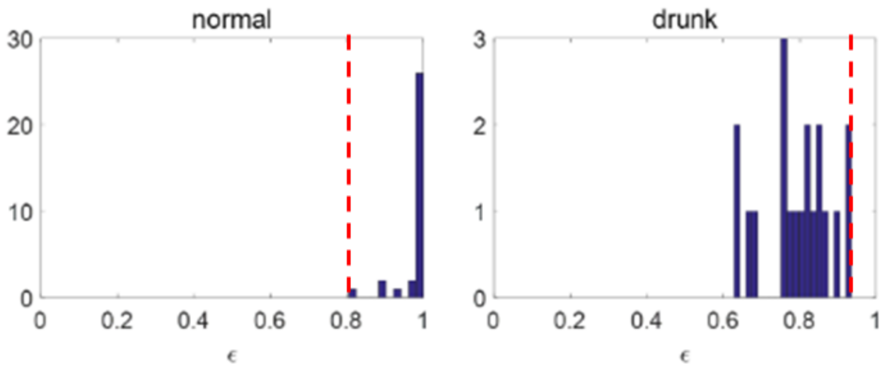
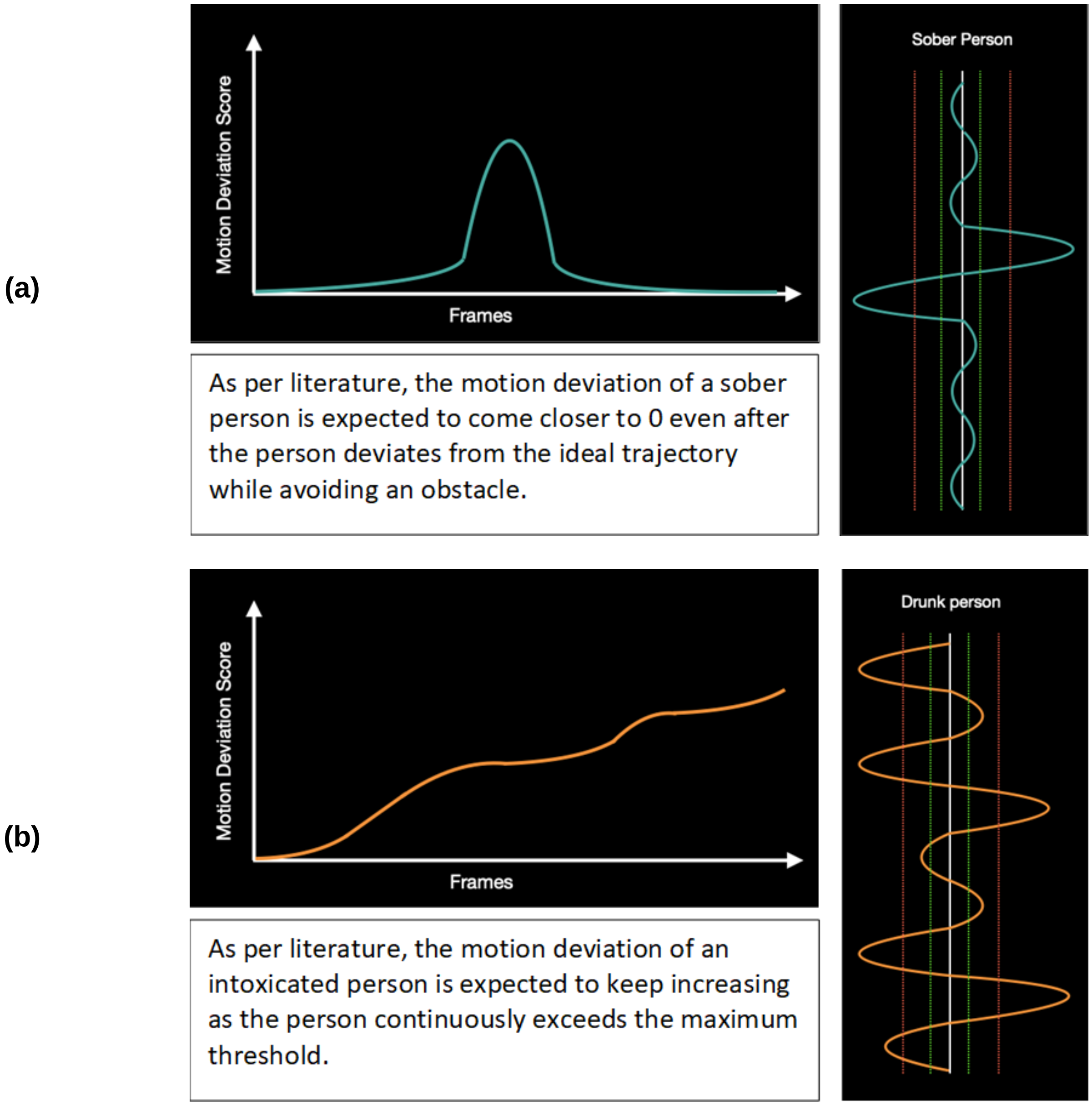


Figure 6: Distribution of motion efficiency of normal and drunk people [9]

The expected motion deviation profiles of both, a sober person with a one-off deviation as well as a drunk person can be modelled to result in Figure 7(a) and Figure 7(b).



As per literature, the motion deviation of a sober person is expected to come closer to 0 even after the person deviates from the ideal trajectory while avoiding an obstacle.

As per literature, the motion deviation of an intoxicated person is expected to keep increasing as the person continuously exceeds the maximum threshold.

Figure 7: (a) Motion deviation profile and score for a sober person avoiding an obstacle, (b) Motion deviation profile and score for an intoxicated person

2.2.3. Auto Encoder

Another approach to detect intoxicated behaviour researched by O. Temuroglu et al proposes the use of pose estimation methods to deduce the human key points. These human key point representations are converted to skeleton representations [10].

Auto encoders are a popular type of neural network which are used for feature extraction. The underlying principle behind autoencoders is that they learn to represent the training data in lower dimensions through the process of compression and decompression of the data. During the detection/inference phase, the autoencoder network will not be able to represent data that are not similar to the data used to train on. This will result in the network returning a high reconstruction error for the dissimilar data. Figure 8 illustrates the process flow for the proposed method [10].

This characteristic has been exploited in this approach to detect intoxicated behaviour. The autoencoder is trained only on normal walking poses from videos, therefore intoxicated walking will be separated based on the high reconstruction errors.

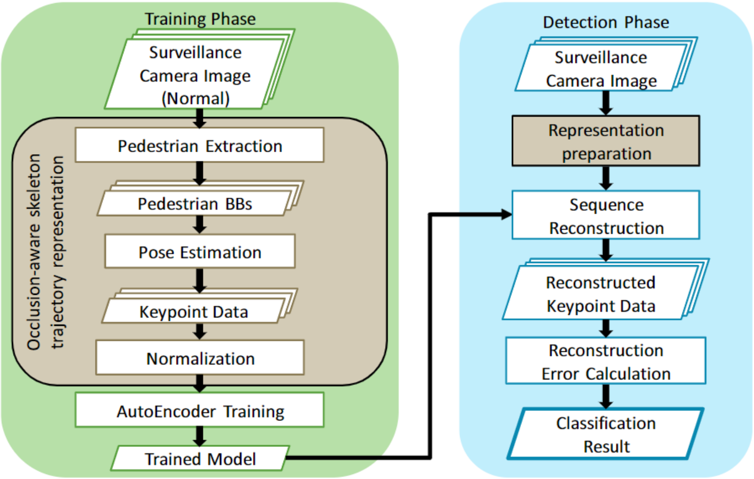

Training Phase
Surveillance Camera Image (Normal)
Occlusion-aware skeleton trajectory representation
Pedestrian Extraction
Pedestrian BBs
Pose Estimation
Keypoint Data
Normalization
AutoEncoder Training
Trained Model
Detection Phase
Surveillance Camera Image
Representation preparation
Sequence Reconstruction
Reconstructed Keypoint Data
Reconstruction Error Calculation
Classification Result

Figure 8: Process flow of the method proposed by O. Temuroglu et al [10]

The autoencoder is a relatively simple network comprised of a few dense layers as the pose skeletons are compact and contain all relevant information. An encoder network and a decoder network are used sequentially to acquire the reconstruction error. When mapping the x, y values of the key points and reconstructing them, a custom loss function shown below is used to classify the sequence as normal or intoxicated, by using a predefined reconstruction threshold [10].

$$L\left(\hat{P}', P'\right) = \frac{1}{2NJ}\sum_{n=1}^{N}\sum_{j=1}^{J} c_n^{\prime j}\left(\left(\hat{x}_n{}^{\prime j} - x_n{}^{\prime j}\right)^2 + \left(\hat{y}_n^{\prime j} - y_n^{\prime j}\right)^2\right)$$

Where:
$c_n{}'^{j}$ is the normalised confidence level of the j^{th} key point in the n^{th} skeleton,
$x_n{}'^{j}$ and $y_n{}'^{j}$ are the normalised 2D coordinates and
$\hat{x}_n{}'^{j}$ and $\hat{y}_n{}'^{j}$ are the normalised 2D reconstructed coordinates

The results of this approach are summarised in Table 2. The classes represent the number of joints unable to be detected out of a batch of frames with 750 total joints, mostly due to occlusion. This shows that the method is capable of achieving a high accuracy on intoxicated persons, although there is no information provided on normal persons [10].

Table 2: Accuracy of the autoencoder on grouped intoxicated data [10]

Dataset	Proposed method
Easy (0-22)	0.986
Moderate (23-86)	0.996
Hard (87+)	0.894

One of the noticeable limitations of this proposed method is that the detection is made using individual frames rather than a sequence of frames. Using a sequence of frames can better understand and detect the behaviour of the subjects and would help improve the accuracy of the method. This is one of the key contributions of this report.

3. System Overview

This section aims to provide a high-level overview of the project and its implementation. Further sections in this report build upon each element of the system in detail.

3.1. High-Level System Design

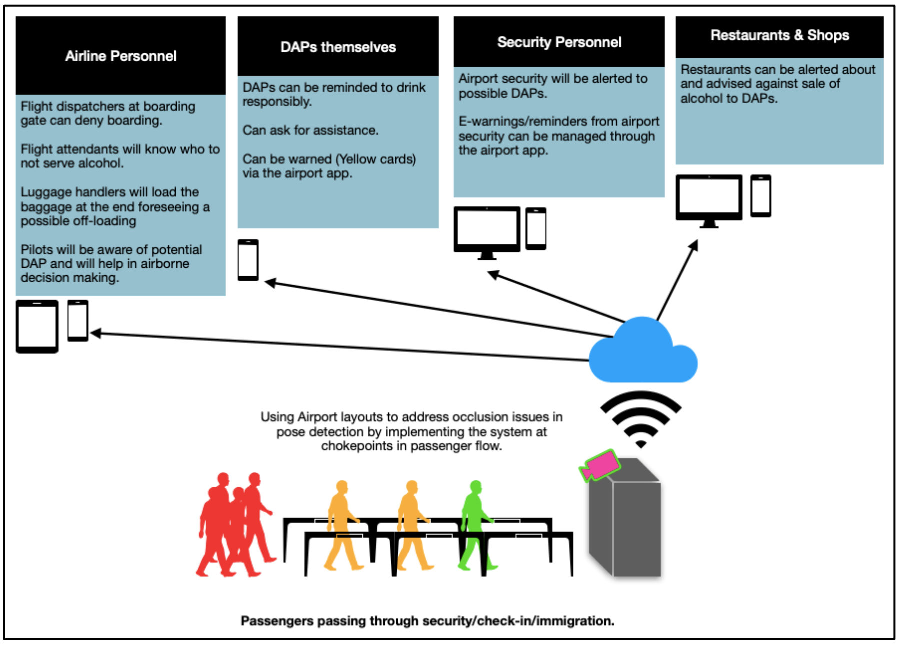
3.1.1. Smart Airports

Increased passenger volumes demand efficient airport operations by ensuring smooth flow of passengers [11]. Currently, the journey through a conventional airport is interrupted by multiple checkpoints for check-in, security, immigration etc.

Smart airports of the future will simplify and streamline passenger flows through an airport. Airlines such as Emirates are already testing out alternative forms of check-in/immigration using identification such as iris scans and facial recognition, therefore it can be foreseen that such technology might involve computer vision to enhance current capabilities [12]. Passenger experience through smart airports of the future can have just the passenger walking through multiple checkpoints without check-in stands and intrusive security checks.

Considering the above, the proposal is for a method of increasing airport security and efficiency by detecting possible DAPs using pose estimation.

3.1.2. System Implementation in Smart Airports



Airline Personnel

Flight dispatchers at boarding gate can deny boarding.

Flight attendants will know who to not serve alcohol.

Luggage handlers will load the baggage at the end foreseeing a possible off-loading

Pilots will be aware of potential DAP and will help in airborne decision making.

DAPs themselves

DAPs can be reminded to drink responsibly.

Can ask for assistance.

Can be warned (Yellow cards) via the airport app.

Security Personnel

Airport security will be alerted to possible DAPs.

E-warnings/reminders from airport security can be managed through the airport app.

Restaurants & Shops

Restaurants can be alerted about and advised against sale of alcohol to DAPs.

Using Airport layouts to address occlusion issues in pose detection by implementing the system at chokepoints in passenger flow.

Passengers passing through security/check-in/immigration.

Figure 9: High-level implementation strategy

The system uses human pose estimation to detect intoxicated passengers as potential DAPs. Similar to smart homes, which have connected devices, the connected smart airport infrastructure makes it possible to integrate multiple systems via IoT and cloud-based technology. The aim is to design a system that can be integrated into this environment. Since there will be a continued need for security checks, the intention is to integrate the proposed system into the existing infrastructure that will continue to serve its purpose in smart airports. It is foreseen that the proposed system will have the ability to provide connectivity and access options to all stakeholders interested in identifying potential DAPs.

Passenger flow chokepoints at airports are prime examples of where the proposed system can be implemented (such as that shown in Figure 9) at airports. They provide a minimal occlusion environment and predictable human behaviour. These chokepoints are therefore a crucial factor in the success of the proposed system and the models have been designed accordingly.

3.1.3. Stakeholders and Use Cases

Although this project is intended to be a proof of concept, Table 3 shows the identified potential stakeholders and use cases applicable to this system upon deployment in an actual airport.

Table 3: Stakeholders and use cases of the Alcohol Intoxication Detection System

Stakeholder		Use Case	Stakeholder Risk	Requirement
Airline Personnel	Flight dispatchers	Deny confirmed DAP from boarding airline.	Flight disruption or delays caused by DAP before, during or after boarding airplane.	To receive notifications of potential DAP boarding airline and DAP engagement guideline messages.
	Flight attendants	Know who not to serve alcohol.	Flight disruption or delays caused by DAP after boarding airplane.	
	Luggage handlers	Load baggage of possible DAP last so offloading baggage can be fast if they are denied boarding.	Flight disruption or delays caused by time to locate and offload luggage of DAP that has been denied boarding.	
	Pilots	To be aware of possible DAP in order not be surprised by DAPB.	Flight disruption or delays caused by DAP before, during or after boarding airplane.	
DAPs		A reminder for DAP to drink responsibly.	Denial of flight boarding or Denial of alcohol on flights	Restaurants can be alerted about possible DAPs and can then be advised to not serve alcohol.
		DAPs can ask for assistance.		
		DAPs can be warned. (Yellow card system)		
Security Personnel		To be alerted of possible DAPB.	Flight disruption or delays caused by DAP before, during or after boarding airplane.	To receive potential DAP notifications and engagement guideline messages
		E-warnings / reminders can be used to allow security personnel to be present in	Potential DAP notifications DAP management advice messages	

	areas that are relatively short-staffed.		
Restaurants & Shops	Restaurants can be alerted about possible DAPs and can then be advised to not serve alcohol.	Increase in DAP alcohol intoxication level and exhibition of disruptive behaviour.	To receive notifications of potential DAP at the airport and DAP engagement advice messages.

