EMERGENCY ROUTING SYSTEM FOR BUILDINGS USING DIJKSTRA'S ALGORITHM

KEE CHIN YEW

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

EMERGENCY ROUTING SYSTEM FOR BUILDINGS USING DIJKSTRA’S ALGORITHM

KEE CHIN YEW

This report is submitted in partial fulfilment of the requirements for the Bachelor of Computer Science (Artificial Intelligent) with Honours.

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2024

DECLARATION

I hereby declare that this project report entitled

Emergency Routing System for Buildings Using Dijkstra's Algorithm

is written by me and is my own effort and that no part has been plagiarized

without citations.

STUDENT : ______________________ Date : 30/08/2024

(KEE CHIN YEW)

I hereby declare that I have read this project report and found

this project report is sufficient in term of the scope and quality for the award of

Bachelor of [Computer Science (Artificial Intelligent)] with Honours.

SUPERVISOR : ______________________ Date : 30/08/2024

(Profesor Madya Dr. Asmala bin Ahmad)

DEDICATION

This final year report is dedicated to my beloved family, whose unwavering support and encouragement have been my foundation throughout this journey. To my parents, for their endless sacrifices, unconditional love, and for always believing in me. Your guidance and support have been my greatest strength. To my siblings, for their constant motivation, encouragement, and belief in my capabilities.

To my esteemed lecturers and mentors at University Technical Malaysia Malacca, whose wisdom, guidance, and patience have been instrumental in my academic and personal growth. Your dedication to teaching and your support have profoundly shaped my educational experience, and I am deeply grateful for the knowledge and mentorship you have provided.

To my friends and classmates, for their camaraderie, support, and the memorable moments we shared. Your friendship and encouragement have made this journey both enriching and enjoyable. The collaborative spirit and shared experiences have been highlights of my university life, and I cherish the bonds we have formed.

To my extended family and loved ones, for their prayers, encouragement, and belief in my potential. Your presence and support have been a source of strength and inspiration.

Finally, to everyone who has contributed to my academic journey in any way, no matter how small, thank you. Your contributions have made a significant impact, and I am grateful for your support and encouragement.

Thank you all for being an integral part of this significant chapter in my life.

ACKNOWLEDGEMENTS

I would like to express my heartfelt gratitude to my project supervisor, Professor Madya Gs. Dr. Asmala Ahmad, for his unwavering guidance, support, and expertise throughout the entire duration of this project. His insightful feedback and constructive criticism have been invaluable in shaping the direction and quality of my work. I am immensely thankful to my parents for their endless love, encouragement, and unwavering belief in my abilities. Their constant support and sacrifices have been the foundation of my academic journey, and I am forever grateful for their presence in my life. I extend my appreciation to the faculty members of University Technical Malaysia Malacca for their continuous support and for providing me with an enriching learning environment. Their knowledge and dedication have played a significant role in my academic growth. My sincere thanks go to my friends and classmates for their camaraderie, brainstorming sessions, and encouragement throughout this project. Their valuable input and discussions have contributed to the refinement of my ideas and the implementation of my project. Lastly, I would like to express my gratitude to all the participants who willingly volunteered their time and provided valuable data for my research. Without their involvement, this project would not have been possible. I am deeply grateful to all individuals and organizations mentioned above, as well as any others who have contributed in any way to the completion of this project. Their collective support has been invaluable, and I am truly fortunate to have had their assistance.

ABSTRACT

This project aims to develop an Emergency Routing System for buildings using Dijkstra's Algorithm to enhance the effectiveness of emergency evacuations. The primary objectives include assessing existing systems, developing an optimized routing algorithm, and evaluating its performance statistically. The system calculates the shortest and safest evacuation routes, minimizing evacuation time and congestion. Utilizing the Spiral Model, the project undergoes iterative cycles of planning, risk analysis, engineering, and evaluation for continuous refinement and risk management. Building plans are analysed using AnyLogic for simulation, providing accurate data for the routing algorithm. Potential risks like data accuracy and system failures are mitigated during the risk analysis phase. The engineering phase involves iterative design, prototyping, coding, and testing, ensuring system functionality and performance. The evaluation phase reviews progress, gathers stakeholder feedback, and assesses key metrics such as response time, route accuracy, system uptime, and user satisfaction. The proposed solution includes a human detection model using a pretrained TensorFlow API and an evacuation time estimation model based on Convolutional Neural Networks. Dijkstra's Algorithm calculates the best evacuation route considering predicted times rather than distances.

Testing against algorithms like Bellman-Ford and Floyd-Warshall shows that the system improves evacuation times and accuracy while ensuring reliability. The system's advantages include real-time updates, adaptability to dynamic scenarios, and higher user satisfaction compared to traditional solutions. Future developments may include integration with smart building technologies and adaptation for larger-scale environments, enhancing its applicability in diverse emergency scenarios. This project aims to provide a reliable and efficient emergency routing system that improves evacuation success rates under varied conditions.

ABSTRAK

Projek ini bertujuan untuk membangunkan Sistem Penghalaan Kecemasan untuk bangunan menggunakan Dijkstra’s Algorithm bagi meningkatkan keberkesanan pemindahan kecemasan. Objektif utama termasuk menilai sistem sedia ada, membangunkan algoritma penghalaan yang dioptimumkan, dan menilai prestasinya secara statistik. Sistem ini mengira laluan pemindahan yang paling pendek dan paling selamat, meminimumkan masa pemindahan dan kesesakan. Menggunakan Model Spiral, projek ini melalui kitaran perancangan, analisis risiko, kejuruteraan, dan penilaian secara berulang untuk penambahbaikan dan pengurusan risiko yang berterusan. Pelan bangunan dianalisis menggunakan AnyLogic untuk simulasi, menyediakan data yang tepat untuk algoritma penghalaan. Risiko berpotensi seperti ketepatan data dan kegagalan sistem diatasi semasa fasa analisis risiko. Fasa kejuruteraan melibatkan reka bentuk berulang, prototaip, pengkodan, dan ujian, memastikan fungsi dan prestasi sistem. Fasa penilaian mengkaji kemajuan, mengumpul maklum balas pihak berkepentingan, dan menilai metrik utama seperti masa respons, ketepatan laluan, masa operasi sistem, dan tahap kepuasan pengguna. Penyelesaian yang dicadangkan termasuk model pengesanan manusia menggunakan TensorFlow API pralatih dan model anggaran masa pemindahan berdasarkan Convolutional Neural Networks. Dijkstra’s Algorithm mengira laluan pemindahan terbaik dengan mempertimbangkan masa yang diramalkan dan bukannya jarak.

TABLE OF CONTENTS

PAGE

LIST OF TABLES

PAGE

LIST OF FIGURES

PAGE

LIST OF ABBREVIATIONS

FYP	**-**	**Final Year Project**
ACO	**-**	**Ant Colony Optimization**
API	**-**	**Application Programming Interface**
CNN	**-**	**Convolutional Neural Network**
GA	**-**	**Genetic Algorithm**
MAE	**-**	**Mean Absolute Error**

LIST OF ATTACHMENTS

PAGE

CHAPTER 1: INTRODUCTION

1.1 Introduction

In urban environments, the safety of victims during emergencies is paramount, particularly in large buildings where efficient evacuation routes are critical. Traditional emergency routing systems often rely solely on static signage, which may not effectively guide victims, especially in dynamic emergency situations. Additionally, overcrowding and blocked routes can impede evacuation efforts, leading to delays and potential hazards.