3.1.4. System Requirements

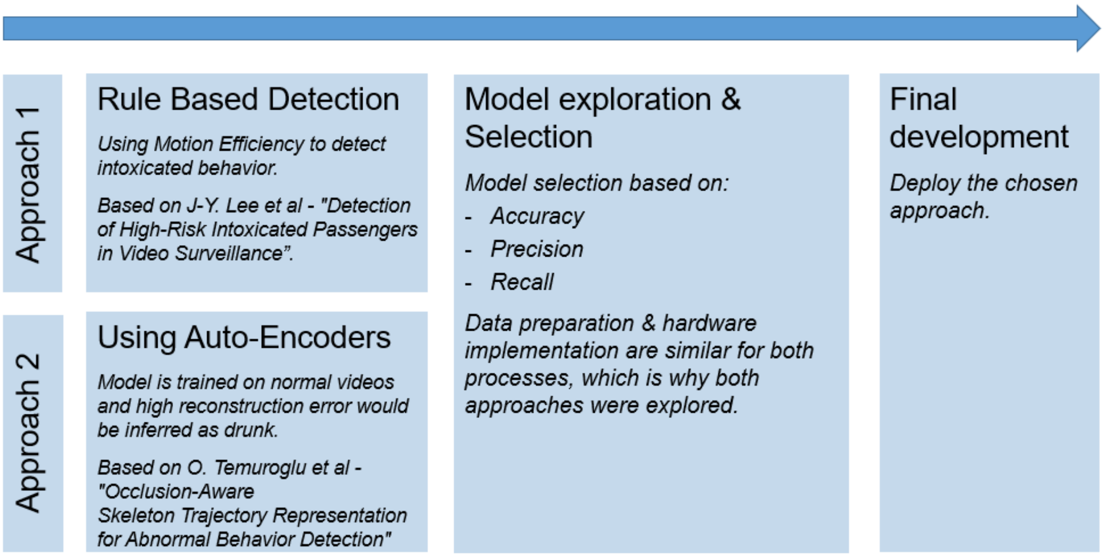
Based on the stakeholders and use cases identified, the system requirements were determined as shown in Table 4 in an attempt to best serve the stakeholders.

Table 4: System Requirements

Requirement	Category	Value
Detect Behavior	Functional Requirement	Alcohol intoxication
Detection Method	Functional Requirement	Pose estimation
Latency	Functional Requirement	Real-time
Precision	Functional Requirement	0.8
Recall	Functional Requirement	0.7
Frame rate	Constraint	> 10 FPS
Occlusion handling	MoE	Maximise
System Cost	Constraint	£1,000
Environment	Environmental Requirement	Airport (Entryways, Security Checkpoints)
Crowd Density	Environmental Requirement	Low level (10/frame)
Cyber Security	Non-Functional Requirement	Integrate with existing protocols
Data Privacy	Non-Functional Requirement	GDPR compliant
Data Management	Non-Functional Requirement	AWS
Cameras	Functional Requirement	(Minimum resolution: 1080p Minimum Illumination: 0 Lux)

3.2. Algorithm Development and Approaches

The team decided to develop two approaches to solve the problem, as shown in Figure 10. The reason for selecting two approaches was to have redundancies in the case of the lack of effectiveness of one of the techniques. Upon careful consideration, the auto encoder approach was chosen to be developed for final implementation. The model selection is discussed further in section 7.



Approach 1

Rule Based Detection

Using Motion Efficiency to detect intoxicated behavior.

Based on J-Y. Lee et al - "Detection of High-Risk Intoxicated Passengers in Video Surveillance".

Approach 2

Using Auto-Encoders

Model is trained on normal videos and high reconstruction error would be inferred as drunk.

Based on O. Temuroglu et al - "Occlusion-Aware Skeleton Trajectory Representation for Abnormal Behavior Detection"

Model exploration & Selection

Model selection based on:

- *Accuracy*
- *Precision*
- *Recall*

Data preparation & hardware implementation are similar for both processes, which is why both approaches were explored.

Final development

Deploy the chosen approach.

Figure 10: Parallel development of two algorithms

Parallel development of two approaches did not utilise excess resources in terms of time and financial budget as both approaches were catered by similar equipment, data and pre-processing pipeline. The parallel streams of development, metrics for final model selection are shown in Figure 10. While developing both models, it was decided to minimise false positives as wrongful accusations could worsen disruptive behaviour among passengers. The system is explained in more detail in the forthcoming chapters of this report.

4. Data

Data, being one of the most important aspects of this system also proved to be one of the biggest challenges addressed during the course of development.

4.1. Two Approaches, Same Data for Different Uses

As mentioned in earlier sections, it was decided to explore two approaches towards model development and then select the one which works best. Since both models used the extraction of human pose key-points from the same deployment environment, hardware and infrastructure, the same data could be used to test and train both models. At a glance, two types of datasets were required:

1. Normal walking behavior
2. Drunk walking behavior

Though the data was the same, it was used differently while being applied to both approaches, as shown in Table 5.

Table 5: Different usage of data for both approaches

Data Type	Normal Walking Videos	Drunk Walking Videos
Approach 1 – Rule Based	Used as testing data	Used as testing data
Approach 2 – Autoencoder	Used as training & test data	Used as testing data

Usage of data for Approach 1 – Rule Based
For the rule-based approach, the model did not have to be trained as it measured the deviation of a person's actual trajectory from the intended trajectory. Therefore, the approach only required testing data as the rule-based algorithm was making absolute calculations. Normal walking data was used alongside the drunk walking data for making test predictions. This is summarised in Figure 11.

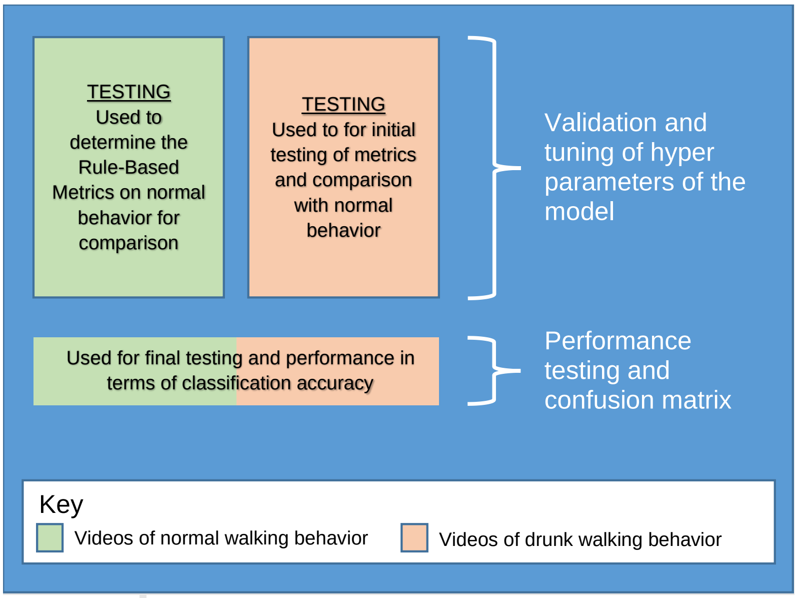

TESTING
Used to determine the Rule-Based Metrics on normal behavior for comparison
TESTING
Used to for initial testing of metrics and comparison with normal behavior
Validation and tuning of hyper parameters of the model
Used for final testing and performance in terms of classification accuracy
Performance testing and confusion matrix
Key
🟩 Videos of normal walking behavior
🟧 Videos of drunk walking behavior

Figure 11: Usage of data for the Rule Based Approach (Approach 1)

Usage of data for Approach 2 – Autoencoder

For the autoencoder approach, a train and test split was used on the normal walking data. The autoencoder model was trained on the normal walking training data to reconstruct during inference stage. Once the weights were trained, the drunk walking videos were used for validation and the reconstruction error was measured. Since the auto-encoder was used to detect the anomalies, the methodology did not require the use of drunk walking for model training. This is summarised in Figure 12.

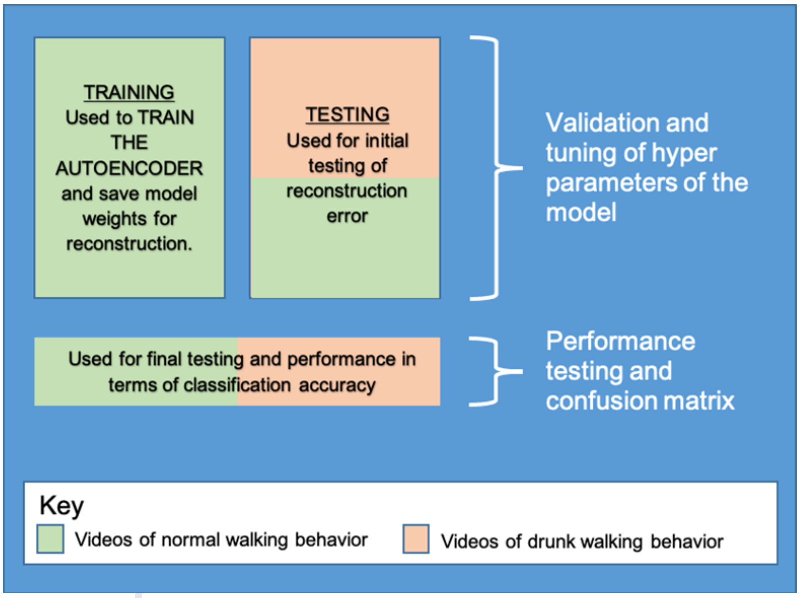

TRAINING
Used to TRAIN THE AUTOENCODER and save model weights for reconstruction.
TESTING
Used for initial testing of reconstruction error
Validation and tuning of hyper parameters of the model
Used for final testing and performance in terms of classification accuracy
Performance testing and confusion matrix
Key
Videos of normal walking behavior
Videos of drunk walking behavior

Figure 12: Usage of data for the Autoencoder approach (Approach 2)

4.2. Building the Dataset

Requirements

The closer the dataset is to the real-world environment, the better and more reliable the performance of both the models would be. The actual/proposed deployment environment consists of flow chokepoints at airports such as security checks, where passengers need to pass through one at a time. In our proposed system design the camera is located at the top of the security checkpoint (as shown in Figure 13), therefore datasets had to be located which provide the same camera angle for training and testing the models.

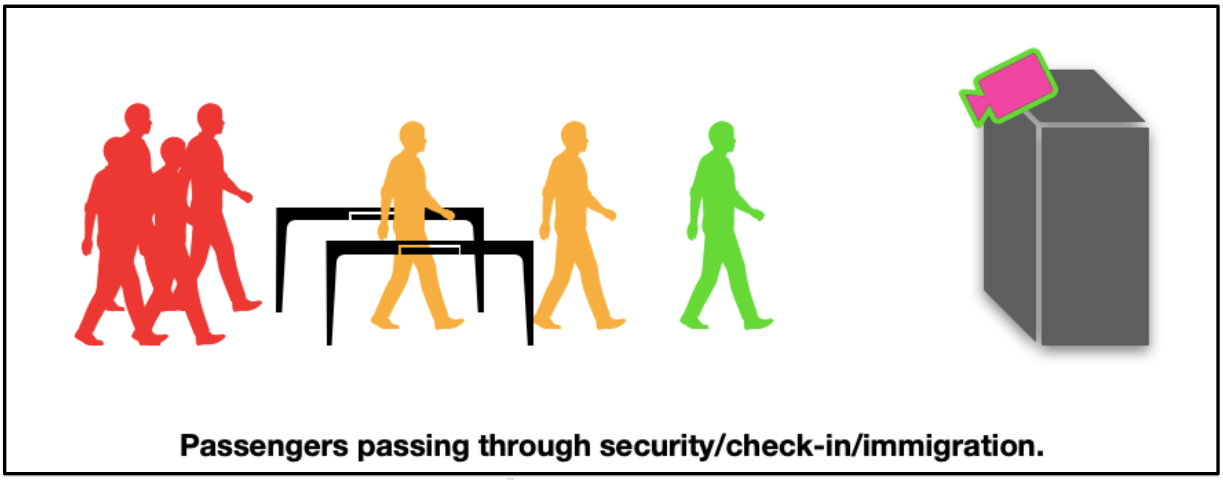


Figure 13: Example of camera location

Availability and Data Gathering

No existing dataset found for drunk behavior, because of which YouTube, iStock libraries were explored. Still, most intoxicated behavior videos were of people who were too drunk to have been able to walk through the airport to the aircraft. Most videos had camera angles inconsistent with requirements mentioned above along with unstable frames and blurry output. Still a few drunk videos were extracted from YouTube.

Videos which could be considered as reliable data should have had persons clearly visible to extract key points, have similar camera location and suitable lighting to enable feature detection. For the normal walking behavior, the only suitable dataset found online was the Chokepoint Dataset [13], shown in Figure 14.

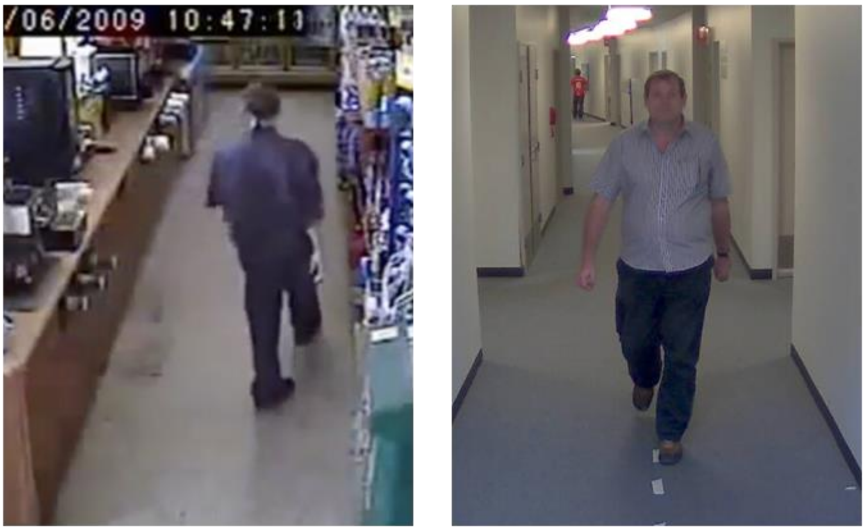


Figure 14: Videos gathered from online resources. (YouTube – Left, Chokepoint Dataset - Right)

To improve the dataset, more data was collected in a controlled environment during an experiment conducted at the CSA (discussed in section 7). The dataset generated through that resulted in videos of both normal and intoxicated walking behavior. The screenshots of the videos have been added in Figure 15 with the consent of the participants.

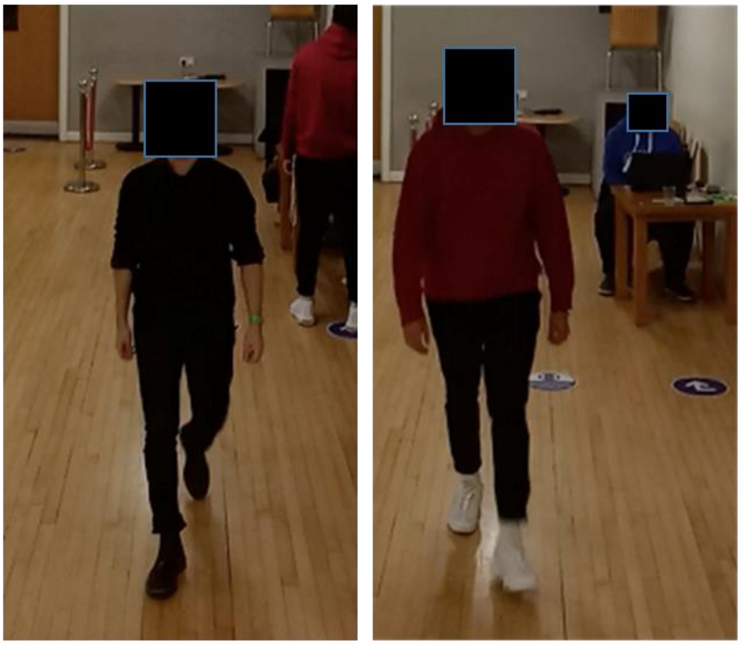


Figure 15: Self-generated dataset (Normal – Left, Drunk – Left)

Overall, 149 videos were gathered from the CSA experiment, including both normal and drunk walking videos; some videos had to be deleted due to lighting issues. Breathalyser readings, number and type of drinks were used to make the classification of who was to be considered drunk and who was to be considered normal. Overall, the videos captured walking behavior from 96 different people. Summary of the data collected from the experiment is shown in Table 6.

Table 6: Summary of data collected from the CSA dataset

Dataset Type	Number of Videos
Normal walking	121
Drunk walking	10
Total	131

5. ML Module Design

Two approaches were selected for development of the ML modules for the project as discussed in Figure 10. Both of the methods use pose estimation as the foundation, with the main difference being one of the approaches is rule based and applies a physics-based approach. The second approach uses an autoencoder to detect intoxicated behaviour.

5.1. Rule-Based Approach

5.1.1. Model Architecture

Based on the approach presented in [9], a rule-based computer vision model was developed. The architecture of the model is as illustrated in Figure 21.

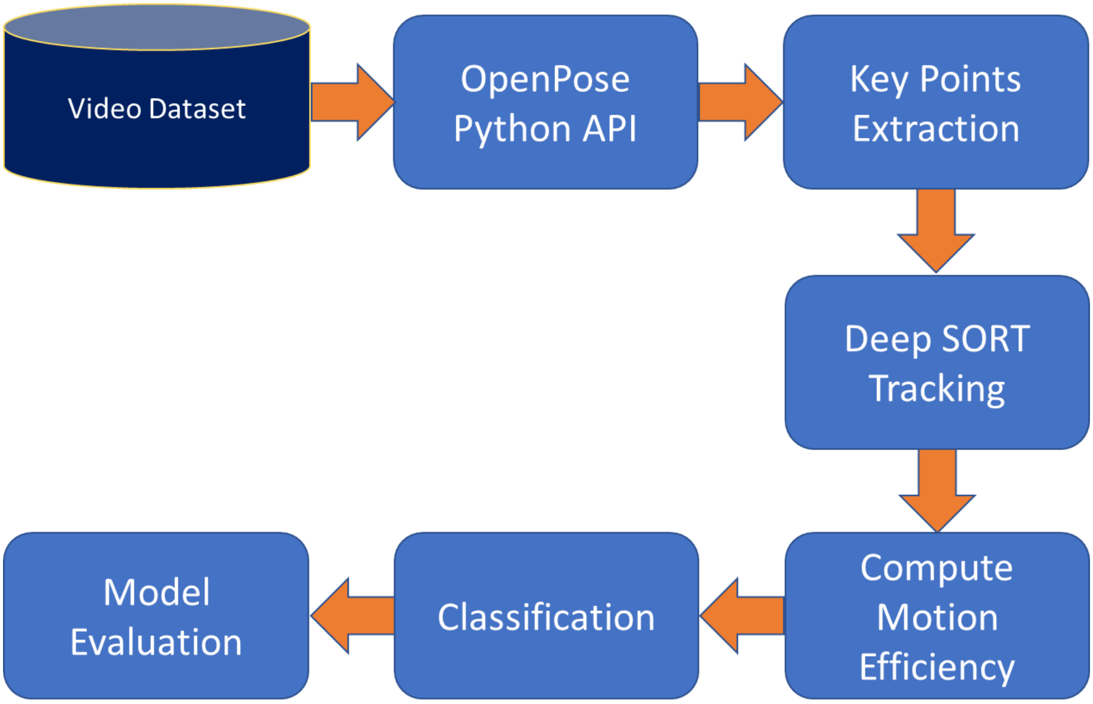


Figure 16: Architecture of rule-based detection model

The model follows the below process flow:

1. The video which is being evaluated is converted into frames using the OpenCV package on Python.

2. OpenPose is a popular open-source Human Pose Estimation library in C++, originally developed at Carnegie Mellon University. OpenPose Python API was used, which allows the import of the package and video data in the form of frames are be sent and retrieved using code [14].

   OpenPose follows a bottom-up approach to pose-estimation. This means that the key points are first detected for each body part of the person and the connection to form human poses happens at the next step. The pipeline can be described at a high-level as shown below and further illustrated in Figure 17 [15]:

a) The approach takes the entire image as the input as passes it into 2 branched CNNs.
b) The first CNN predicts the 2D confidence maps of body part locations.
c) In parallel, the second CNN predicts the part affinity field (PAF). This is a 2D vector of the confidence measure of the association for each pair of body pair associations.
d) Finally, the confidence maps and PAFs are parsed by greedy inference to get a list of human poses for each person in the image.

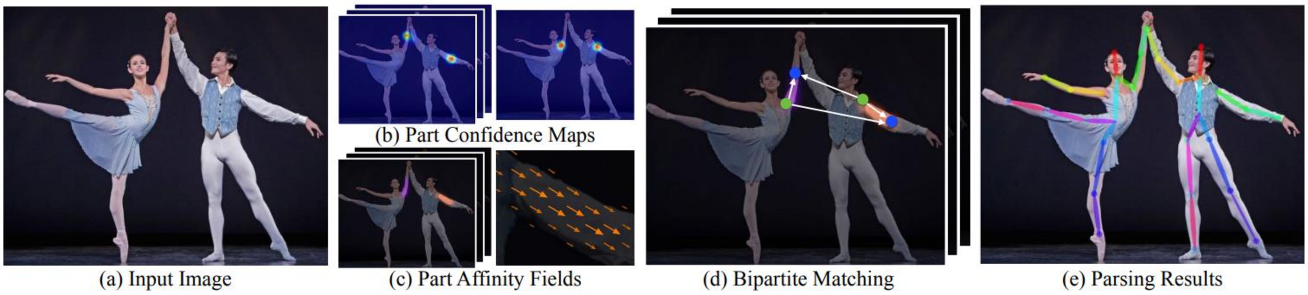

(a) Input Image
(b) Part Confidence Maps
(c) Part Affinity Fields
(d) Bipartite Matching
(e) Parsing Results

Figure 17: OpenPose pipeline [15]

3. OpenPose returns 25 (with indexing 0 to 24) distinct key points of the human body in a dictionary format with their x, y coordinates and detection confidence levels. This data is available for all the frames in the video. These key points form the basis of the rule-based algorithm [14].

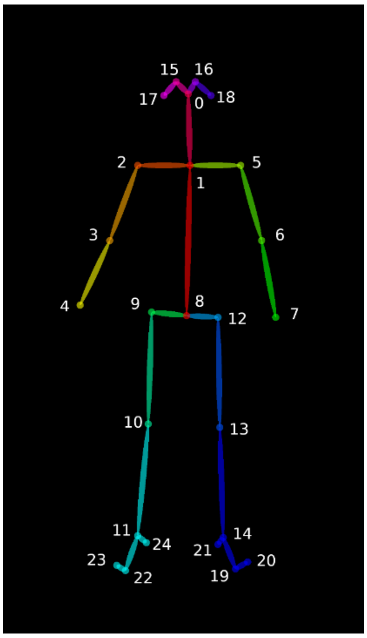


Figure 18: Pose output format [14]

4. Having detected the human body key points, the next challenge is to track the humans. To do so, it is important to be able to distinguish the people in the videos, as there can be multiple persons in a frame. Therefore, to track the humans and associate a unique ID, the human pose estimations are used to estimate the region of interest (ROI). The ROI is calculated using the key point coordinates to find the x, y, width and height [16].

$$ROI = (x, y, width, height)$$

Simple Online and Real-Time Tracking with a deep association metric, also known as DeepSORT is one of the most widely used open-source tracking algorithms. The region of interest where OpenPose has predicted human key points is presented to the DeepSORT algorithm.

DeepSORT's predecessor, SORT used a Kalman filter for motion prediction and frame-by-frame data association. However, this still results in identity switches as it relies on the overlap measured in intersection over union (IoU) of predicted and actual bounding boxes for the association. To remedy this, a pre-trained convolutional neural network (CNN) is incorporated into the DeepSORT algorithm, which is trained to discriminate pedestrians on a large-scale pedestrian dataset, therefore it is able to re-identify even if the person goes out of the frame and comes back after a short time and handle occlusions. Association is based on feature similarity, which is used to create the tracker objects for uniquely identifying humans in the frames [17].

5. The next step is the computation of motion efficiency for each person using the key points and tracker objects that provide a unique ID. The motion efficiency is computed by tracking the left ankle key point (24) to trace the trajectory covered by the person of interest.

   The cumulative averaging method described in the literature review from [9] was followed to compute the motion deviation. The method initialises the motion deviation to zero and accumulates the motion deviation for deviated frames (computing at a frame interval of every 5 frames), but decreases the scores for the normal frame. This ensures that one-off deviations are not classified as drunk.

6. The motion deviation threshold was set to 19 through an experimental process to achieve optimal classification results. It is important to note that this parameter is expected to change based on the camera angle, because the trajectory perceived by the camera is sensitive to the distance between the object and the camera focal length. Therefore, tuning would be required for use in a different environment.

   The results of the classification are presented using a bar and text next to each drunk person on the video as shown in Figure 19. The results are discussed in further detail in section 7 of the report.

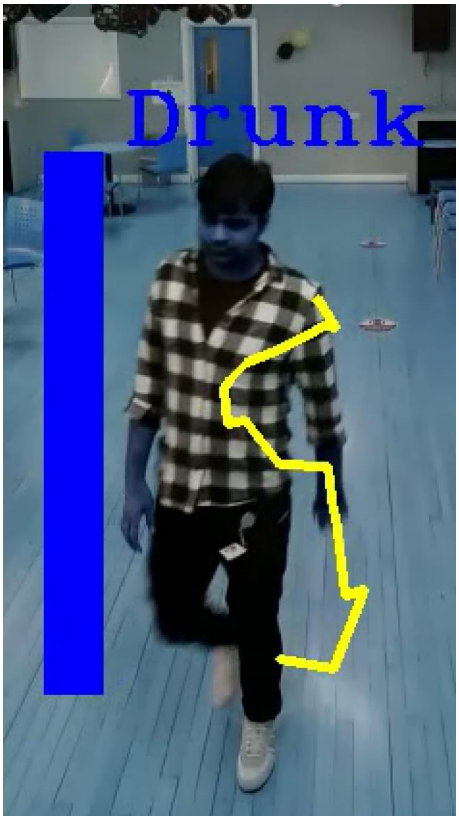


Figure 19: Screenshot of the classification displayed on the video

5.1.2. Tracking 3D Motion in 2D Frames

A person walking towards a camera is a 3D motion with height, width and depth. However, when using pose estimation, the parameters used were pixel values of key point locations in terms of two dimensions of height and width only (Figure 19).

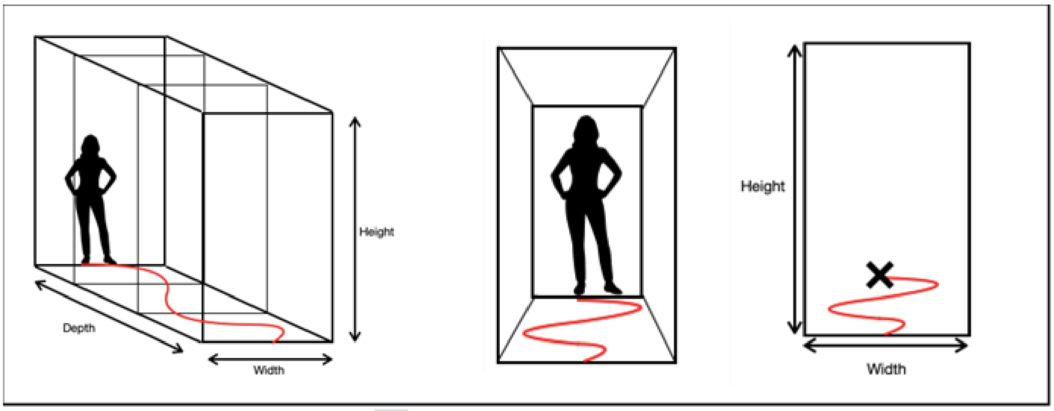


Figure 19: 2D representation of 3D walking

Given the tracking of the key point of interest one may think that the best way to track the motion of a person in two dimensions would be to opt for a top-down view, which would provide the horizontal plane for tracking (Figure 20). However, if used this technique would prevent the usage of pose estimation as it would not present the proper posture required for pose estimation.

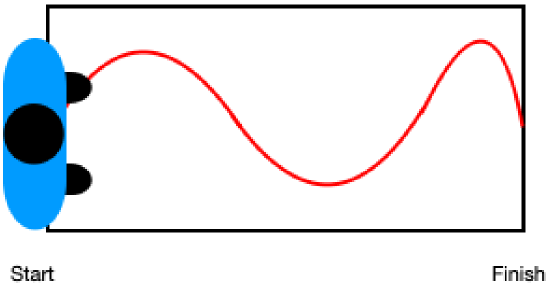


Figure 20: Top-Down view of walking trajectory

It was therefore decided to use as much camera tilt as possible to ensure maximum representation of the walking trajectory and therefore minimise the effects of 3D projection onto a 2D frame (Figure 21). It can be seen in Figure 21 that increasing the camera tilt has the potential to maintain feature extraction requirements and at the same time provide a less compressed trajectory as opposed to the videos captured without any camera tilt. Since only information on the vertical axis was being compressed, it is recommended that all videos used should have the same camera tilt to ensure that the motion efficiency ratios can be uniformly compared.

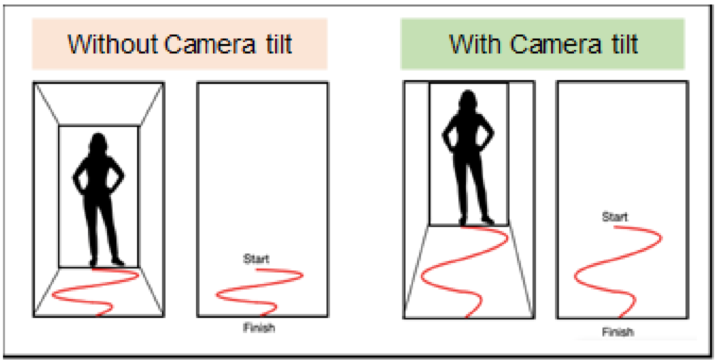


Figure 21: Effects of camera tilt on 3D projection of walking trajectory

5.2. Auto Encoder Approach

5.2.1. Limitations of Existing Approach and Proposed AI Innovation

As discussed in the literature review, the current autoencoder approach in [10] uses individual frames to train the autoencoder without considering the sequence of the frames in the videos captured. Analysing the sequence of frames is a more powerful method of predicting the behaviour compared to individual frames. This was achieved through the use of a custom loss function.

The original loss function used in [10] is shown below:

$$L\left(\hat{P}', P'\right) = \frac{1}{2NJ}\sum_{n=1}^{N}\sum_{j=1}^{J} c_n'^j \left(\left(\hat{x}_n{}'^j - x_n{}'^j\right)^2 + \left(\hat{y}_n'^j - y_n'^j\right)^2\right)$$

The proposed AI innovation is to introduce the time sequence of the frames to predict the behaviours of the persons. This was done through the use of an LSTM deep learning model, which contains the time dimensions. The proposed loss function in the literature cannot be used, therefore mean squared error was used as the loss function to train the model.