The current emergency routing systems in many buildings lack the adaptability and real-time responsiveness needed to effectively guide victims to safety during dynamic emergency situations. Static signage and manual instructions may not adequately address changing conditions, such as blocked exits, hazardous areas, or shifting evacuation routes. As a result, victims may face confusion, delays, or even danger during evacuations. (Zhang et al., 2021)

Existing emergency routing systems often fail to effectively address crowd density during evacuations, leading to inefficient and potentially hazardous evacuation processes. These systems typically lack mechanisms to dynamically adjust evacuation routes based on real-time crowd density data, resulting in congestion and delays that compromise the safety of building victims. (Wong et al., 2017)

Current emergency evacuation systems often overlook the diverse needs of building victims, including considerations such as age, gender, body size, and mobility limitations. This oversight can lead to inadequate evacuation plans that fail to accommodate individuals with disabilities or those requiring specialized assistance. As a result, during emergency situations, vulnerable populations may face increased risks and challenges in safely evacuating buildings. (Hadzic et al. 2011)

1.2 Problem statement

Current emergency routing systems often fail to adequately accommodate the diverse needs of victims and adapt to dynamic evacuation scenarios. This results in inefficient evacuations and increased risks to individuals during emergencies. There is a critical need to identify and implement improvements that enhance the flexibility and responsiveness of these systems to ensure the safety and well-being of all evacuees. (Oyola et al., 2017)

During emergencies, determining the shortest and safest evacuation routes while minimizing evacuation time and congestion within buildings remains a significant challenge. Existing systems lack the capability to effectively balance these factors, potentially leading to delayed evacuations and increased hazards for individuals. It is essential to develop a robust method for calculating optimal evacuation routes that prioritize both speed and safety. (Deng et al., 2022)

Evaluating the performance of emergency routing systems that utilize algorithms like Dijkstra's is crucial to ensure their efficiency, reliability, and scalability in various emergency scenarios and building environments. However, there is a lack of comprehensive assessment frameworks to measure these attributes. This gap necessitates the development of thorough evaluation methodologies to benchmark and enhance the effectiveness of such routing systems. (Sabri et al., 2014)

1.3 Objectives

This project embarks on the following objectives:

a) To assess the effectiveness of existing emergency routing systems in accommodating diverse victim needs and adapting to dynamic evacuation scenarios.

b) To develop an emergency routing system for buildings using Dijkstra's Algorithm.

c) To evaluate the performance of the system using statistical methods.

1.4 Project Scope

Module to be developed:

1. Human Detection for detecting the existence of humans

2. Best Route Calculation to find out the shortest and safest path

3. Evacuation Time Estimation Model for evaluating the time used for the path

1.5 Project Contribution

a) The implementation of an Emergency Routing System utilizing Dijkstra's Algorithm significantly enhances building victims' safety during emergencies by providing optimized evacuation routes based on current conditions, ensuring swift and secure evacuation paths.

b) Leveraging Dijkstra's Algorithm enables the system to continuously analyse building layouts and environmental factors, dynamically adjusting evacuation routes in real-time to guide victims away from hazards and towards safe exits, thereby maximizing their safety.

c) Utilizing Dijkstra's Algorithm allows the system to calculate the shortest and safest evacuation paths, minimizing evacuation time and congestion, thus significantly improving overall efficiency during emergencies.

1.6 Report Organisation

This report is structured into six chapters, each addressing different aspects of the Emergency Routing System for Buildings Using Dijkstra’s Algorithm. Chapter 1 provides an overview of the problem’s background, objectives, and project scope. Chapter 2 includes a comprehensive literature review and problem review with solution. In Chapter 3, the report delves into the methodology employed for system development, project schedule, project workflow, and performance measurement for evaluation. Chapter 4 focuses on the proposed solution with the experiment design to get the result. Furthermore, Chapter 5 covers the results and the analysis and discussion based on the results. Lastly, Chapter 6 presents an assessment of the contributions and limitations of the Emergency Routing System for Buildings Using Dijkstra’s Algorithm, as well as the improvement in future.

1.7 Summary

In conclusion, this chapter introduces the critical need for efficient evacuation routes in buildings during emergencies, highlighting the limitations of traditional routing systems in dynamic situations These systems often fail to adapt to changing conditions, manage crowd density, or accommodate diverse victim needs, including those with disabilities. The chapter also outlines the problem statements focused on improving these systems, calculating optimal evacuation routes, and evaluating performance using Dijkstra's Algorithm. The project's objectives include assessing current systems, developing a new routing system, and evaluating its performance. The scope involves modules for human detection, route calculation, and evacuation time estimation. The contributions emphasize enhanced safety, real-time route adjustments, and efficient evacuations, with the report structured into six detailed chapters covering background, methodology, proposed solutions, results, and future improvements. Next chapter will be focus on the research of the system to ensure the system is successfully developed with the proposed solution.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

Before implementing the system, reviewing related projects and references is important for gathering ideas and constructing model architecture. The literature review and project methodology used to implement Emergency Routing System for Buildings Using Dijkstra's Algorithm will be elaborated in this chapter. A literature review is a critical summary and evaluation of existing research and scholarly works related to my project topic. It involves identifying, analysing, and synthesizing relevant literature to provide context, establish the current state of knowledge, and highlight areas for further research. Besides, problem review outlines the steps taken to achieve project objectives. It is important to guide the entire research process, ensuring that the project is solving the problems.

2.2 Related Work

For better understanding of the principles and methods used for implementing Emergency Routing System for Buildings Using Dijkstra's Algorithm, this section will present all the materials that have been gathered from various sources, such as journals, research papers, and books.

2.2.1 Domain

The related domains of Emergency Routing System for Buildings Using Dijkstra's Algorithm are listed out in this section.

2.2.1.1 Environment Update

For continuously update and gather environmental information to provide the best evacuation route, there is a variety of sophisticated methods and technologies. These mechanisms are the main components to collect data into the system, providing a comprehensive understanding of the surrounding environment in real-time. From traditional sensors to cutting-edge imaging technologies, each avenue offers unique advantages in capturing crucial data points and ensuring the accuracy and reliability of our environmental monitoring efforts.

Heat sensors and thermometers represent one such method, offering insights into temperature variations across different spaces. By strategically placing these sensors, the system can effectively monitor thermal dynamics and detect anomalies indicative of environmental changes or irregularities. This helps the system to detect whether the place is available for humans to pass through or not, ensure the route is safe and able to be followed by victims. (Dayana et al., 2020)

In parallel, camera detection systems equipped with advanced image recognition software serve as the eyes of the system, continuously scanning the environment for movements and changes in lighting conditions. This visual data not only enhances situational awareness but also provides valuable context for interpreting other sensor readings. By collecting the information from environment, the system can find out the victims from the building, provide suggestions based on the items at the place to increase evacuation success rate and predict accident based on the condition of environment such as building collapse. (Tsai et al., 2022)

Vibration sensors constitute another vital component, detecting mechanical vibrations that could signal seismic activity or structural issues. By promptly capturing these vibrations, we can assess potential risks and take proactive measures to mitigate them. Hence, the victims at bottom floor will be arranged to evacuate faster based on the condition before the scenario becomes worse due to the vibration. (Yamashita et al., 2020)

Sound sensors and microphones offer yet another avenue for environmental monitoring, capturing changes in sound levels that could indicate disturbances or anomalies. This auditory data provides additional layers of insight, complementing other sensor readings and enabling a more holistic understanding of the environment. By having the feature, the system can detect the victims that are under construction and provides a route for the rescue. (Tronstad et al., 2021)

Motion sensors contribute to security and situational awareness by detecting movement within the environment. Whether it is tracking the presence of individuals or monitoring wildlife activity, these sensors play a crucial role in detecting and responding to emerging threats or incidents. This is important when harmful living things is inside the building, which real-time provides the location of it and adjust the evacuation route that keep the victims away from it. Besides, the system can calculate the speed of evacuation based on the movement and adjust the plan to make the evacuation process more efficient. (Zhang et al., 2023)

In short, these methods provide environmental information to the system, enabling us to stay vigilant, informed, and proactive in our efforts to understand and respond to changes in the building.

2.2.1.2 Shortest Path Calculation

Based on the shortest path problem, we must use different algorithms to find out the shortest path as every algorithm has its own advantages and disadvantages. Applying suitable algorithms on specific problems can improve the efficiency of finding the shortest path. Here are two kinds of shortest path algorithms which are single-source shortest path algorithms and all-pairs shortest path algorithms.

1. Single-Source Shortest Path Algorithms:

Single-Source Shortest Path Algorithms aim to find the shortest path from a single source node to all other nodes in a graph. These algorithms explore the graph iteratively, updating the shortest path distances as they progress. One key characteristic of these algorithms is that they are designed to handle graphs with positive edge weights.

a. Dijkstra's Algorithm

Dijkstra's Algorithm works by maintaining a priority queue of nodes based on their tentative distances from the source node. It repeatedly selects the node with the shortest tentative distance, updates the distances to its neighbours, and relaxes the edges. Dijkstra's Algorithm guarantees finding the shortest paths in graphs with non-negative edge weights. (Navone, 2020)

b. Bellman-Ford Algorithm

Bellman-Ford Algorithm is another algorithm that can handle graphs with negative edge weights. It iteratively relaxes all edges in the graph for |V| - 1 iterations, where |V| is the number of vertices. This process ensures that the algorithm converges to the shortest paths, even in the presence of negative edge weights or cycles. (Upadhyay, 2023)

c. Ant Colony Optimization (ACO)

Ant Colony Optimization is a metaheuristic inspired by the foraging behaviour of ants. ACO involves simulating the movement of ants on the graph, which ants deposit pheromone trails on edges and then stronger trails indicating shorter paths. In the end, the pheromone trails guide subsequent ants to discover shorter paths, effectively finding the shortest path from the source node to other nodes in the graph. (Rezvanian et al., 2023)

d. A* Search Algorithm

A* Search Algorithm is a heuristic search algorithm that efficiently finds the shortest path from a starting node to a goal node in a graph. It uses a heuristic function to estimate the cost of reaching the goal from each node and combines this with the actual cost incurred to reach the node. By prioritizing nodes with lower estimated costs, A* Search Algorithm explores the graph to find the shortest path while ensuring optimality. (Geeksforgeeks, 2024)

2. All-Pairs Shortest Path Algorithms:

All-Pairs Shortest Path Algorithms aim to find the shortest paths between all pairs of nodes in a graph. These algorithms compute the shortest paths between every pair of nodes simultaneously, providing a comprehensive understanding of the shortest paths in the graph.

a. Floyd-Warshall Algorithm

Floyd-Warshall Algorithm is a dynamic programming-based algorithm that efficiently computes the shortest paths between all pairs of nodes in a graph. It iteratively considers all possible intermediate nodes and updates the shortest path distances between every pair of nodes. Floyd-Warshall Algorithm is suitable for graphs with both positive and negative edge weights, but it does not handle negative cycles. (Geeksforgeeks, 2024)

b. Genetic Algorithm (GA)

GA is a metaheuristic inspired by the process of natural selection and evolution. GA involves evolving a population of candidate solutions, where each solution represents a set of shortest paths between all pairs of nodes. Through processes such as selection, crossover, and mutation, GA iteratively improves the population towards optimal solutions, providing a practical approach to finding shortest paths between all pairs of nodes in a graph. (Mallawaarachchi, 2017)

2.2.1.3 Shortest Path Selection

Shortest path selection involves identifying the most efficient route between two points in a network or graph, aiming to minimize the total cost, such as distance or travel time. It's a fundamental problem in computer science and optimization, with applications in transportation, logistics, and urban planning. Algorithms like Dijkstra's Algorithm and Bellman-Ford Algorithm are commonly used to find the shortest path, considering factors like edge weights and user preferences. The goal is to determine the optimal sequence of edges or links to traverse from a starting point to a destination, meeting specific constraints or objectives while optimizing resource usage and travel efficiency. Here are some factors that can be considered while selecting the path.

1. Distance

The distance between nodes serves as a fundamental factor influencing path selection. Algorithms aim to minimize the overall distance travelled along the path to reduce travel time. In evacuation routing systems, algorithms prioritize routes that minimize distance to reduce the evacuation time and optimise the number of saved victims. (Pratiwi et al., 2020)

2. Travel Time

Travel time encompasses various factors such as density of people, walking speed, and type of path. Algorithms consider these factors to estimate the time required to traverse each path. Paths with shorter travel times are preferred as they minimize delays and enhance overall efficiency. For example, in GPS navigation systems, algorithms consider real-time traffic data to recommend routes with the shortest estimated travel times, helping users reach their destinations faster and avoid congested areas. (Astana et al., 2023)

3. Elevation or Slope

Changes in elevation or slope along the path significantly impact the effort required for traversal. Algorithms may prioritize paths with minimal elevation changes to minimize physical exertion, conserve energy, and enhance user comfort. For example, in hiking or biking trail applications, algorithms favour routes with gentle slopes or gradual inclines to provide a more enjoyable and manageable outdoor experience. (Zhu et al., 2022)

4. Crowdedness or Density

The level of crowdedness or density of people or obstacles along the path affects the ease and efficiency of travel. Algorithms consider crowd density to avoid congested areas and select routes with smoother traffic flow. Paths with lower crowdedness are favoured to minimize delays and reduce stress. For instance, in public transportation routing systems, algorithms prioritize less crowded routes to provide passengers with a more comfortable and pleasant travel experience. (Wong et al., 2017)

5. Safety or Security

Safety considerations such as rate of accident, dangerousness of the place and air condition play a crucial role in path selection. Algorithms prioritize paths perceived as safer to minimize the risk of accidents, incidents, or security threats. Paths with higher safety ratings are preferred, particularly in applications where user safety is warranted, such as going to lower level using stairs instead of elevator. Algorithms may incorporate safety metrics and user feedback to identify and recommend safer paths for navigation. (Deng et al., 2022)

6. Infrastructure Condition

The condition of infrastructure such as road quality, pavement condition, and maintenance status directly impact travel comfort and safety. Algorithms consider infrastructure condition to recommend routes with well-maintained roads. Paths with better infrastructure conditions are preferred to ensure smoother and more reliable travel experiences. For example, algorithms prioritize routes with high-quality infrastructure to optimize the evacuation efficiency and minimize the usage of energy. (Mahmudah et al., 2022)

7. Accessibility

Accessibility considerations are essential for ensuring inclusivity and accommodating individuals with disabilities or special needs. Algorithms prioritize paths with accessibility features such as ramps, elevators, and tactile paving to ensure everyone can pass through. Paths that meet accessibility standards are preferred to help individuals with mobility impairments evacuate in time. For example, algorithms prioritize accessible routes to allow victims of all abilities to escape the building safely and independently. (Ternero et al., 2023)

8. Environmental Factors

Environmental conditions such as temperature, air quality, and natural hazards significantly influence travel conditions and safety. Algorithms consider environmental factors to recommend routes that minimize exposure to adverse conditions and optimize travel experiences. Paths with suitable environmental conditions are chosen to enhance the survival rate of victims. For example, algorithms include air quality indices into the calculation to suggest routes with pleasant conditions for evacuation. By integrating environmental data, algorithms provide utilised recommendations that align with victims' comfort and safety. (Dayana et al., 2020)

9. Path Width

The width of the path directly impacts the ease of movement, flow of traffic, and potential for congestion. Algorithms consider path width to recommend routes that accommodate varying volumes of victims. Paths with wider width are preferred to facilitate smoother traffic flow, reduce congestion, and enhance victims’ safety. For example, algorithms prioritize routes with wider lanes to optimize victims' mobility and ensure efficient evacuation. (Robbani et al., 2018)

2.3 Critical Review of Current Problem and Justification

2.3.1 Human Detection

For real time updating the shortest route, human detection is needed to find out the humans in the building. The system can be based on the result to find out the most suitable route for the persons from the location to the exit.

Astana et al. (2023) and Permana et al. (2019) used human density as their model input to find out the evacuation time of the route and adjust the route to reduce the time. Haghpanah et al. (2021), Mahmudah et al. (2022) and Zhang et al. (2023) are taking the location of the patients through detection and finding the shortest route that has the least people for evacuating the patients. Besides, Zhang et al. (2021), Zhu et al. (2022) and Han et al. (2021) separate the victims into different route based on the number of the victims at every point. This is to optimise the usage of the route to prevent every victim stuck at the same place and prevent the evacuation points are overloaded with victims.

Wong el al. (2017) proves that the number of victims, direction of evacuation, splitting or merging of crowd affect the effectiveness of evacuation. From the research, the human detection is required for real time adjusting the escape route based on the situation. However, the technique of controlling the way of victims evacuate to reduce the evacuation time is not considered, because it needs the victims fully following the instructions. (Zhang et al., 2021; Haghpanah et al., 2021; Mahmudah et al., 2022; Zhang et al., 2023; Zhu et al., 2022; Han et al., 2021) Since the action of victims is not controllable, including the movement of victims may causing unreliable results.

2.3.2 Best Route Calculation

Finding the best route is challenging since the best route can be defined based on different aspects such as the safest, fastest, shortest, easiest and others. Besides, different aspects also need different variables for calculating the weight of the aspect.

Robbani et al. (2018) applied fuzzy to find out the weight of the route using the distance, road density and refugees of the route, and used Dijkstra's Algorithm to find out the best route with the lowest weight. Rahayuda et al. (2021) used Dijkstra's Algorithm and Bidirectional Dijkstra's Algorithm to find out the best route by using the distance. Han et al. (2020) designed a way to reduce the time of finding best route in multi-exit building, which is assigning the nearest exit to the location and find the best route using Dijkstra's Algorithm based on the time. Astana et al. (2023) uses Dijkstra's Algorithm to find out the best route based on the time. Oyola et al. (2017) used reversed Dijkstra's Algorithm to find out the best route in dynamic environments using the distance. Pratiwi et al. (2020) used Dijkstra's Algorithm to find out the best route using the coordinate of the location. Lu et al. (2005) implemented a Dijkstra's Algorithm based algorithm to limit the group of evacuees based on the best route to reduce the evacuation time. Patel et al. (2016) tried to find out the more accurate best route using Dijkstra's Algorithm by considering the edge-blocking, evacuation-order-dependency and redundant-path-update. Zhang et al. (2021) used Dijkstra's Algorithm to find out the best route based on the weight and used Best First Search (BFS) to adjust the route while the point in the route suddenly cannot passed through. Sabri et al. (2015) utilised Dijkstra's Algorithm based on the distance using AI technique which are Ant Colony Optimisation (ACO) and Particle Swarm Optimisation (PSO) in high and close building.