5.2.2. Data Processing

For the LSTM, two different datasets were used for training. First, the ChokePoint data set was available on the internet, and the second dataset was created from the experiment carried out at CSA. The two data sets need two different pre-processing methods. The data is saved as frames in JPG format for the ChokePoint dataset. The CSA data set was contained in MP4 videos.

The ChokePoint data set only contains one person in the frames and it is made from continuous frames of one video. So, a for-loop was created to run an inference with MoveNet to each frame to take the body key points and take 90 frames at a time and create a list. As the model contains LSTM layers, the input needs to have a batch, a timestamp and number of features.

The CSA dataset contains different videos for each participant, so the data needs to be processed differently. For this, two for-loops were used, one to take one video at a time, take all the frames from it and run MoveNet inference for each frame, and store the key points to a list. After that, it is combined with the ChokePoint data set key point list and converted into a NumPy array. The NumPy array was split using indexing to create a validation and train set in a 20:80 ratio.

5.2.3. Model Architecture

The MSE loss function was incorporated into the model architecture by making changes to the original architecture in [10]. The original architecture consisted of only a few dense layers. However, since the proposed approach will make predictions based on time series, long short-term memory (LSTM) layers were incorporated to handle the time dimension.

LSTM network was selected over RNN due to its ability to hold information in memory for a long period of time. This is possible as they use special units which have memory cells, which let them learn longer-term dependencies i.e., problems where the desired output depends on inputs that are presented in far apart times in the past [18].

LSTMs also deal with the vanishing and exploding gradient problems better in the learning process using the input and forget gates, which provide better control over the gradient flow and preserve the long-range dependencies. The initial number of frames used by the LSTM was 150. This equates to 5 seconds (at 30 FPS). However, this resulted in the exploding gradient issue. To overcome this issue, the number of frames was reduced to 90 and the issue was solved. This would only be possible with LSTM because theoretically, RNN and Gated Recurrent Units (GRU) cannot train with longer sequence than LSTMs.

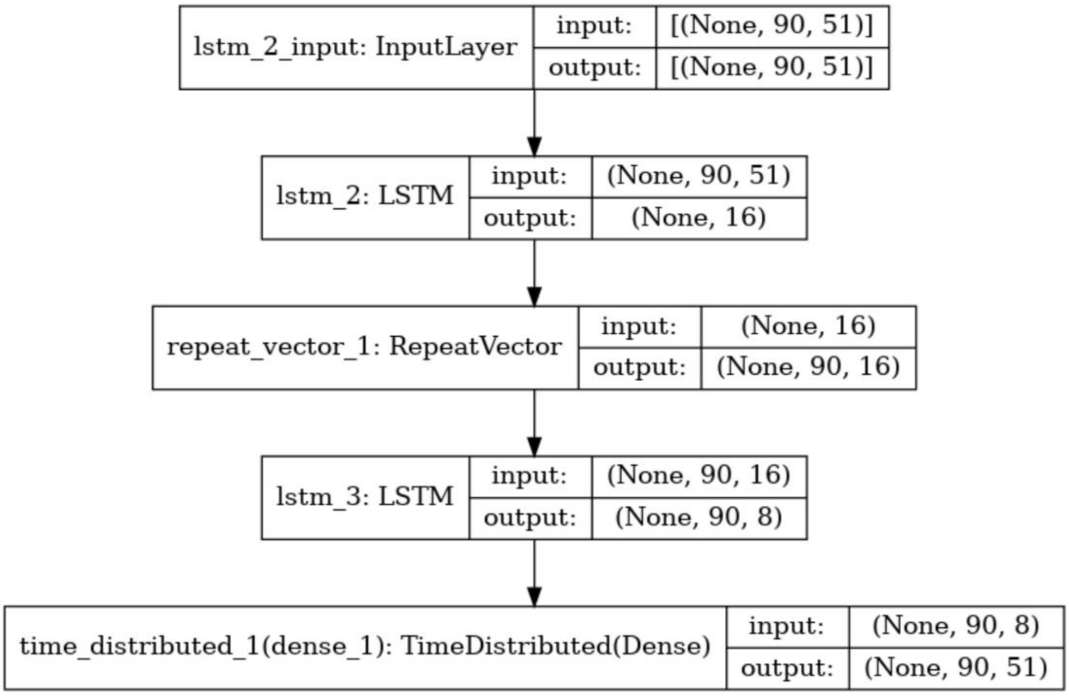


Figure 20: Architecture of the autoencoder model

The Autoencoder network was constructed using the TensorFlow machine learning library on Python as shown below:

1. The first layer is the input layer, which contains the 17 key points from the MoveNet pose estimation model, which includes their x and y coordinates along with the confidence levels. This results in 51 (17 times 3) features from 90 frames at a time.

2. The features from the input layer are fed into an LSTM layer with 16 units to learn the long-term dependencies between the time steps. The ReLU activation function is used since it is a non-linear function allowing complex relationships to be learned from the data. This would be difficult to do so from a linear function. A 1D representation is output from this layer to act as the input for the following layer.

3. Repeat vector converts it back to 2 dimensions by repeating the 1D array 90 times. Therefore, the output can be fed into the next LSTM layer for reconstruction.

4. The following LSTM layer reconstructs the sequence to 16 units as initially performed on the first LSTM layer.

5. A dense layer that has 51 nodes wraps around the time distribution function and outputs the sequence in the original dimensions as seen in the input layer. This concludes the process of deconstructing and reconstructing the input sequence.

The flow diagrams of the training and inference stages are summarised in Figure 21.

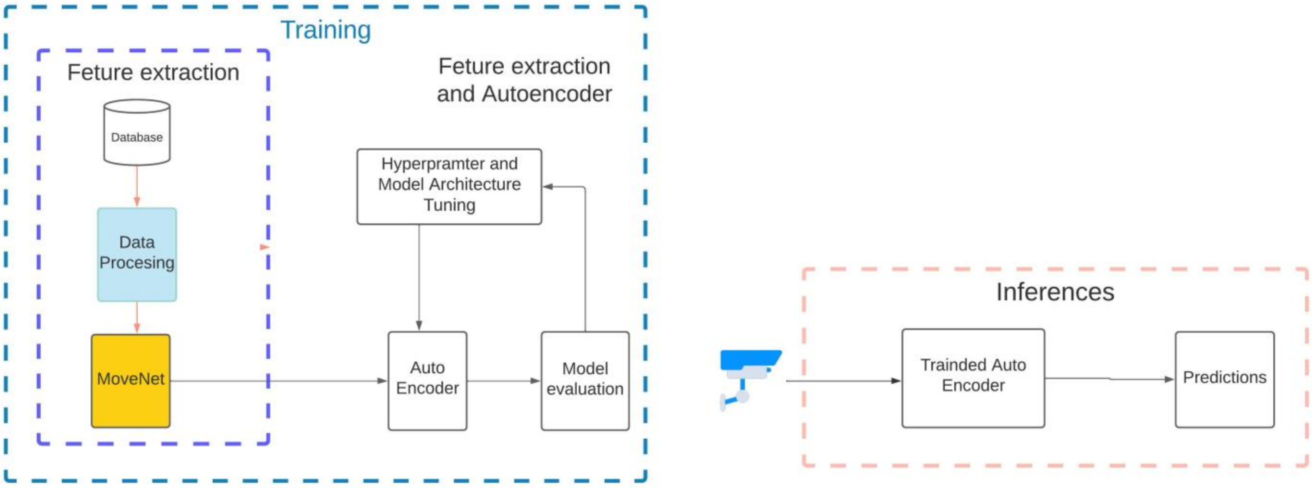


Figure 21: Process flow for the training and inferences stages of the autoencoder model

5.2.4. Model Training

The training of the neural networks was monitored by plotting the custom loss function, which increases as the reconstruction error between the original input and reconstructed input increases.

The learning curve from the training dataset shows how well the model is learning, while the learning curve from the validation dataset shows how well the model is able to generalise. Training the model beyond a certain number of epochs will result in the overfitting of the model to the noise in the training dataset and will no longer be able to generalise on an unseen dataset. This is the motivation for monitoring the training process.

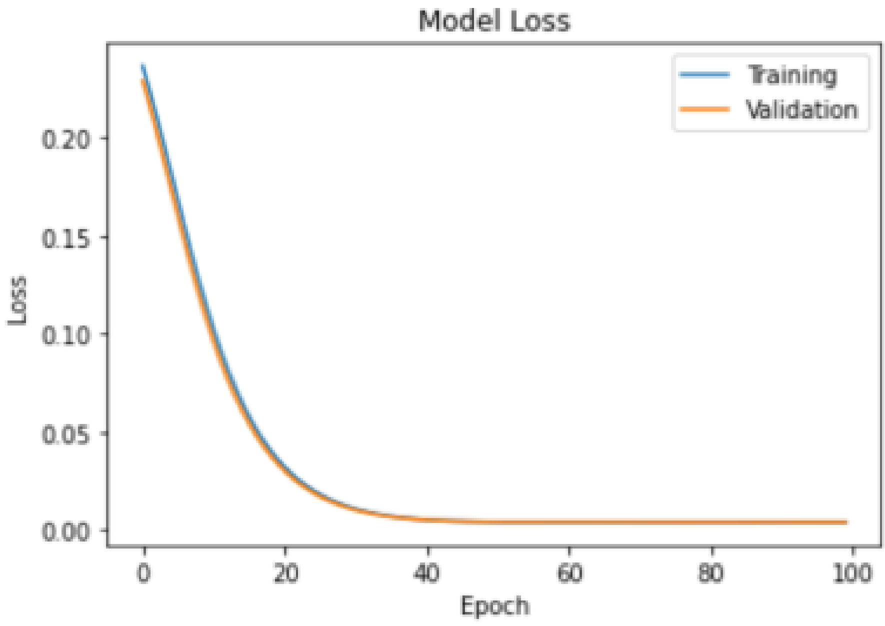


Figure 22: Learning curve for the autoencoder model

Figure 22 shows the learning curve for the autoencoder model, based on the custom loss. Early stopping monitor was implemented with a patience value of 3 epochs. When the validation loss increases beyond 3 consecutive epochs, the training stops and the best weights before the trigger are applied. A learning rate call back was used to control the learning rate. When the loss does not reduce for 2 consecutive epochs, the learning rate is reduced by 90%.

The figure shows that the model has achieved a good fit. This means that the validation loss has both reached a level of stability and has a minimal gap with the training curves. Further training will result in an overfit model.

In addition, the cosine similarity was plotted as an additional metric, which is shown in Figure 23.

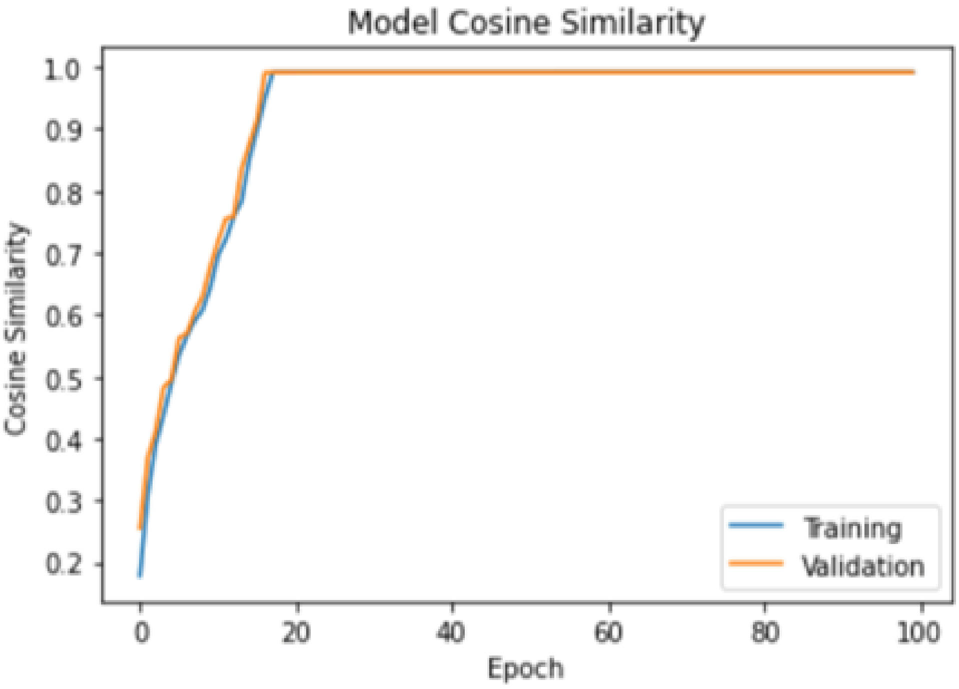


Figure 23: Cosine similarity metric plotted against number of epochs

5.2.5. Hyperparameter Tuning

The following hyperparameters were tuned for the autoencoder models:

1. Epochs represents the number of times the learning algorithm works through the dataset through the forward and backpropagation processes. The number of epochs was set to a high enough number so that the model could learn from the training data.

2. Batch size is an important hyperparameter that affects the training process. It is the number of samples from the training dataset processed before the model is updated. Small batch sizes guarantee convergence, at the cost of high computational time. However, a large batch size results in faster computation, at the cost of poor generalisation [19].

3. Early stopping was used to monitor the loss-function with number of epochs. The training loss decreases with epochs, but beyond a certain limit the model fits to the noise in the training data and cannot generalise to new data. Therefore, the loss function of the hold-out validation dataset is monitored with the loss function of the training dataset. When there is an increase in the validation loss at an epoch when compared to previous epochs, the learning process is stopped by specifying a patience parameter.

4. Learning rate is a crucial hyperparameter that decides the extent to which the new weights are changed after each epoch. Specifying too large learning rates will lead to an unstable learning process and potentially miss the minima. In contrast, a small learning rate will lead to a long computation time. The best approach is to use an adaptive learning rate as shown in Figure 24. This was implemented by using a proven optimisation algorithm, Adaptive moment estimation (Adam), with a specified initial learning rate [20].

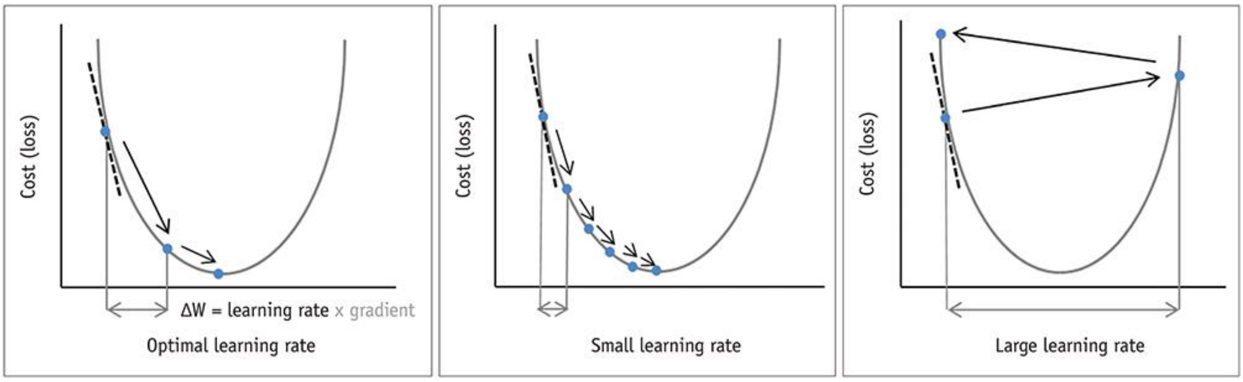


Figure 24: Effect of learning rate on weight optimisation [21]

The hyperparameters were trialled through an experimental process, to study the effects on the predictions and optimise. The final hyperparameters of the network are summarised in Table 7.