Dayana et al. (2020) implemented a PI system with sensors such as smoke sensor, heat sensor and water level sensor to find out the best route in building using Bellman Ford’s Algorithm. Permana et al. (2019) find out the best route using Floyd-Warshall Algorithm based on the population at the area and the distance of route. Haghpanah et al. (2021) used Floyd-Warshall Algorithm to find out the best route by considering the priority of victims, the movement speed of victims and the condition of the victims. Mahmudah et al. (2022) used coordinates to find out the distance of points and took it as the input of Floyd-Warshall Algorithm to find out the best route. Zhang et al. (2023) compared three algorithms which are Dijkstra's Algorithm, Genetic Algorithm and ACO based on the distance to find out the best route that separates the covid patients and victims while escaping, Zhu et al. (2022) optimised Dijkstra's Algorithm to find out the best route based on the sea level, distance and the gradient of the route and the shelter options. Han et al. (2021) used Dijkstra's Algorithm to separate the zones in the building into groups, then used Partitioned and Staged Evacuation Planning (PSEP) to find out the best route for every point to schedule their evacuation order and reduce the evacuation time. Deng et al. (2022) proposed a method that using prediction model to predict the fire in the building and adjust the route plan that obtained from A* Algorithm based on the distance and the result of the prediction model.

Based on the research I read, Dijkstra's Algorithm will be the best algorithm in this project. Dijkstra's Algorithm has the shortest processing time to find out the best route, which is important in a real time environment that the route plan may need to be changed frequently. Dijkstra's Algorithm also will be implemented reversely for shortening the evacuation time. (Oyola et al., 2017) The distance, width and number of victims of the route also will be considered in the algorithm for finding the best route, these variables will be the input of prediction model to provide the route weight which is the evacuation time for the algorithm. (Han et al., 2020; Robbani et al., 2018; Astana et al., 2023; Zhang et al., 2021; Deng et al., 2022) Besides, the route will real time updated based on the input data by using Dijkstra's Algorithm because BFS only considering change of a point instead of the whole building, hence the plan can be inaccurate if other groups change the route too at the same time. (Zhang et al., 2021)

2.3.3 Evacuation Time Estimation Model

For finding the best route in the building, various variables have to be considered to determine the weight of the route. However, the relationship between the variables is difficult to discovered. Hence, the AI prediction model is required to predict the evacuation time of the route based on the variables collected.

Robbani et al. (2018) used Sugeno Fuzzy Inference System to determine the weight of the route. The dataset of the routes is collected to create the membership function to determine the variable level of the route, then find out the weight of the route based on the Sugeno’s Inference Rules. Deng et al. (2022) used PyroSim for fire simulation, to predict the flow of fire by analysing the smoke movement, the temperature, and the toxic gas concentration in the fires. This helps the system to real time adjust the route away from the fire based on the prediction.

From the research, AI prediction model is possible to help in planning the evacuation route. The dataset of the route will be collected for training the prediction model, to assign the weight of the route which is the evacuation time of the route. Hence, the accuracy of finding the best route will be increased because more variables are considered on the route.

2.3.4 Summary

Table shows the summary of the related works in terms of human detection and shortest paths algorithm, with their performance analysis.

Table 2.1: Research Summary

Journal	Summary	Method	Performance Analysis
Khalid Shibghatulloh Robbani, Wiranto, & Esti Suryani, Fuzzy-Dijkstra Algorithm Implementation on Determining Logistic Distribution Route for Evacuation Post of Merapi Eruption Victims, 2018	This study focuses on using Fuzzy Logic and algorithm to determine the most efficient route for evacuation post of Merapi eruption victims.	Sugeno Fuzzy Inference System for Fuzzy Logic and Dijkstra's Algorithm for searching shortest path.	The results show that combination of Fuzzy Logic and algorithm can find out the most efficient route based on the time than using manuals with approximate distance.
I G S Rahayuda & N P L Santiar, Dijkstra and Bidirectional Dijkstra on Determining Evacuation Routes, 2021	The testing was carried out to compare the performance of Dijkstra's Algorithm and Bidirectional Dijkstra's Algorithm.	Normal Dijkstra's Algorithm is used while Bidirectional Dijkstra's Algorithm is using Dijkstra's Algorithm for the start and the end of the route separately, then combine two results become a best route.	In ten testing cases, five cases show that Dijkstra's Algorithm and Bidirectional Dijkstra's Algorithm get same results, four cases show that Bidirectional Dijkstra's Algorithm is better, and one case shows that Dijkstra's Algorithm is better.

Litao Han, Huan Guo, Haisi Zhang, Qiaoli Kong, Aiguo Zhang & Cheng Gong, An Efficient Staged Evacuation Planning Algorithm Applied to Multi-Exit Buildings, 2020	This study focuses on finding out the best way to generate the most efficient escape route plan.	Partitioned and staged evacuation planning (PSEP) algorithm is used to assign victims into groups and manage their escape order for better efficiency. Dijkstra's Algorithm also used to find the shortest path.	PSEP has the same performance as other algorithms while only one exit exists, however in multi-exit building, PSEP has better performance than other algorithms based on the evacuation time and operation efficiency.
I Nyoman Yudha Astana, I Dewa Ketut Sudarsana & Ni Made Widya Puspa, Designing Evacuation Route Using DIJKSTRA Methods, 2023	The primary objective of this study is to design evacuation route during disaster.	The algorithm used to design the route is Dijkstra's Algorithm based on the density of people on the path to find out the best route.	The algorithm considered the density of people on the path to include the delayed time in calculation for getting more accurate escape time. This helps the algorithm to choose the best route with shortest escape time.
Angely Oyola, Dennis G. Romero, & Boris X. Vintimilla, A Dijkstra-based algorithm for selecting the Shortest-Safe	This paper investigates the algorithms for finding shortest safe path in dynamic environment.	Dijkstra's Algorithm based Shortest-Safe Evacuation Routes (SSER) is used to determine the shortest path, and the weight of the vertex in graph is adjusted	The results show that SSER does not have the shortest escape time for selected path, but the path selected is safer compared to other shorter paths.

Evacuation Routes in dynamic environments (SSER), 2017		based on the safeness of environment.	
A F Pratiwi, S D Riyanto, R Listyaningrum, & G M Aji, The Shortest Path Finder for Tsunami Evacuation Strategy using Dijkstra Algorithm, 2020	This paper presents a comprehensive examination of obtaining shortest path of tsunami disaster evacuation in Cilacap Regency.	The paper develops Dijkstra algorithm in Matlab to get better presentation of the route.	The results show the shortest path finder can obtain the shortest path successfully.
Qingsong Lu, Betsy George, & Shashi Shekhar, Capacity Constrained Routing Algorithms for Evacuation Planning: A Summary of Results, 2005	The aim of this research is to reduce the evacuation time and cost of planning evacuation.	A hybrid algorithm, Capacity Constrained Route Planner (CCRP) is proposed. This algorithm applies limitation of edge capacities and edge travel time in Dijkstra's Algorithm.	The solution quality of CCRP is almost same with MRCCP and NETFLO but the run time of CCRP is half of MRCCP and lesser than one thrid of NETFLO, which greatly reduced the cost of algorithm.
Nishaben Patel, Manki Min, & Sunho Lim, Accurate Evacuation Route Planning Using	This paper focuses on implementing a new algorithm based on Static Multiple Path (SMP) by	A hybrid algorithm, Forward-Backward Shortest Path (FBSP) is designed based on SMP which solved the edge-blocking, evacuation order	FBSP has well performance on evacuation time compared to SMP, but its execution time is lower than SMP while there is large

Forward-Backward Shortest Paths. 2016	solving issues found in the algorithm	dependency, and redundant path update issues.	number of victims only. Hence, FBSP is better on handling large size input.
Huajun Zhang, Qin Zhao, Zihui Cheng, Linfan Liu, & 1and Yixin Su, Dynamic Path Optimization with Real-Time Information for Emergency Evacuation, 2021	This study proposed the way to optimise the shortest path searching with real time information.	Dijkstra's Algorithm is used for obtaining prior evacuation paths and Breadth First Search (BFS) is used to update the paths based on the real time information.	The results show the searching time for shortest path can be 2% to 92% shorter when using BFS to update the path instead of using Dijkstra's Algorithm to search again the whole model based on the information.
Nor Amalina Mohd Sabri, Abd Samad Hasan Basari, Burairah Husin, Khyrina Airin Fariza Abu Samah, The Utilisation of Dijkstra's Algorithm to Assist Evacuation Route in Higher and Close Building, 2015	This research presents the way to create layout plan and use the algorithm to label out the evacuation path on the plan.	AutoCAD software is used to convert real layout plan into blueprint layout, then uses MATLAB to create visibility graph and implement Dijkstra's Algorithm for finding the shortest path.	The results show a clear route on the layout plan which provides a global view to the whole evacuation plan for better visualization.
Baby D. Dayana, Shiv Pratap Singh, Shraman Das, Pankaj Gautam, Emergency	This research implements a model with modules and	Raspberry Pi acts as the main component of the model to receive all data and	Bellman Ford's Algorithm has longer run time compared to Dijkstra's

Escape Routing and Evacuation using Bellman Ford's Algorithm, 2020	sensors to determine the best escape route.	applies Bellman Ford's Algorithm on finding the best escape route from weighted graph.	Algorithm. However, Bellman Ford's Algorithm is used because Dijkstra's Algorithm cannot detect negative cycle in the graph, hence Bellman Ford's Algorithm works better for the model.
Ritesh Bhat, P. Krishnanda Rao, C. Raghavendra Kamath, Vipin Tandon & Prashant Vizzapu, Comparative analysis of Bellman-Ford and Dijkstra's algorithms for optimal evacuation route planning in multi-floor buildings, 2024	The testing was carried out to compare Bellman-Ford's Algorithm and Dijkstra's Algorithm based on the evacuation time, path efficiency, scalability, adaptability, and computational efficiency.	A building with multi-floor and multi-exit is used to test the performance of two algorithms. The nodes also weighted with positive and negative value to evaluate the benefits of the route.	Bellman-Ford's Algorithm is performing better than Dijkstra's Algorithm after the nineth second on all evaluated factors. Since the negative weight exists in the graph, hence Bellman-Ford's Algorithm greatly optimise the evacuation route planning compared to Dijkstra's Algorithm.
D Permana, F Rahmadani & Y Rizal, A shortest path problem for tsunami	This paper investigates the performance of algorithm to solve	Floyd-Warshall's Algorithm is used for planning the shortest	The travel time for all evacuation paths is lesser than 30 minutes, hence

evacuation in Padang City using Floyd-Warshall algorithm, 2019	the shortest path problem in Padang City.	path for tsunami evacuation.	Floyd-Warshall's Algorithm can be used for obtaining the evacuation paths to escape from tsunami.
Fardad Haghpanah, Kimia Ghobadi & Benjamin W. Schafer, Multi-hazard hospital evacuation planning during disease outbreaks using agent-based modeling, 2021	The aim of this research is to evacuate patients based on the patients' conditions to prevent collision and manage the assistants effectively during the time that COVID-19 is spread.	Agent-based model is developed for planning evacuation, which includes patient classification, path planning and collision avoidance. Floyd-Warshall's Algorithm and the predictive collision avoidance model developed by Karamouzas is used for this model.	The evacuation simulation using the developed model requires 26-50 minutes to evacuate all patients including COVID patients. However, the evacuation time can be vary based on the hospital policies, hence there is some improvement can be done for higher efficiency
Haniah Mahmudah, M. Fajar Ibrahim, Okkie Puspitorini, Ari Wijayanti & Nur Adi Siswandari, Floyd-Warshall Application for the Shortest Route Search in a Traffic Accident	This paper focuses on developing an application that sends notification and finds the shortest path to lead the rescuers to the traffic accident scene to provide help in time.	The longitude and latitude of the points in path is obtained for Floyd-Warshall's Algorithm to calculate and find out the shortest path from available paths.	The functions of the application work well in testing and the internet connection of the application is smooth. Floyd-Warshall's Algorithm also able to select the shortest

Notification App, 2022			path from the obtained paths.
Xinli Zhang, Yu Wang, Renjie Du, Yuan Guo & Abdullah AL Mamun, Evacuation Path of Patients with Infectious Disease based on Three Algorithms, 2023	This study presents a comprehensive examination on the performance of three algorithms on finding evacuation path of patients based on the path distance, number of people flow during evacuation and run time.	The building layout is presented as grid model for the algorithms, Genetic Algorithm (GA), Ant Colony Algorithm (ACO) and Dijkstra's Algorithm are used for finding the evacuation path.	The comparisons show that GA found the shortest path, ACO has the lowest evacuation flow and Dijkstra has run time that is about 235 times faster than the other two algorithms. Dijkstra's Algorithm may be useful for emergency, but it cannot find solution from the model sometimes.
Yang Zhu, Hong Li, Zhenhao Wang, Qihang Li, Zhan Dou, Wei Xie, Zhongrong Zhang, Renjie Wang & Wen Nie, Optimal Evacuation Route Planning of Urban Personnel at Different Risk Levels of Flood Disasters Based on the Improved 3D	The aim of this research is to improve the performance of algorithm by considering 3D factors while calculating the shortest evacuation path.	Google Map is used to obtain the geographical data of the research area, including interview for detailed information. Dijkstra's Algorithm is used for calculating the shortest path based on the data such as the pedestrian speed, slope, road width,	The results show a 1-6% of improvement rate by comparing 2D Dijkstra's Algorithm and 3D Dijkstra's Algorithm. 3D Dijkstra's Algorithm obtains shorter evacuation time because of more factors are included for more accurate calculation,

Dijkstra's Algorithm, 2022		pedestrian density, shelter occupancy and distance, and other factors.	improving the efficiency of planning evacuation route.
Litao Han, Cheng Gong, Lei Gu, Hu Qiao, Aiguo Zhang & Mengfan Liu, A Multi-Zone Staged Indoor Emergency Evacuation Algorithm Based on Time Equalization, 2021	This paper proposed a modified algorithm to improve the efficiency of planning the evacuation route by separating the evacuees into groups.	The partitioned and staged evacuation planning (PSEP) algorithm, the distance-based staged algorithm, and proposed algorithm which is modified from PSEP are compared on separating the evacuees into groups and declaring the zone in the building for evacuation. Then, Dijkstra's Algorithm is used for finding out the shortest path based on the zone.	The test is carried out for single-exit building and multi-exit building to investigate the importance of declaring zone for evacuation. The evacuation time for the three grouping methods has the same performance even changing the capacity of the exit in single-exit building. However, in the multi-exit building, the proposed algorithm has the lowest total evacuating time for every zone, but it has slightly greater average path length, compared to the distance-based staged algorithm which has the lowest

			average path length. The proposed algorithm also has the lowest total evacuation time while the evacuation density is increased. Hence, the proposed algorithm greatly improves the performance of PSEP and better than distance-based staged algorithm.
Sai-Keung Wong, Yu-Shuen Wang, Pao-Kun Tang & Tsung-Yu Tsai, Optimized evacuation route based on crowd simulation, 2017	This research presents an approach to optimize evacuation route planning based on the crowd simulation.	Crowd motions, collision prevention, evacuation direction and intersection of route are considered for simulating crowd, then Dijkstra's Algorithm is used for finding out the shortest path and adjusted based on the crowd.	The results show that the proposed approach successfully evacuated all agents in the shortest time compared to other algorithms. Besides, under the multi-exit environment, the proposed approach shows better performance on evacuating all agents in short time. However, the proposed approach is not perfect due to the

			psychological issues such as emotion of human, selfishness, and more.
Kunxiang Deng, Qingyong Zhang, Hang Zhang, Peng Xiao & Jiahua Chen, Optimal Emergency Evacuation Route Planning Model Based on Fire Prediction Data, 2022	This paper proposed an improved algorithm by collecting environment information to find out the best route for evacuation.	Fire Dynamics Simulator (FDS) is implemented to provide information to algorithms to update the condition of nodes. A* Algorithm also used as the algorithm to combine with FDS and find out the shortest path.	Based on the results shown, the proposed method found a longer evacuation path, but greatly lesser computational time compared to the traditional A* Algorithm. Hence, the evacuation planning is optimized and safer, and performing better in large-scale evacuation.

2.4 Summary

In short, this chapter focuses on reviewing existing research and methodologies crucial for developing an Emergency Routing System for Buildings Using Dijkstra's Algorithm. The literature review synthesizes current knowledge, highlighting the importance of environmental monitoring through various sensors and exploring diverse shortest path algorithms like Dijkstra's, Bellman-Ford, ACO and other. Path selection factors such as distance, travel time, safety, environmental conditions and other are discussed in relation to optimizing evacuation routes. Additionally, the critical review emphasizes the significance of real-time human detection for route adaptation, underscores Dijkstra's Algorithm as optimal for real-time route calculations, and underscores the role of AI prediction models in enhancing evacuation planning accuracy. Next chapter is about the methodology and the milestones of the project for presenting the workflow of the system development.