Table 7: Summary of hyperparameters used in the autoencoder model

	Auto Encoder
Epochs	100
Batch Size	8
Optimiser	Adam
Initial Learning Rate	0.01
Early Stopping Patience	3
Validation Split	20%

6. Testing and Validation

6.1. Experiment Description

Once both models were prepared, it was decided to test the system as well as the models with data from an environment similar to that of the deployment environment. The Cranfield Students' Association (CSA) was used as a test environment, where the students were requested to volunteer as participants, who were treated as passengers.

Both normal and drunk walking videos were collected and used to test the two models under development. The planning and execution steps followed are shown below in Figure 25. The consent forms and associated documentation are attached in Appendix A.

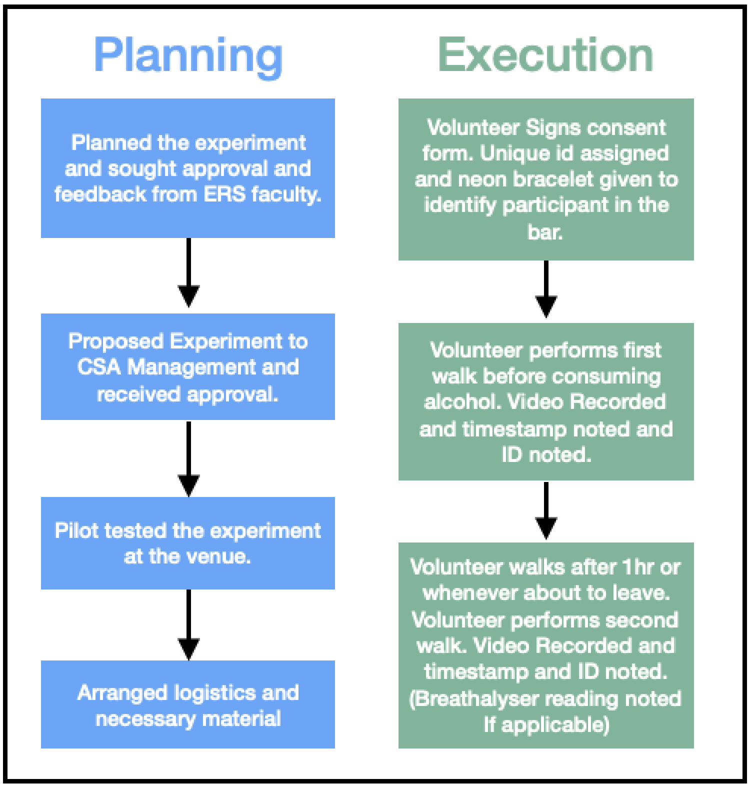

Planning
Execution
Planned the experiment and sought approval and feedback from ERS faculty.
Proposed Experiment to CSA Management and received approval.
Pilot tested the experiment at the venue.
Arranged logistics and necessary material
Volunteer Signs consent form. Unique id assigned and neon bracelet given to identify participant in the bar.
Volunteer performs first walk before consuming alcohol. Video Recorded and timestamp noted and ID noted.
Volunteer walks after 1hr or whenever about to leave. Volunteer performs second walk. Video Recorded and timestamp and ID noted. (Breathalyser reading noted If applicable)

Figure 25: Planning and execution of the CSA experiment

6.2. The Testing Environment

A section of the bar was used to simulate an airport chokepoint, such as that of the security checkpoints. The dimensions of the testing environment were kept as close as possible to those recommended by Transportation & Security Administration's (TSA) guidelines illustrated in Figure 26. The test environment used at the CSA that simulates the airport environment is shown in Figure 27.

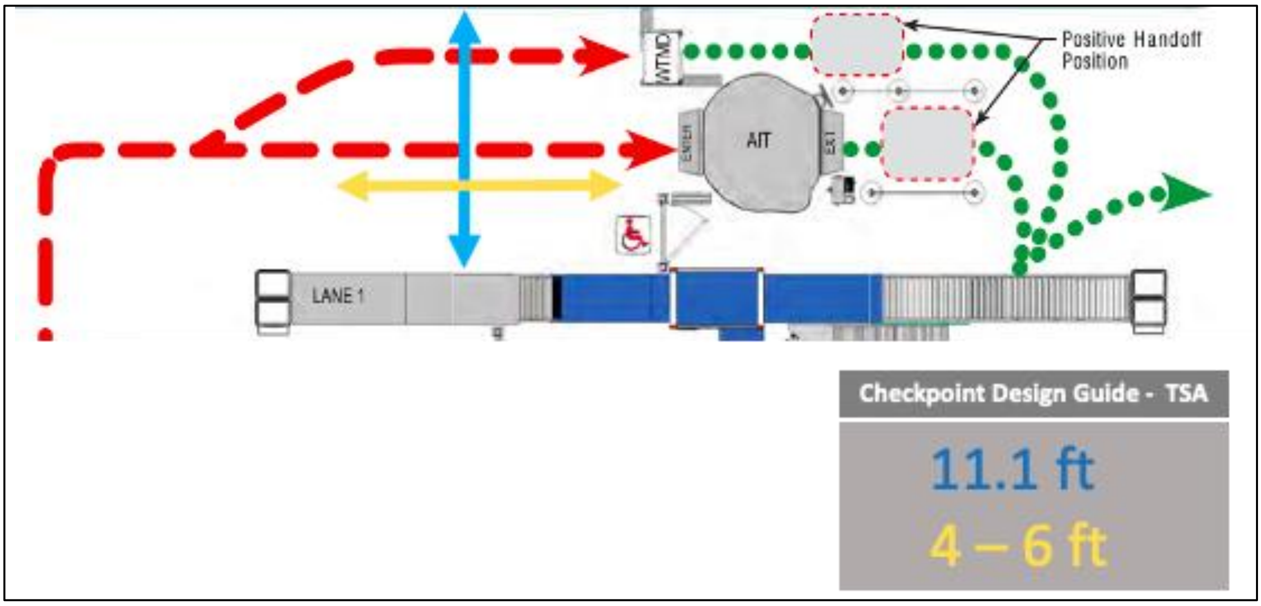


Figure 26: TSA recommended airport layouts

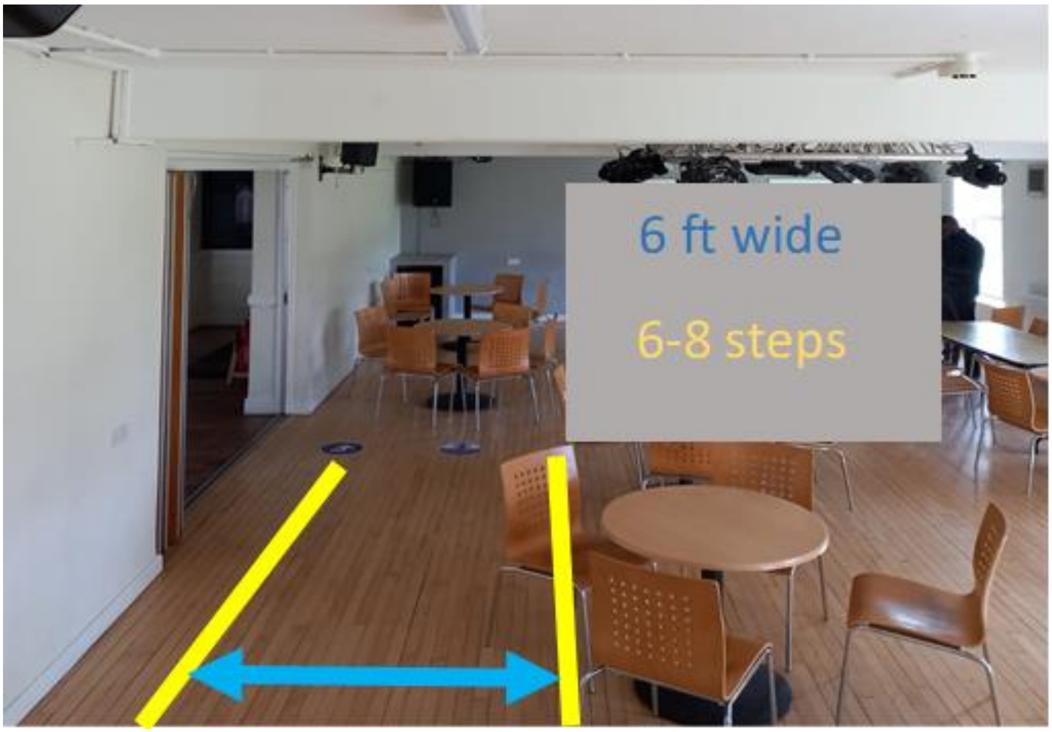


Figure 27 CSA Testing Area – Chairs and tables were removed to clear a path that would be like that of an airport checkpoint.

6.3. Objectives of the Experiment

The objectives of the Preliminary testing at the CSA were to:

1. Conduct mid-development analysis of how the system would perform.
2. To compare the performance of both models to determine which one to move forward with.
3. Gather additional data most representative of deployment environment.
4. Identify any design changes and possible shortcomings that might arise when humans would interact with the system.

6.3.1. Conducting Mid-Development Analysis

With regards to the first objective, camera positioning and angles were tested. The camera was situated close to the ceiling as it would be located on top of a metal detector at the airport. In order to minimise unwanted field of vision, the camera was set to record videos in portrait orientation, resulting in fewer people in the frame making it easy for the model to focus on people walking towards the camera. Figure 28 shows the selection of the field of vision. The region shaded green was selected as the preferred option, as opposed to the red box in the landscape orientation of the camera.

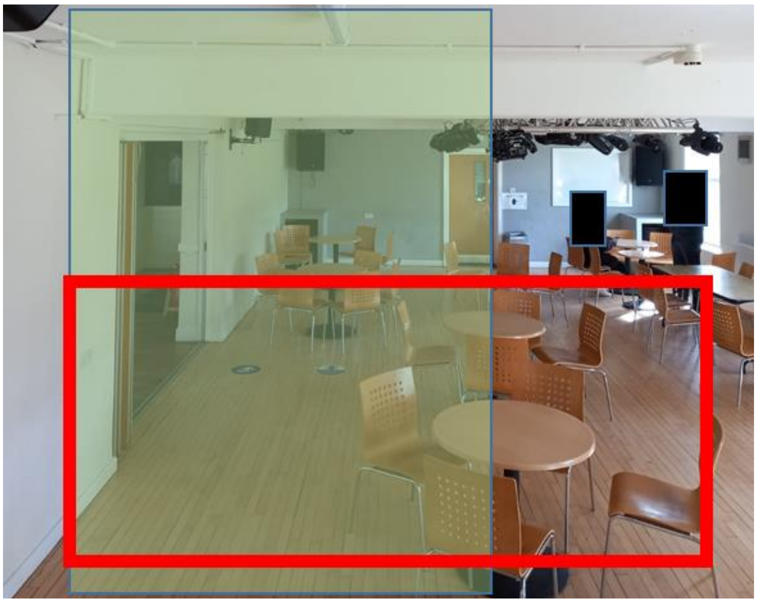


Figure 28: Selecting the field of Vision

6.3.2. Comparing Performance of the Models

Secondly, the data generated when used with both models resulted in better results for approach 2, which used the auto encoder to determine the intoxicated state of a person. This allowed us to choose auto encoder approach as the preferred way forward.

6.3.3. Gather Data from Simulated Airport Environment

Thirdly, one of the major concerns during the project development was a lack of training data available for the auto encoder approach. The performance of the autoencoder depended on the diversity of training data. A very important by-product of this experiment was the collection of normal walking videos of 96 different people, thus providing adequate variance in the training dataset.

6.3.4. Identify Shortcomings During Human Interaction with the System

Finally, in the case of the rule-based approach, people were required to display enough trajectory deviation for the model to calculate motion deviation. One of the concerns was to check the likelihood of significant deviation in walking trajectory after a person was directed to walk towards a target in a controlled environment. It was observed that even with the most intoxicated participant, there was not much visible trajectory deviation.

This suggested that the auto encoder approach was better suited for this problem. It was also observed that airport design might have to be changed slightly to increase the walking distance to the body scanner so that a more reliable inference can be made by the model.

6.3.5. Limitations of the Experiment

While measures were taken to create a similar environment to that of an airport checkpoint, there were certain limitations that were discovered during and after the experiment was completed:

1. Dark figures (skin and clothing colour) were not identified by the pose estimation model. The causes for this were suspected to be the usage of a phone front camera and the orientation of lighting.
2. The breathalyser which was used could not be operated on a candidate if they consumed alcohol within 90 minutes of alcohol consumption. Moreover, the device was unable to conduct more than 5 tests in an hour without residual alcohol build-up within the device affecting the accuracy of the results.
3. The most intoxicated subjects who reported the highest blood alcohol Levels during the experiment were not drunk enough to demonstrate significant motion deviation in their walking trajectory.
4. It was noticed that subjects became camera conscious when walking towards the designated point. This could have resulted in the participants making a conscious effort to walk straight instead of their naturally intoxicated walk.
5. Some people took the experiment as a game and challenged themselves to walk straight while consuming alcohol.

7. Results and Model Selection

7.1. Rule-Based Approach

The primary metric used to determine whether a person was drunk or not was the motion deviation score which measured the deviation of the person from his/her intended trajectory. Given the upper/lower thresholds used to control deviation scores using the cumulative averaging technique from literature [9], the motion deviation scores were calculated for each person walking.

A threshold score for the cumulative motion deviation of 16 was used for classifying based on [9]. All people who had motion deviation scores above this threshold were classified as drunk and all the people who had motion deviation scores below this threshold were sober. When evaluated on the testing dataset, a confusion matrix was produced as shown in Figure 29.

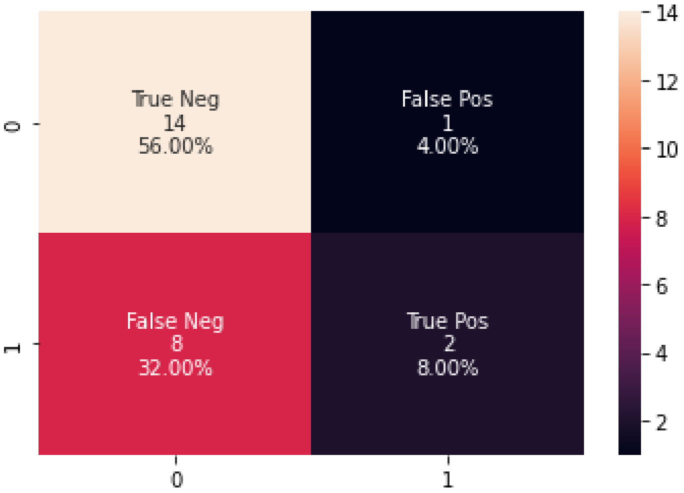


Figure 29: Confusion Matrix - Rule Based Approach

The above confusion matrix was used to determine the following evaluation metrics in Table 8.

Table 8: Precision, Recall and F1 Score for Rule Based Approach

Precision	0.67
Recall	0.2
F1-Score	0.308

Precision
Precision was used as the driving metric for classification. It demonstrates the fraction of true positives with respect to total number of positive predictions (True Positives + False Positives). The model was able to achieve 67% precision. This meant that 33% of the people predicted to be drunk were possibly sober. This is not desirable since a large percentage of people could be misclassified as drunk.

Recall
Recall demonstrated a fraction of true positives with respect to total number of actual positives (True Positives + False Negatives). The model was able to achieve 20% recall. This meant that 80% of the people who were actually-drunk were predicted to be sober. This demonstrated that the rule-based technique failed to detect the majority of the drunk persons. It further led the team to believe that this was not a suitable technique to be deployed.

7.2. Auto Encoder Approach

The primary metric used to determine whether a person was drunk or not was the mean absolute error, which was used to determine the reconstruction error of the auto encoder trained on normal walking behaviour. The higher the mean absolute error, the more abnormal the behaviour was predicted to be, in this case the abnormal behaviour was associated with drunk behaviour.

Since there was a range of values for the mean absolute error, it was decided to find a threshold which would ensure minimum false positives without affecting the true positives by a significant amount. This was achieved using a histogram of the mean absolute errors, as shown in Figure 30.

Since the true classes are visible on the histogram, the threshold was set in such a way that it minimises the instances of false positives. The reason for minimisation of false positives was to mitigate the adverse effects of misclassification and further triggering passenger anger due to wrongful classification.

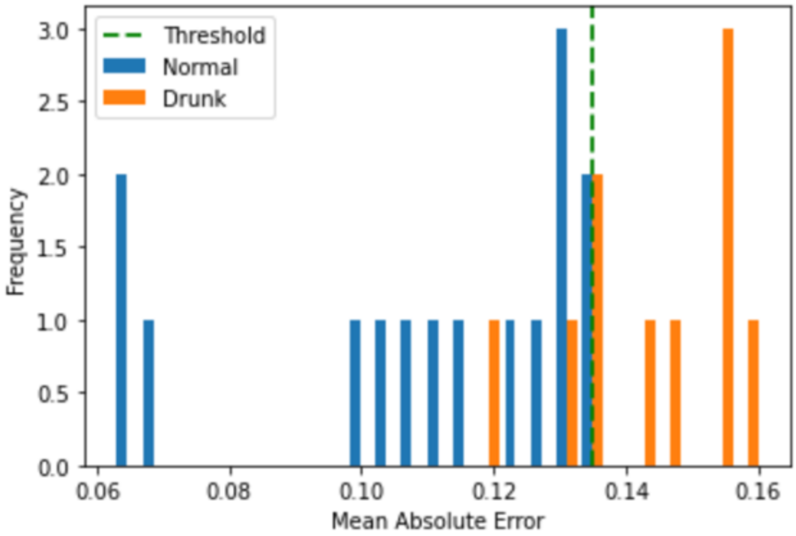


Figure 30: Histogram of auto-encoder Mean Squared Errors for reconstruction.

The above technique resulted in a threshold of 0.135, which was then used to construct the confusion matrix shown in Figure 31. Below the threshold, persons were considered as sober and above this threshold they were classified as drunk.

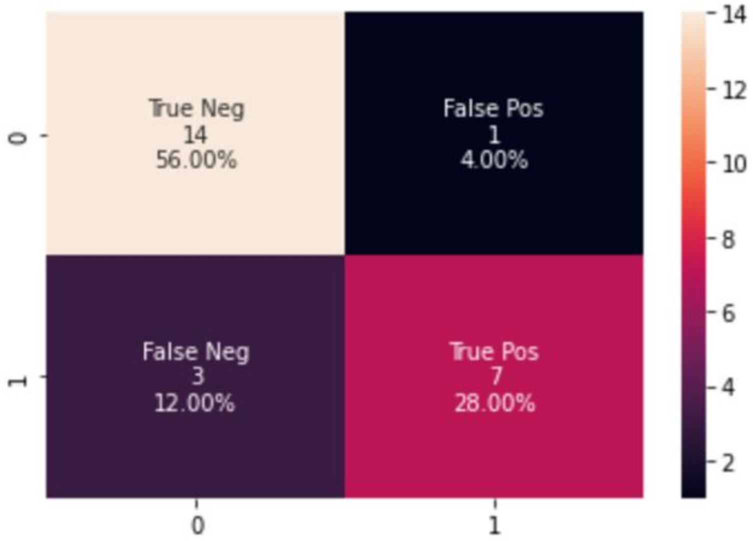


Figure 31: Confusion Matrix - Auto Encoder Technique

The above confusion matrix produced the following precision, recall and F1-Score metrics in Table 9.

Table 9: Precision, Recall and F1 Score - Auto Encoder Approach

Precision	0.875
Recall	0.700
F1-Score	0.778

Precision

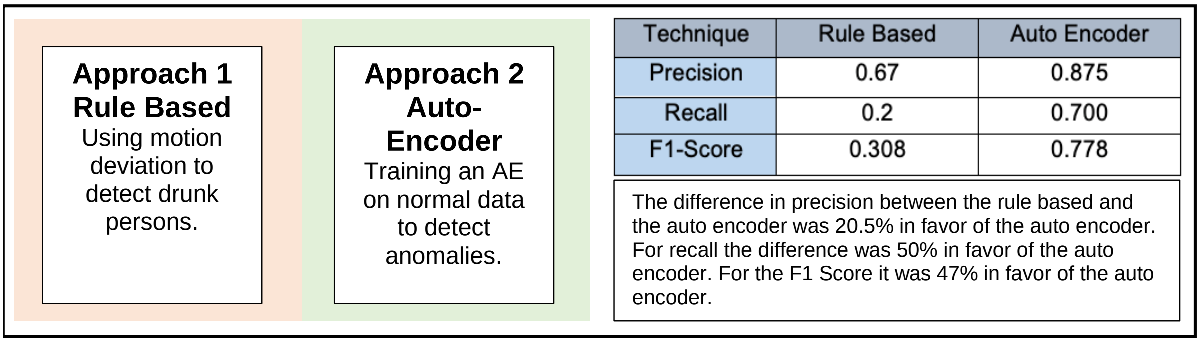
Precision was used as the driving metric for classification. It demonstrates the fraction of true positives with respect to total number of positive predictions (True Positives + False Positives). The model was able to achieve 87.5% precision. This meant that 12.5% of the people predicted to be drunk were possibly sober. The performance is expected to get better once the system is placed in the field and is trained with actual data over time. For now, results are considered to be promising.

Recall

Recall demonstrated a fraction of true positives with respect to total number of actual positives (True Positives + False Negatives). The model was able to achieve 70% recall. This meant that 30% of the people who were actually-drunk were predicted to be sober. At an airport with significant passenger flow and high potential for disruption due to misclassification, this was an acceptable compromise in favour of high precision mentioned earlier.

7.3. Final Model Selection

As mentioned in the report earlier, two approaches were explored in parallel in order to determine the most effective technique to detect intoxicated passengers at airports.



Approach 1
Rule Based
Using motion deviation to detect drunk persons.

Approach 2
Auto-Encoder
Training an AE on normal data to detect anomalies.

Technique	Rule Based	Auto Encoder
Precision	0.67	0.875
Recall	0.2	0.700
F1-Score	0.308	0.778

The difference in precision between the rule based and the auto encoder was 20.5% in favor of the auto encoder. For recall the difference was 50% in favor of the auto encoder. For the F1 Score it was 47% in favor of the auto encoder.

Figure 32: Comparison of Rule Based and Auto Encoder approaches

Both approaches were tested on the same set of test data to ensure a like-for-like comparison. After testing both approaches on real world data, it was noticed that the auto encoder performed better, while the rule-based approach required significant abnormal behaviour, which could also be detected by human observation.

The auto encoder demonstrated the ability to detect small nuances, which were not even discernible by human evaluation. Moreover, the auto encoder demonstrates the promise of further improvement in performance after deployment at airports. The more normal data is provided to the auto encoder, the better it can become at detecting anomalies. Therefore, with more data being gathered during initial operation at airports, the model demonstrates potential to perform better. The auto encoder model will improve even further over time, this would not be possible with the rule-based approach. Therefore, the auto encoder approach was selected as the final model for deployment.

8. System Integration

To demonstrate a proof of concept, the deployment of the chosen model (auto encoder) was carried out on Amazon Web Services (AWS). AWS will allow for cloud GPU access for the training, testing and real-time inference using the model. Moreover, it demonstrates the cloud-based deployment that would be a key requirement at smart airports.

8.1. AWS SageMaker

The AWS SageMaker can create endpoints that are fully managed services that allow the users to make real-time inferences or batch inferences. For this project, the requirement was to do real-time inferences. When creating a SageMaker endpoint, a few factors need to be considered.

The first is cost efficiency. The cost was determined by the type of instances used and the number of hours the model instances were used, including the idle time. For this project, two models were used, the autoencoder model was a lightweight model with a few layers. However, the MoveNet Lighting model demands considerable amount of computation power. Therefore, to deploy the autoencoder model "T2.medium" instances were used. T2 instances do not contain any GPU. However, as the MoveNet model required a GPU to get real-time inferences, a "P2.xlarge" instance was used to create an endpoint. The elastic inferences can be used to increase the computation power the endpoint has on demand.

The first step of deploying a machine learning model as a SageMaker endpoint is uploading the trained model architecture and weights to an S3 bucket. To deploy the model as a SageMaker endpoint, the saved model needs to be in a ".pb" format and the file structure needs to follow the required format for AWS. Once the model was converted to the required format, the model was deployed by calling the "deploy" method in SageMaker.

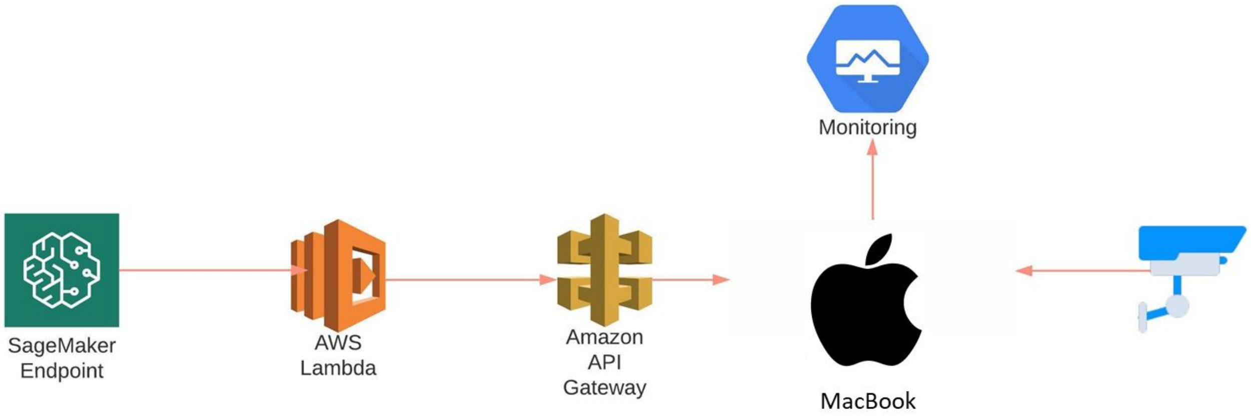


Figure 33: System Integration Flowchart

8.2. AWS Lambda

Once the endpoint was created, the next step was to connect the endpoint with the CCTV to send the data from the CCTV to the endpoint and make inferences. To make this connection two more services that are available in the AWS were used. The first one was AWS Lambda function. This service allows running a code that can unpack the data received from the CCTVs and convert it to the required format to pass it to the endpoint.

8.3. AWS API Gateway

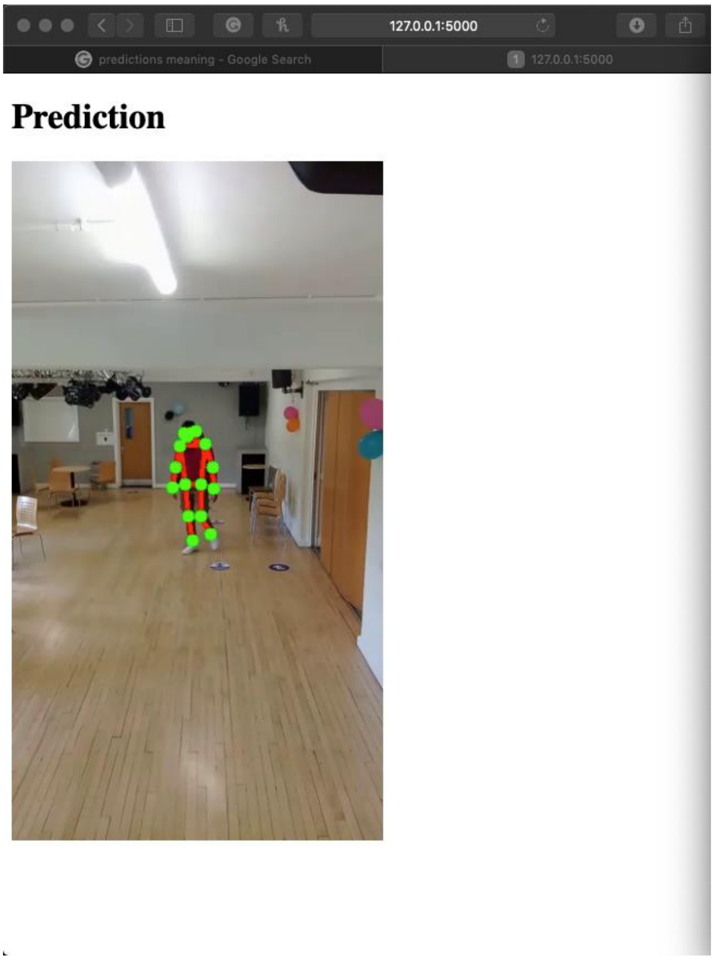
API gateway is another fully managed service that was used to create a REST API endpoint that was used by the web application to send data using the HTTPS “post” protocol. Following this, the data was passed back to AWS Lambda.

8.4. Web Application

To visualise the inference data that AWS sends and to send the data from the CCTVs, a web application was developed. To create the web application, Flask and HTML were used.

The created web application is able to display the live streams received from the CCTVs. As a prototype, the web application was programmed to use the web camera of the MacBook as the group did not have access to a CCTV. OpenCV was used to access the webcam and process the data so that a CCTV camera can be connected as an IP camera to the backend of the web application.

The web application was coded to convert the frames received from the CCTVs and collect all the frames as a list and then pass it to the API endpoint once it collects 90 frames (time duration expected by the auto encoder). The list was converted to a JSON file and sent to the API endpoint to make inferences. From the model prediction, the mean average error (MAE) value was calculated, and classification was made and the result was displayed on the livestream. Figure 34 shows the developed web application.



Prediction

Figure 34: Web application with the inferences displayed

9. Systems Engineering and Project Management

9.1. Project Team

Figure 35 illustrates the organisation chart (organogram) of the project team, which highlights the organisation structure, roles and responsibilities.

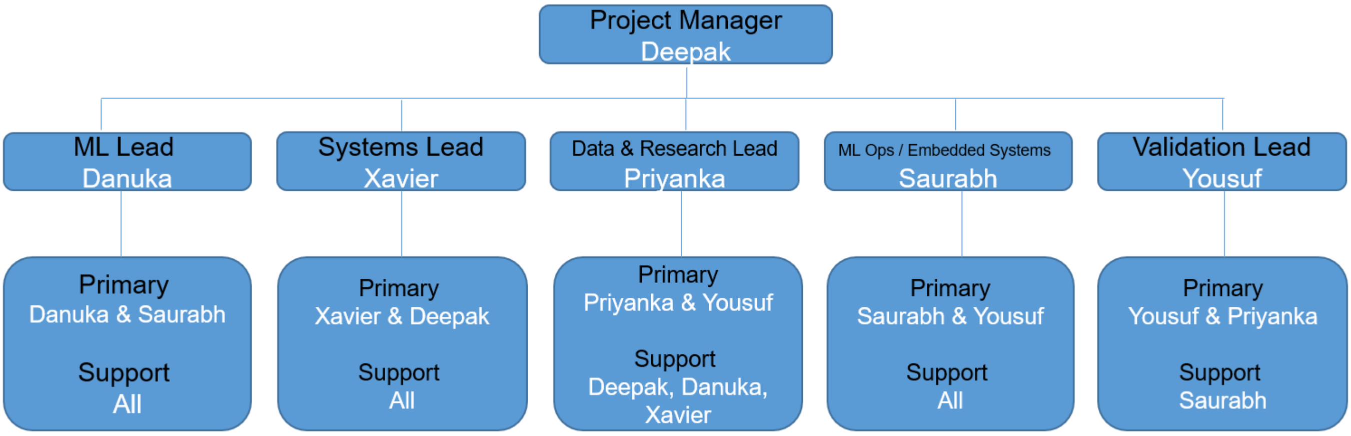


Figure 35: Project Organisation Chart

9.2. Project Schedule

The Gantt chart for this project is presented in Appendix C. It was used to monitor and control the project’s progress during the course of the work in order to mitigate the risk of schedule slippage and avoid the undesirable outcome of late project completion.