CHAPTER 3: PROJECT METHODOLOGY

3.1 Introduction

This chapter presents a detailed explanation of the methodology chosen for this research project. It outlines the phases of the research workflow and the activities done in every phase to achieve the research objectives. The chapter also includes the description of the data collection processes, the tools and techniques used, the milestones, and the performance measurement and evaluation metrics.

3.2 Operational Framework/ Research Workflow

The methodology model used for this project is Spiral Model. The Spiral Model is a software development approach that emphasizes iterative refinement and risk management. It breaks the development process into cycles, each consisting of planning, risk analysis, engineering, and evaluation phases. This iterative approach allows for continuous improvement of the system, making it well-suited for Emergency Routing System for Buildings Using Dijkstra's Algorithm with high levels of uncertainty and complexity.

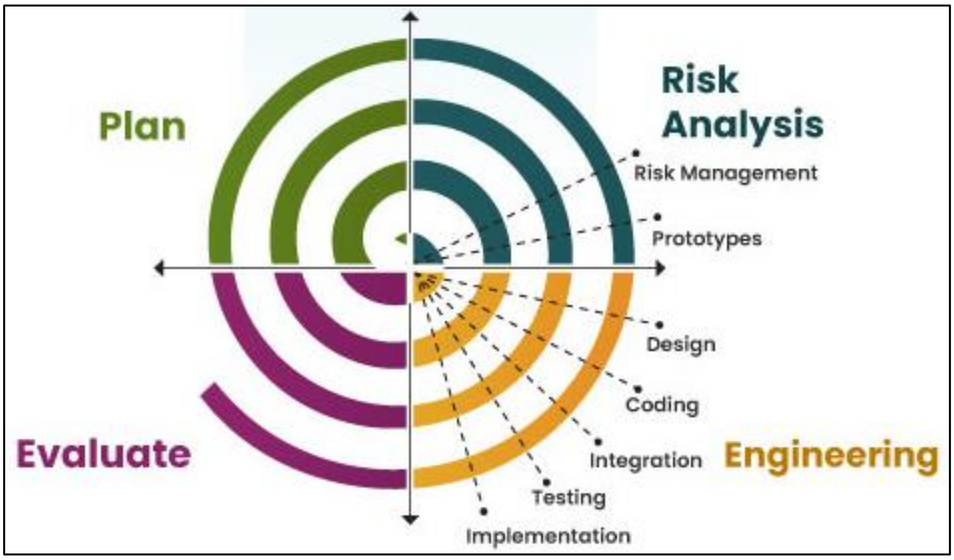

Figure 3.3.1: Spiral Model (Sumaiya Simran 2023)

3.2.1 Planning Phase

The primary objective during this phase is to define the project's scope, objectives, and requirements in detail. The project aims to ensure the safe and efficient evacuation of affected areas during emergencies, with a focus on minimizing congestion and providing real-time updates to both emergency responders and the public. The necessary data inputs are collected and analysed to provide necessary variables to the routing algorithms and decision-making processes of the system. Additionally, a preliminary project plan is established for outlining key milestones, resource requirements and timelines for keeping track of activities throughout every phase of the project.

The data are collected from the building plans created by me for getting more actual data for the buildings. However, these building plans are designed based on the actual building plan design for accuracy. Based on the route data, software AnyLogic is used for simulation to get the total time needed for specified number of people to go through the route. Pugh et al. (2021) and Niu et al. (2023) used the same software for their evacuation simulation. The results from the simulation will be used for training prediction model. Additionally, a comprehensive project plan is established to delineate key milestones, resource requirements, and timelines. This plan guides subsequent activities throughout the project's lifecycle, ensuring alignment with project goals and stakeholder expectations.

3.2.2 Risk Analysis Phase

The risk analysis phase is important for identifying and mitigating potential risks that could impact the implementation of the system. Various risk factors, technical challenges, operational uncertainties, and external dependencies are conducted in this phase. The risk may include unstable data accuracy that affected by incomplete dataset, scalability challenges associated with handling large volumes of real-time data during emergency situations, integration complexities with existing real-time data sources and communication networks, uncontrollable victims' behaviour during evacuation and the potential for system failures or downtime during critical evacuation events. These risk factors will be addressed and analysed during this phase to reduce the risk of implementing this project.

The successful implementation of the Emergency Routing System for Buildings Using Dijkstra's Algorithm hinges on effectively addressing key risks identified through rigorous analysis. The risks are the potential performance limitations of Dijkstra's Algorithm in complex building environments, concerns regarding the reliability and accuracy of data sourced from building plans and camera, and the challenge of managing an extended implementation timeline. To mitigate these risks, the project employs strategic approaches tailored to each concern. Modifications to Dijkstra's Algorithm are being explored to optimize its performance within diverse building layouts, ensuring efficient and accurate evacuation route calculations. Rigorous validation through multiple simulations is planned to enhance data reliability, utilizing iterative testing to refine camera inputs and decision-making algorithms. Furthermore, efforts are underway to streamline implementation processes, employing agile methodologies and modular design principles to expedite system deployment and integration. Continuous monitoring and reassessment are integral to these strategies, ensuring that risk mitigation measures remain responsive to evolving project dynamics and effectively support the project's goals of enhancing emergency response capabilities in built environments.

3.2.3 Engineering Phase

This phase is characterized by iterative design, prototyping, coding, and testing activities, aims to progressively refine the system's functionality and performance. Detailed system specifications are designed, including the architectural design, database schema, and user interface layout. Prototyping played a crucial role in validating design decisions and gathering early feedback, which helps in iterating rapidly and incorporating change. The coding phase involved the implementation of core system components, such as the route optimization algorithms, real-time human detection module, fuzzy analysis module and user interface elements. Then, testing procedures that include unit testing, integration testing, and user acceptance testing will be conducted to validate the functionality, performance, and reliability of the system under various scenarios and use cases. Each iteration of development culminated in a tested and validated increment of the system for further refinement and enhancement in subsequent cycles.

The engineering phase begins with the detailed design of system modules using comprehensive flowcharts, which serve to intricately map out the operational processes of each module. Once these designs are finalized, the next step involves translating them into pseudocode, ensuring that the implementation accurately reflects the intended functionality and logic. The performance of the prediction and detection model will be evaluated using graph and metrics before integration. After the modules are successfully implemented, they are integrated into the larger framework of the Emergency Routing System for Buildings Using Dijkstra’s Algorithm. Extensive testing procedures are then carried out to assess the system's performance, focusing on functionality, reliability, and efficiency. This testing phase is iterative, with each cycle of implementation followed by thorough testing to identify and address any issues or improvements needed. This iterative approach not only ensures the robustness of the system but also allows for continuous refinement and enhancement to meet the project's objectives effectively.

3.2.4 Evaluation Phase

The evaluation phase reviews the project progress, assesses project performance, gathers feedback from stakeholders, and plans for future iterations and enhancements of Emergency Routing System for Buildings Using Dijkstra’s Algorithm. The objectives and milestones defined in the project plan are compared for identifying areas of success and opportunities for improvement in the next cycle. Besides, performance assessment involved the evaluation of key metrics such as response time, route accuracy, system uptime, and user satisfaction levels, providing valuable insights into the effectiveness and efficiency of the system in supporting emergency evacuation operations. The iterative nature of the Spiral Model ensured that the evaluation phase provides analysis that helps for continuous improvement and evolution of the system to makes the evacuation success under any situation.

The evaluation phase of the project involves a thorough comparative analysis where the system's performance will be rigorously assessed alongside two alternative algorithms. This comparative study aims to provide a comprehensive understanding of how well the system performs in various scenarios compared to existing solutions. In addition to comparative analysis, extensive testing protocols will be implemented to evaluate the system's scalability, processing speed, and its overall effectiveness in optimizing evacuation time across different routes. These tests will generate quantitative data on the average evacuation times for each route, which will be graphically represented to offer detailed visual insights into the system's operational efficiency and performance capabilities. Through these comprehensive evaluations, the project aims to validate the efficiency of the Emergency Routing System for Buildings Using Dijkstra’s Algorithm and identify areas for potential enhancement or optimization to ensure robust performance in real-world emergency situations.

3.3 Project Activities and Milestones

This section shows the project schedule and milestones for the development of the Emergency Route Planning System Using Dijkstra's Algorithm. Refer to Table 3.1 for the Gantt Chart Table 3.2 for the description of activities for each phase.

Table 3.1 Gantt Chart

PHASE	MONTHS						
	MARCH	APRIL	MAY	JUNE	JULY	AUGUST	SEPTEMBER
1. PLANNING							
1.1 Project proposal and discussion							
1.2 Project research and study							
1.3 Project requirements							
1.4 Data collection							
2. RISK ANALYSIS							
2.1 Define the scope							
2.2 Identify risks							
2.3 Risk assessment							
2.4 Risk evaluation							
2.5 Develop risk mitigation strategies							
3. ENGINEERING							
3.1 Develop detection modules							
3.2 Develop algorithm modules							
3.3 Integrate system modules							
3.4 Perform initial testing							
4. EVALUATION							
4.1 Report Documentation							
4.2 Project demonstration							
4.3 Final presentation							
4.4 Final report presentation							

Table 3.2 Description of the activities for each phase

No	Task	Action
1.0	Planning Phase	
1.1	Discuss and validate the proposed project title and project scope.	Student, Supervisor
1.2	Gather research papers related to the project and conduct an analysis.	Student
1.3	Identify and outline all the essential project requirements and methodologies.	Student
1.4	Gather and acquire data for the project.	Student
2.0	Risk Analysis Phase	

2.1	Define clearly the objectives and outlines of the risk analysis for the emergency routing system.	Student
2.2	Identify potential risks through brainstorming and historical data analysis.	Student
2.3	Evaluate the identified risks based on their impact using qualitative and quantitative methods.	Student
2.4	Determine and categorize the risk levels by comparing them against predefined criteria.	Student
2.5	Formulate strategies to mitigate identified risks, including avoidance, reduction, transfer, and acceptance approaches.	Student
3.0	Engineering Phase	
3.1	Develop modules: Route Planning System Human Detection System Evacuation Time Analysis	Student
3.2	Integrate the modules into a system.	Student
3.3	Preforming initial testing.	Student
4.0	Evaluation Phase	
4.1	Present the system to the supervisor to gather valuable feedback.	Student, Supervisor
4.2	Final presentation will be delivered to the evaluator and the supervisor at the end of the project.	Student, Supervisor, Evaluator
4.3	Prepare the final report and project documents for submission	Student

3.4 Performance Measurement/ Evaluation Metrics

The performance is measured on three different environments to show that the system able to give reliable route in different buildings under any situation. The environments are multi-exit building, multi-floor building and buildings with different area. Han et al. (2020), Oyola et al. (2017), Zhang et al. (2021) and Han et al. (2021) used multi-exit building as their testing building to determine their proposed solution performance on planning the victim groups to suitable evacuation point with minimum time. Astana et al. (2023), Sabri et al. (2015), Bhat et al. (2024) and Haghpanah et al. (2021) tested their solutions by using multi-floor building to determine the performance of finding the best route to the first floor and escape. Lu et al. (2005), Patel et al. (2016) and Zhu et al. (2022) tested their proposed solutions based on the number of nodes, which is also the area of the region to find out the performance due to the scalability.

For achieving the objectives, the metrics to be evaluated are the evacuation time, the solution processing time and the total evacuees. Han et al. (2020), Astana et al. (2023), Lu et al. (2005), Patel et al. (2016), Zhang et al. (2021), Haghpanah et al. (2021), Zhu et al. (2022), Wong el al. (2017), Deng et al. (2022) evaluated their solutions on the evacuation time. This is the main aspect that have to be evaluated because the main purpose of implementing the evacuation solution is to minimise the evacuation time. Oyola et al. (2017), Lu et al. (2005), Zhang et al. (2021), Dayana et al. (2020), Zhang et al. (2023), Deng et al. (2022) evaluated their solutions' processing time or running time. Short processing time is important when the solution is trying to change the route plan under unexpected situations. Evaluating the processing time or running time can helps to determine the solution is applicable for multiple running within a short time. Bhat et al. (2024), Haghpanah et al. (2021), Zhang et al. (2023), Han et al. (2021), Deng et al. (2022) evaluated their solutions on the number of evacuees. This is for determining the number of evacuees at different exits, the number of evacuees in a time period, the number of evacuees at different routes to make further improvements on the solutions and evaluate the evacuation speed that the solution improved.

3.5 Summary

In this chapter, the operational framework and methodology are detailed using the Spiral Model, emphasizing iterative refinement and risk management. The research workflow progresses through distinct phases which are Planning, Risk Analysis, Engineering, and Evaluation. Each phase is meticulously structured to achieve specific objectives, such as defining project scope and requirements, mitigating potential risks, iterative design and testing of system modules, and rigorous performance evaluation. Key activities include data collection, algorithm optimization, system integration, and comprehensive testing across varied building environments to assess evacuation efficiency. Performance metrics focus on evacuation time, processing speed, and scalability, ensuring robustness under diverse conditions. The chapter also presents a detailed Gantt chart outlining project activities and milestones, underscoring the systematic approach to developing the Emergency Routing System for Buildings Using Dijkstra’s Algorithm. Next chapter is about the fully description of the proposed method with the experiment environment.

CHAPTER 4: PROPOSED METHOD

4.1 Introduction

This chapter presents the proposed solution for solving the problems of this project. The proposed solution is briefly explained, including the architecture and the design to achieve the objectives effectively. Besides, this chapter outlines the experiment design, including the overall flow of the process, the simulation setup, and the comparisons of techniques used to validate the proposed method. This chapter ensures a clear understanding of how the solution is developed to work and its expected impact on the problems.

4.2 Proposed Solution

4.2.1 Human Detection

The human detection model chosen for this project is a pretrained object detection model available on GitHub, which has been trained using the Tensorflow Object Detection API by Dat Tran et al. (2017). This particular API is highly advantageous because it allows the model to be trained using a massive amount of data, significantly enhancing its capability to detect a wide variety of objects with greater accuracy. The pretrained model includes 90 distinct classes, one of which is the human class. However, for the purposes of this project, the model will be utilized specifically for the detection of humans within building environments.

The Tensorflow Object Detection API facilitates efficient and scalable training processes, making it possible to leverage large datasets to improve the model's accuracy and robustness. This extensive training allows the model to better generalize across different scenarios and environments, ensuring reliable detection performance. The pretrained model's inclusion of 90 classes demonstrates its versatility and comprehensive detection capabilities, but in this project, the focus will be on optimizing its use for human detection.

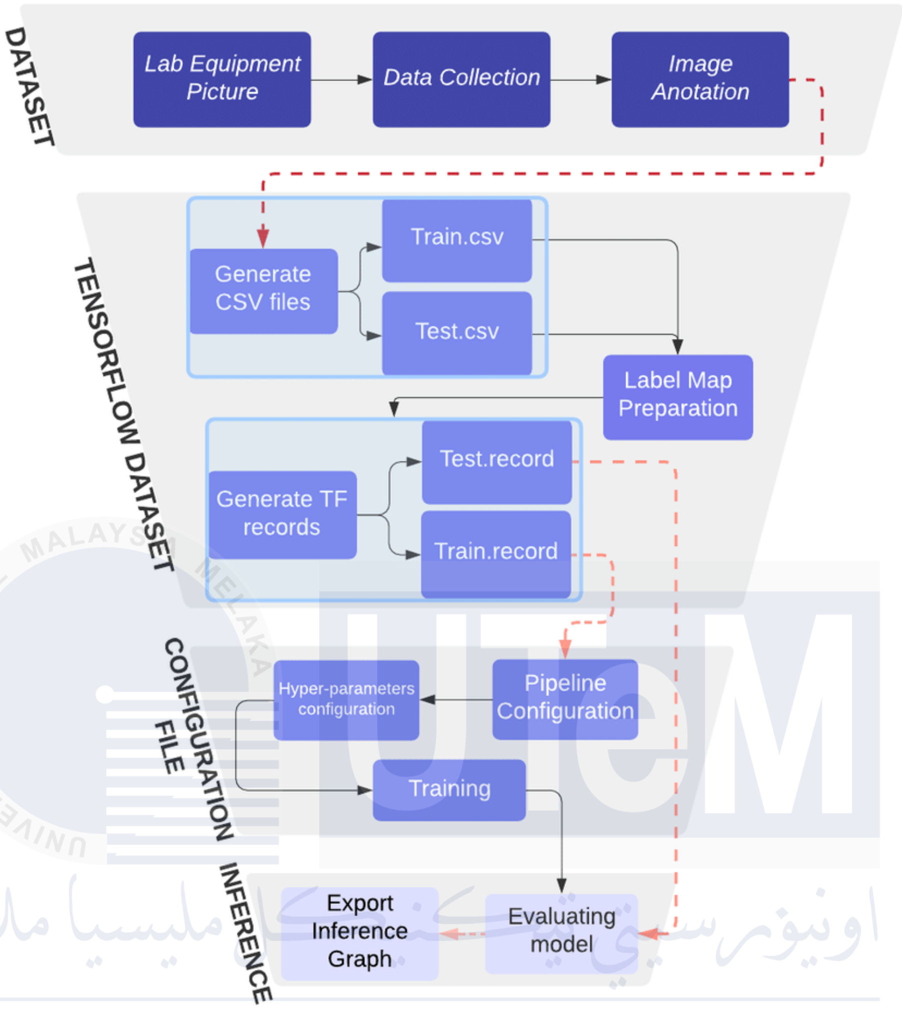

Figure 4.1 Training Proses Using TensorFlow Object Detection API (John Estrada et al., 2022)

Furthermore, the model is equipped with the ability to extract images where humans are detected by employing bounding box techniques. This means that the model not only identifies the presence of humans but also delineates the specific regions within the image where humans are located. This localized detection is crucial for providing detailed information about human presence and movement within a building, which is essential for effective emergency response planning.