9.3. Risk Management

Managing systems engineering and project risks on the project was essential to delivery of the AI system on time, within budget and ensuring that it meets the expected performance requirements.

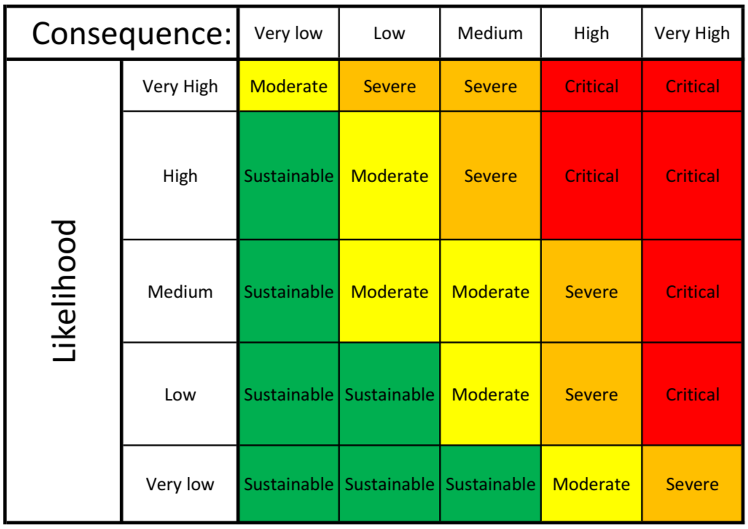
The risks to the system development effort, and the overall success of the project were identified, analysed and mitigated in order to eliminate or minimise any possible negative performance, cost or schedule impact.

9.3.1. Risk Analysis and Assessment

The team arrived at the identified risks by leveraging on the combined education and experience of all the team members to examine and analyse the identified system requirements during a risk assessment brainstorming session carried out on this project. During the risk assessment, the risks were identified and analysed using a risk matrix (see Table 10) to generate the risk register (see Appendix D) which was used to monitor the risks during the project.

The risk matrix presented in Table 10 was used to measure the overall impact of the identified risks during a risk assessment session of the system development and the project.

Table 10: Risk matrix



Consequence:		Very low	Low	Medium	High	Very High
Likelihood	Very High	Moderate	Severe	Severe	Critical	Critical
	High	Sustainable	Moderate	Severe	Critical	Critical
	Medium	Sustainable	Moderate	Moderate	Severe	Critical
	Low	Sustainable	Sustainable	Moderate	Severe	Critical
	Very low	Sustainable	Sustainable	Sustainable	Moderate	Severe

9.3.2. Risk Mitigation Status

Table 11 shows a summary of the mitigation status of the identified systems engineering and project risks.

Table 11: Risk mitigation status

Risk Category	Risk Description	Risk Mitigation	Actions Taken
Technical Risks	Cyber security	AWS security protocols	AWS subscribed.
	Limited training data	Use an auto-encoder (trained on normal videos). Create additional datasets using real-world video recordings.	Auto-encoder trained on available normal videos. Obtained real-world video recordings from the CSA.
Environmental Risks	Occlusion	Minimal occlusion due to implementation in streamlined areas such as security checkpoint.	Simulated people walking towards a chokepoint at the CSA and recorded the videos.
	Human pose variations	Implement in predictable environment such as security checkpoint (passenger expected to walk).	Simulated walking towards security checkpoint at CSA.
Project Risks	Cost	Off the shelf components (Jetson Nano, web camera and AWS Subscription).	Procured off the shelf components.
	Late components procurement	Off the shelf components (next day delivery).	Procured: NVIDIA Jetson Nano, web

			camera, AWS subscription.
	Procured defective components	Test component once received in case it needs to be replaced.	Tested components.
	Late algorithm completion	Track and measure actual against planned progress.	Completed.
Operational Risks	False positives	Target high precision (≥ 0.8).	Completed.

9.4. Specialised and Collaboration Tools

In the course of carrying out this work, the project team used the following specialised and collaborative tools:

Table 12: Specialised and collaboration tools used in the project

S/N	APPLICATION SOFTWARE	DESCRIPTION	PROJECT USE
1	MS Teams	A collaboration software application developed by Microsoft Corporation.	It was used by the project team for project related task tracking, videoconference meetings, file sharing and chatting.
2	WhatsApp	A multi-platform instant messaging application developed by Meta Platforms, Inc.	The project team used WhatsApp for exchanging project related text messages, sharing documents, videos, images, and making voice and video calls.
3	Google Colab	A Google LLC web-based type of Jupyter Notebook that permits collaborative writing, editing and interaction with the Notebook on Google Cloud with a relatively fast code execution time using its GPU (Graphics Processing Unit) and TPU (Tensor Processing Unit) options.	It was used by the project team for algorithm development.
4	AWS	Amazon Web Services, Inc. is a subsidiary of Amazon providing on-demand cloud computing platforms and APIs to individuals, companies, and governments, on a metered pay-as-you-go basis.	It was used for model deployment of the rule-based approach.
5	GitHub	A collaborative internet hosting website for software development and version control developed by GitHub, Inc.	It was used by the project team for algorithm development.

10. Conclusions

Alcohol intoxication is one of the major causes of disruptive behaviour at airports and onboard aircrafts. This project investigated AI-based solutions to identify intoxicated passengers at airports. The rationale for doing so is that the best means to prevent disruption onboard aircrafts is to identify such behaviours early in the process i.e., at the airports.

Research strongly suggested the use of pose estimation for this application due to its neutrality compared to other methods such as facial recognition. Therefore, two pose-estimation based approaches were developed and tested in parallel: rule-based approach and auto encoder approach. Both of the approaches used existing literature as the starting point with elements of AI innovation by the project team.

An experiment was conducted at the CSA to collect data of normal and drunk walking videos from an environment that simulates the intended deployment environment at the airports. The experiment was also used to identify potential limitations of the system in the deployment environment. The data collected was used to train and test the systems with a common dataset.

The results showed that the auto encoder approach was able to perform better than the rule-based approach. This is because the rule-based system depends on a very high level of intoxication leading to zig-zag walking. This level of intoxication was not present in any of the participants in the experiment, nor is it common in the airport scenario. The auto encoder was however able to pick up on subtle differences in the walking sequence through the use of reconstruction error as the classifying parameter.

The auto encoder delivered a precision of 0.875 and a recall of 0.7. These are promising results and deliver a proof of the concept. This was possible through the AI innovation in the project, which is the addition of the time sequence of the frames in the video through an LSTM model.

For successful implementation of the proposed system at airports, some key considerations need to be made. It was observed that lighting had a significant effect on the accuracy of the model. Therefore, if this model is implemented in actual airports, the lighting conditions should be in compliance with the pose detection models. Walking distance to the camera should be sufficient to make an inference, this might require slight changes to the airport layout.

11. Further Work

Some further work that has been identified will be required to build upon this proof of concept:

- Deploy in an airport environment to train the auto encoder model with more data. This will require a human in the loop to carry out and validate the model predictions.

- The model could be deployed in non-chokepoint environments with more crowd and occlusion. This would require higher model complexity such as the inclusion of action detection as the diversity of actions is expected to increase.

- Ensure the model is trained with a diverse dataset, including but not limited to elderly and disabled people.

12. References

[1] S. S. Mclinton, D. Drury, S. Masocha, H. Savelsberg, and K. Lushington, "'Air Rage': A systematic review of research on disruptive airline passenger behaviour 1985-2020," *Journal of Airline and Airport Management*, vol. 10, no. 1, pp. 31–49, 2020, doi: 10.3926/jairm.156.

[2] Institute for Alcohol Studies, "Alcohol and Air Travel," 2020. Accessed: Mar. 22, 2022. [Online]. Available: https://www.ias.org.uk/wp-content/uploads/2020/08/Alcohol-and-air-travel.pdf

[3] "Manchester Airport: Air rage and disorder doubled since 2014 - BBC News." https://www.bbc.co.uk/news/uk-england-manchester-40849181 (accessed Mar. 23, 2022).

[4] "Grounded: Police warn drinkers at Manchester airport bars they could be stopped from flying - Manchester Evening News." https://www.manchestereveningnews.co.uk/news/greater-manchester-news/grounded-police-warn-drinkers-at-manchester-690043 (accessed Mar. 23, 2022).

[5] "Airport staff tip off police for group bookings to these three resorts in booze crackdown - YorkshireLive." https://www.examinerlive.co.uk/news/uk-world-news/airport-staff-tip-police-group-16819201 (accessed Mar. 23, 2022).

[6] V. Mehta, D. P. Yadav, S. S. Katta, and A. Dhall, "DIF : Dataset of Perceived Intoxicated Faces for Drunk Person Identification." [Online]. Available: https://sites.google.com/view/difproject/home

[7] "Racial Discrimination in Face Recognition Technology - Science in the News." https://sitn.hms.harvard.edu/flash/2020/racial-discrimination-in-face-recognition-technology/ (accessed Mar. 23, 2022).

[8] Department for Transport, *Code of Practice for Preliminary Impairment Tests*. 2017. Accessed: Mar. 25, 2022. [Online]. Available: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/607267/Fit_Code_of_Practice_1st_April_2017.pdf

[9] J. Y. Lee, S. Choi, and J. Lim, "Detection of High-Risk Intoxicated Passengers in Video Surveillance," *Proceedings of AVSS 2018 - 2018 15th IEEE International Conference on Advanced Video and Signal-Based Surveillance*, Feb. 2019, doi: 10.1109/AVSS.2018.8639485.

[10] O. Temuroglu *et al.*, "Occlusion-Aware Skeleton Trajectory Representation for Abnormal Behavior Detection," 2020.

[11] C. Rizk, F. Mora-Camino, and H. Batatia, "Optimization of Passenger Screening Operations in Air Terminals," *Transportation Research Procedia*, vol. 35, pp. 23–34, Jan. 2018, doi: 10.1016/J.TRPRO.2018.12.004.

[12] S. O'Neill and B. S. Skift, "The Rise of Smart Airports: A Skift Deep Dive," *Skift*, Nov. 20, 2019. https://skift.com/2019/11/20/the-rise-of-smart-airports-a-skift-deep-dive/ (accessed Mar. 25, 2022).

[13] "ChokePoint Dataset." http://arma.sourceforge.net/chokepoint/ (accessed Apr. 05, 2022).

[14] "OpenPose: OpenPose Doc - Output." https://cmu-perceptual-computing-lab.github.io/openpose/web/html/doc/md_doc_02_output.html (accessed Apr. 01, 2022).

[15] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields," *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, vol. XXX, Accessed: Apr. 04, 2022. [Online]. Available: https://gineshidalgo.com

[16] R. Fabré Solà, "Development and Integration of a Real-time Human Pose Estimation and Activity Classification System," 2020.

[17] N. Wojke, A. Bewley, and D. Paulus, "SIMPLE ONLINE AND REALTIME TRACKING WITH A DEEP ASSOCIATION METRIC."

[18] T. Lin, B. G. Horne, P. Tiňo, and C. L. Giles, “Learning long-term dependencies in NARX recurrent neural networks,” *IEEE Transactions on Neural Networks*, vol. 7, no. 6, pp. 1329–1338, 1996, doi: 10.1109/72.548162.

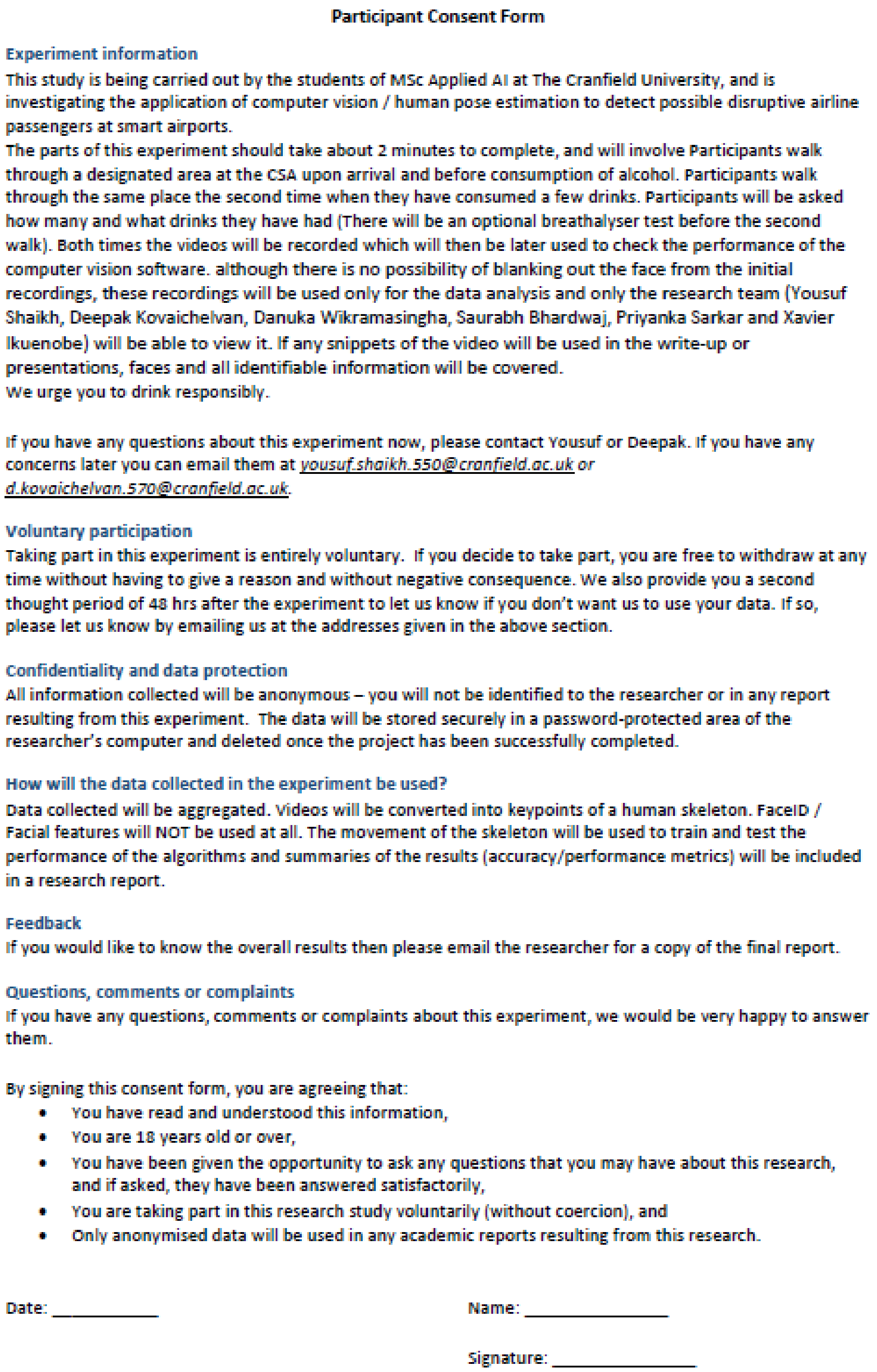
[19] J. Brownlee, “How to Control the Stability of Training Neural Networks With the Batch Size,” 2019. https://machinelearningmastery.com/how-to-control-the-speed-and-stability-of-training-neural-networks-with-gradient-descent-batch-size/ (accessed Jan. 28, 2022).

[20] S. Albahli, F. Alhassan, W. Albattah, and R. U. Khan, “Handwritten Digit Recognition: Hyperparameters-Based Analysis”, doi: 10.3390/app10175988.

[21] “How to Use Learning Rate Annealing with Neural Networks?” https://analyticsindiamag.com/how-to-use-learning-rate-annealing-with-neural-networks/ (accessed Jan. 28, 2022).

Appendices

Appendix A – Consent Form Template



Participant Consent Form

Experiment information

This study is being carried out by the students of MSc Applied AI at The Cranfield University, and is investigating the application of computer vision / human pose estimation to detect possible disruptive airline passengers at smart airports.

The parts of this experiment should take about 2 minutes to complete, and will involve Participants walk through a designated area at the CSA upon arrival and before consumption of alcohol. Participants walk through the same place the second time when they have consumed a few drinks. Participants will be asked how many and what drinks they have had (There will be an optional breathalyser test before the second walk). Both times the videos will be recorded which will then be later used to check the performance of the computer vision software. although there is no possibility of blanking out the face from the initial recordings, these recordings will be used only for the data analysis and only the research team (Yousuf Shaikh, Deepak Kovaichelvan, Danuka Wikramasingha, Saurabh Bhardwaj, Priyanka Sarkar and Xavier Ikuenobe) will be able to view it. If any snippets of the video will be used in the write-up or presentations, faces and all identifiable information will be covered.

We urge you to drink responsibly.

If you have any questions about this experiment now, please contact Yousuf or Deepak. If you have any concerns later you can email them at *yousuf.shaikh.550@cranfield.ac.uk* or *d.kovaichelvan.570@cranfield.ac.uk*.

Voluntary participation

Taking part in this experiment is entirely voluntary. If you decide to take part, you are free to withdraw at any time without having to give a reason and without negative consequence. We also provide you a second thought period of 48 hrs after the experiment to let us know if you don't want us to use your data. If so, please let us know by emailing us at the addresses given in the above section.

Confidentiality and data protection

All information collected will be anonymous – you will not be identified to the researcher or in any report resulting from this experiment. The data will be stored securely in a password-protected area of the researcher's computer and deleted once the project has been successfully completed.

How will the data collected in the experiment be used?

Data collected will be aggregated. Videos will be converted into keypoints of a human skeleton. FaceID / Facial features will NOT be used at all. The movement of the skeleton will be used to train and test the performance of the algorithms and summaries of the results (accuracy/performance metrics) will be included in a research report.

Feedback

If you would like to know the overall results then please email the researcher for a copy of the final report.

Questions, comments or complaints

If you have any questions, comments or complaints about this experiment, we would be very happy to answer them.

By signing this consent form, you are agreeing that:

- You have read and understood this information,
- You are 18 years old or over,
- You have been given the opportunity to ask any questions that you may have about this research, and if asked, they have been answered satisfactorily,
- You are taking part in this research study voluntarily (without coercion), and
- Only anonymised data will be used in any academic reports resulting from this research.

Date: ___________ Name: _______________

Signature: _______________

Appendix B – Python Code

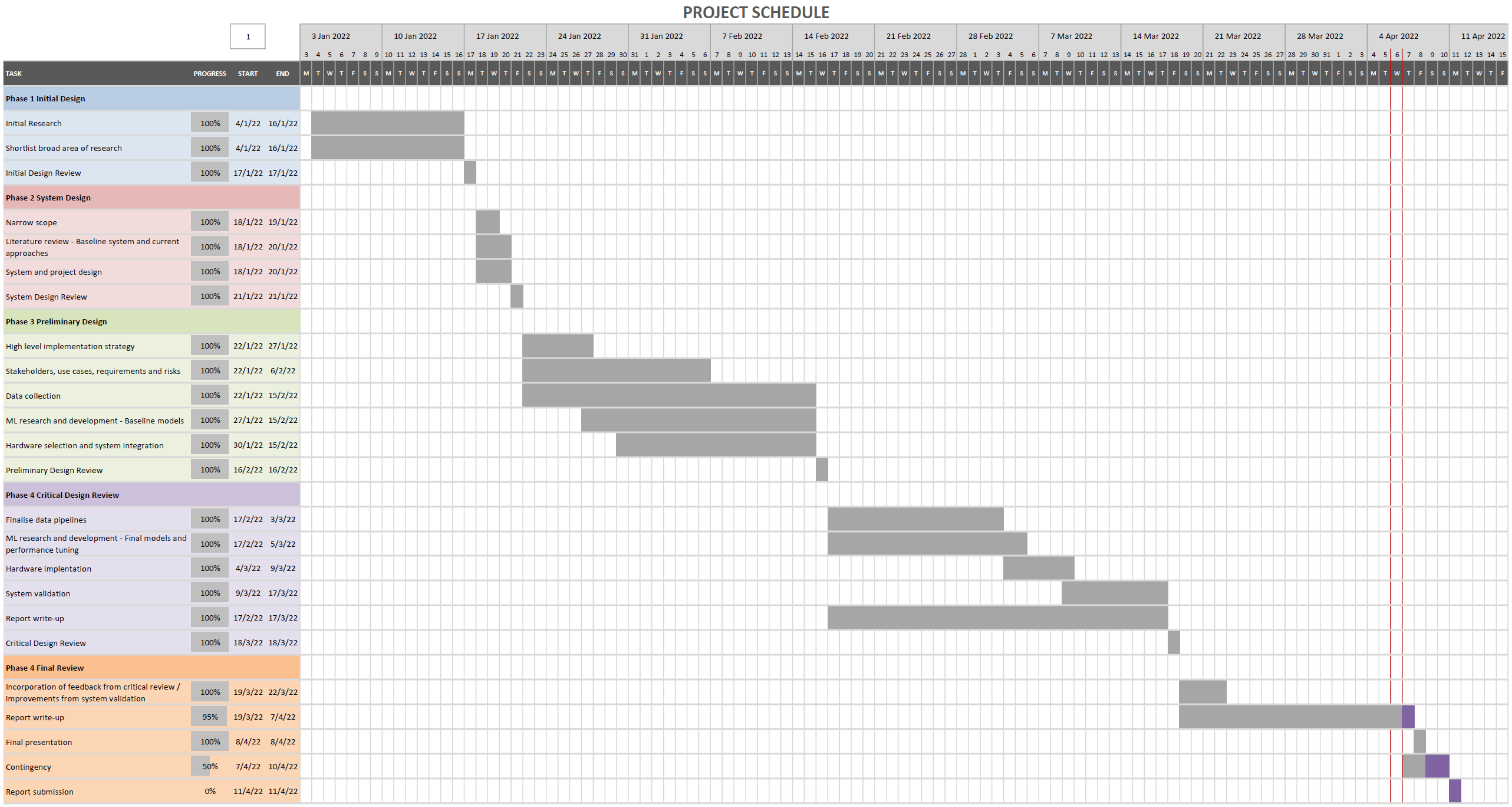
1. The rule-based approach algorithm is available to download from GitHub using the below URL:

   Rule-Based Approach

2. The auto encoder approach algorithm was the selected approach from the model selection exercise. The code is available to download from GitHub using the below URL:

   Auto Encoder Approach

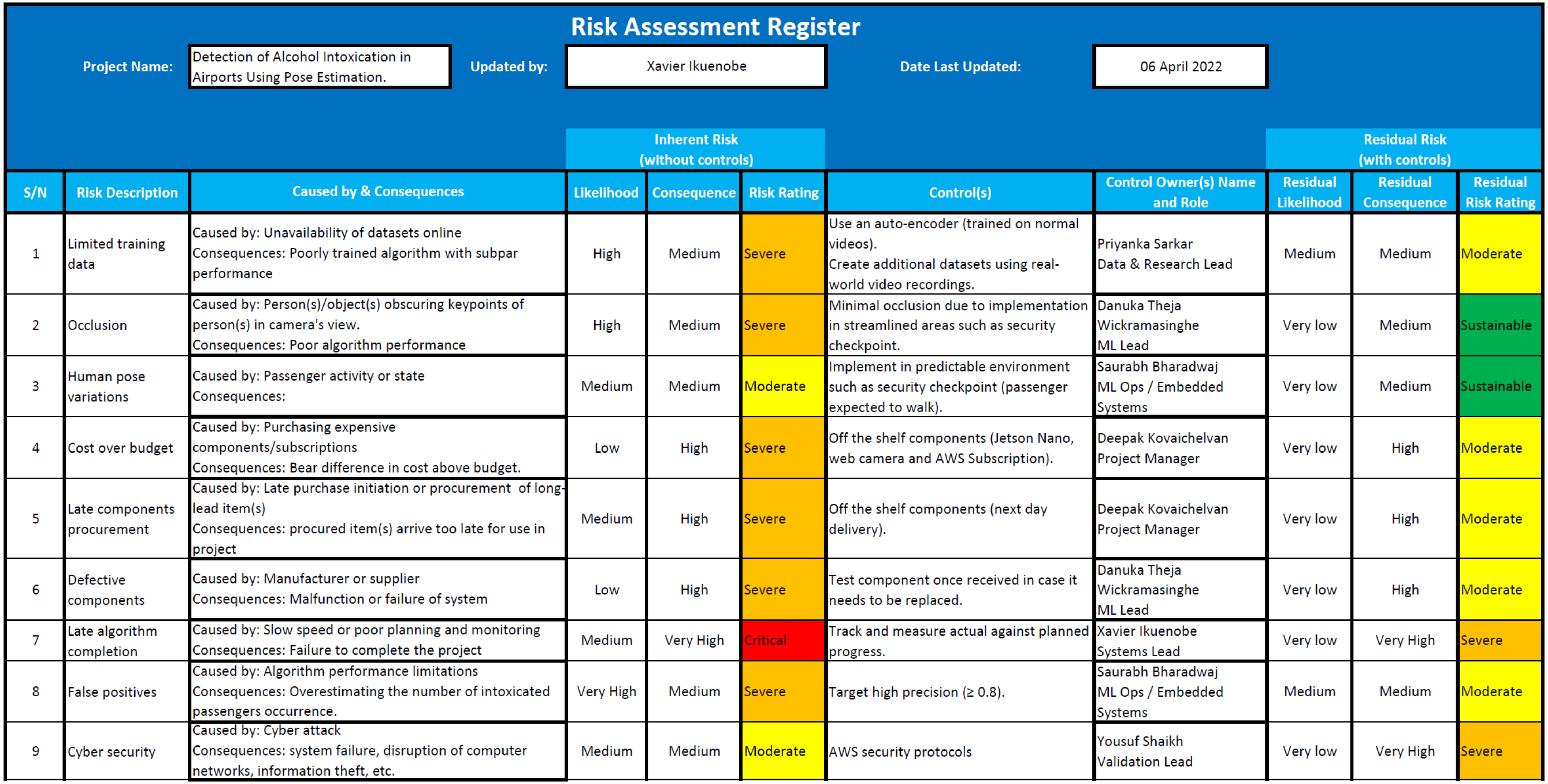
Appendix C – Gantt Chart



PROJECT SCHEDULE

TASK	PROGRESS	START	END
Phase 1 Initial Design			
Initial Research	100%	4/1/22	16/1/22
Shortlist broad area of research	100%	4/1/22	16/1/22
Initial Design Review	100%	17/1/22	17/1/22
Phase 2 System Design			
Narrow scope	100%	18/1/22	19/1/22
Literature review - Baseline system and current approaches	100%	18/1/22	20/1/22
System and project design	100%	18/1/22	20/1/22
System Design Review	100%	21/1/22	21/1/22
Phase 3 Preliminary Design			
High level implementation strategy	100%	22/1/22	27/1/22
Stakeholders, use cases, requirements and risks	100%	22/1/22	6/2/22
Data collection	100%	22/1/22	15/2/22
ML research and development - Baseline models	100%	27/1/22	15/2/22
Hardware selection and system integration	100%	30/1/22	15/2/22
Preliminary Design Review	100%	16/2/22	16/2/22
Phase 4 Critical Design Review			
Finalise data pipelines	100%	17/2/22	3/3/22
ML research and development - Final models and performance tuning	100%	17/2/22	5/3/22
Hardware implentation	100%	4/3/22	9/3/22
System validation	100%	9/3/22	17/3/22
Report write-up	100%	17/2/22	17/3/22
Critical Design Review	100%	18/3/22	18/3/22
Phase 4 Final Review			
Incorporation of feedback from critical review / improvements from system validation	100%	19/3/22	22/3/22
Report write-up	95%	19/3/22	7/4/22
Final presentation	100%	8/4/22	8/4/22
Contingency	50%	7/4/22	10/4/22
Report submission	0%	11/4/22	11/4/22

Appendix D – Risk Register



Risk Assessment Register

Project Name: Detection of Alcohol Intoxication in Airports Using Pose Estimation. **Updated by:** Xavier Ikuenobe **Date Last Updated:** 06 April 2022

			Inherent Risk (without controls)					Residual Risk (with controls)		
S/N	**Risk Description**	**Caused by & Consequences**	**Likelihood**	**Consequence**	**Risk Rating**	**Control(s)**	**Control Owner(s) Name and Role**	**Residual Likelihood**	**Residual Consequence**	**Residual Risk Rating**
1	Limited training data	Caused by: Unavailability of datasets online Consequences: Poorly trained algorithm with subpar performance	High	Medium	Severe	Use an auto-encoder (trained on normal videos). Create additional datasets using real-world video recordings.	Priyanka Sarkar Data & Research Lead	Medium	Medium	Moderate
2	Occlusion	Caused by: Person(s)/object(s) obscuring keypoints of person(s) in camera's view. Consequences: Poor algorithm performance	High	Medium	Severe	Minimal occlusion due to implementation in streamlined areas such as security checkpoint.	Danuka Theja Wickramasinghe ML Lead	Very low	Medium	Sustainable
3	Human pose variations	Caused by: Passenger activity or state Consequences:	Medium	Medium	Moderate	Implement in predictable environment such as security checkpoint (passenger expected to walk).	Saurabh Bharadwaj ML Ops / Embedded Systems	Very low	Medium	Sustainable
4	Cost over budget	Caused by: Purchasing expensive components/subscriptions Consequences: Bear difference in cost above budget.	Low	High	Severe	Off the shelf components (Jetson Nano, web camera and AWS Subscription).	Deepak Kovaichelvan Project Manager	Very low	High	Moderate
5	Late components procurement	Caused by: Late purchase initiation or procurement of long-lead item(s) Consequences: procured item(s) arrive too late for use in project	Medium	High	Severe	Off the shelf components (next day delivery).	Deepak Kovaichelvan Project Manager	Very low	High	Moderate
6	Defective components	Caused by: Manufacturer or supplier Consequences: Malfunction or failure of system	Low	High	Severe	Test component once received in case it needs to be replaced.	Danuka Theja Wickramasinghe ML Lead	Very low	High	Moderate
7	Late algorithm completion	Caused by: Slow speed or poor planning and monitoring Consequences: Failure to complete the project	Medium	Very High	Critical	Track and measure actual against planned progress.	Xavier Ikuenobe Systems Lead	Very low	Very High	Severe
8	False positives	Caused by: Algorithm performance limitations Consequences: Overestimating the number of intoxicated passengers occurrence.	Very High	Medium	Severe	Target high precision (≥ 0.8).	Saurabh Bharadwaj ML Ops / Embedded Systems	Medium	Medium	Moderate
9	Cyber security	Caused by: Cyber attack Consequences: system failure, disruption of computer networks, information theft, etc.	Medium	Medium	Moderate	AWS security protocols	Yousuf Shaikh Validation Lead	Very low	Very High	Severe