Once a human is detected within the bounding box, the model performs additional detections within that box to ensure accuracy. It then provides a confidence score indicating the likelihood that the detected object is indeed a human. This high level of detail and accuracy is imperative for the project's goal of ensuring safe and efficient evacuation routes in emergency situations. By accurately detecting and tracking human locations, the system can dynamically adjust evacuation plans to avoid congestion and optimize escape routes, ultimately enhancing safety and efficiency during emergencies.

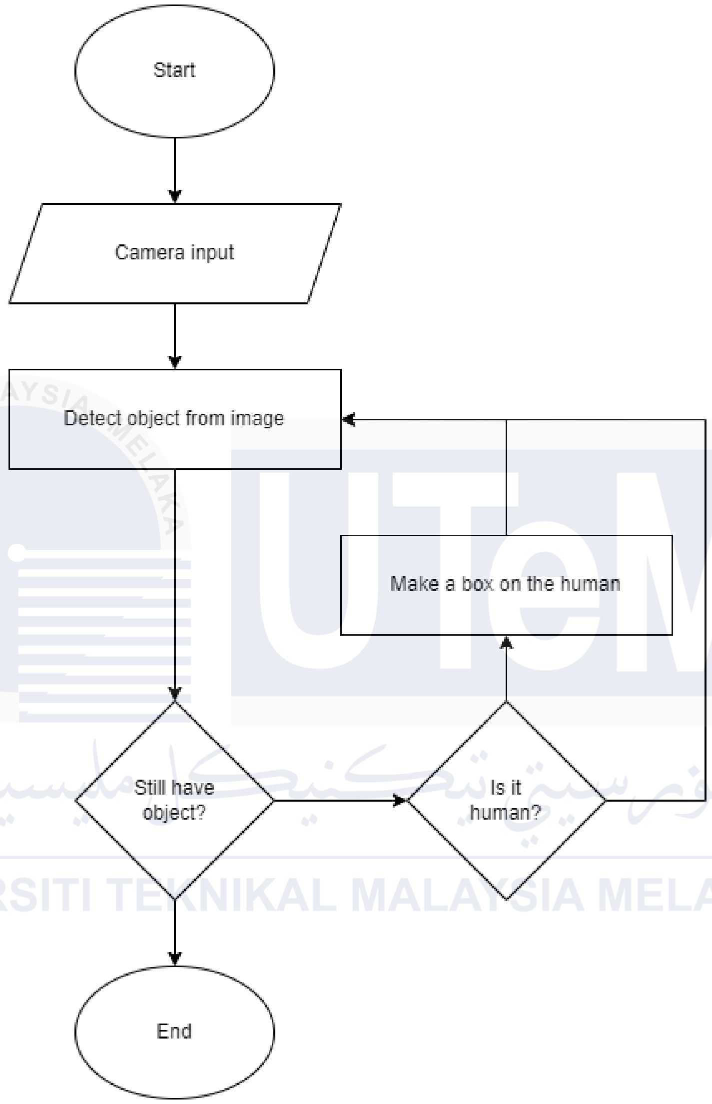

Figure 4.2 Human Detection Flowchart

Detection API is pretrained object detection model, powered by the Tensorflow Object Detection API, offers a robust and highly accurate solution for human detection within buildings. Its ability to handle extensive datasets and perform precise localized detections makes it an integral component of the proposed emergency routing system, ensuring that it can reliably support the safe evacuation of individuals in complex and dynamic environments.

4.2.2 Evacuation Time Estimation Model

The dataset for this project will be meticulously obtained from building plans that I will create. These building plans are meticulously referenced from actual building plans to ensure high accuracy and relevance. By using these detailed building data, the AnyLogic simulation software will be employed to simulate human movement through specified evacuation routes. This simulation is crucial for obtaining precise estimates of the time required for a specific number of individuals to traverse the designated routes during an evacuation scenario.

Once the simulation data is gathered, it will be utilized to train an estimation model designed to predict the time needed for evacuation based on several key variables which are the distance of the route, the width of the route, and the number of people using the route. The dataset will be carefully split into two subsets to facilitate robust training and testing of the model where 80% of the data will be allocated for training purposes, ensuring the model learns effectively from a substantial amount of information, while the remaining 20% will be reserved for testing. This split ensures that the model's predictions can be rigorously evaluated against unseen data, providing a clear measure of its generalization capabilities.

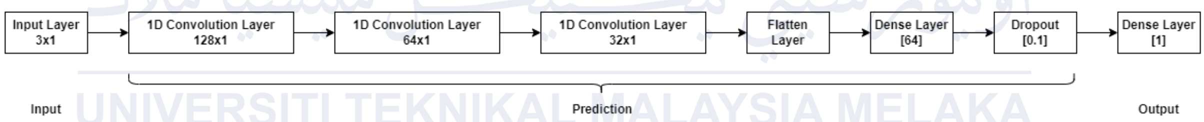

Figure 4.3 Estimation Model Networks

The estimation model itself will be designed using a CNN architecture that consisting of seven layers. CNNs are chosen for their powerful ability to capture complex patterns and relationships in the data, which is essential for accurately estimating evacuation times based on various route characteristics. The seven-layer architecture is intended to balance complexity and performance, enabling the model to learn nuanced features while maintaining computational efficiency.

To evaluate the performance of the estimation model, the MAE metric will be employed. MAE is a widely used metric for assessing the accuracy of regression models, as it measures the average magnitude of errors between predicted values and actual values, without considering their direction. The goal for the model's performance is to achieve an MAE below 0.2. This threshold ensures that the model's predictions are within a small margin of error, providing reliable and actionable insights for emergency evacuation planning. By calculating the MAE, the model's accuracy can be quantitatively assessed, ensuring that the predictions it generates are precise and dependable.

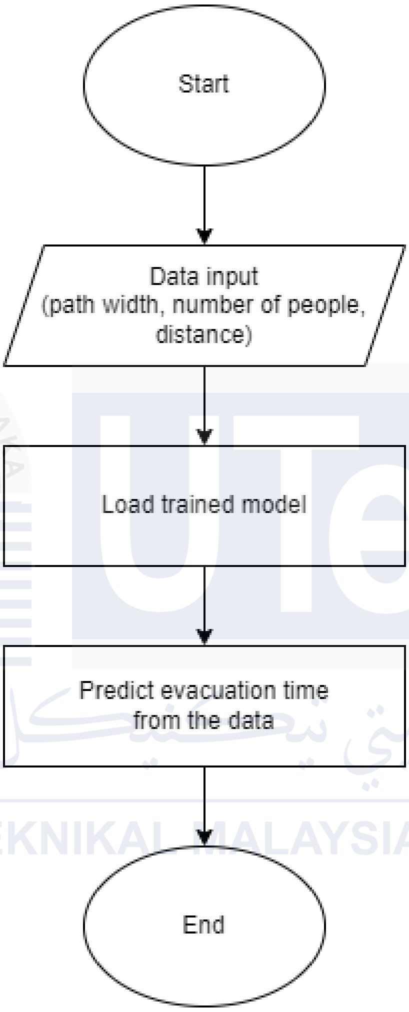

Figure 4.4 Estimation Model Flowchart

In conclusion, the dataset derived from accurately referenced building plans, simulated through AnyLogic, will form the backbone of the estimation model. This model, constructed using a seven-layer CNN, will be rigorously trained and tested to predict evacuation times with high accuracy. The performance assessment through MAE ensures that the model meets stringent accuracy standards, thereby providing a reliable tool for optimizing evacuation routes and enhancing safety during emergencies.

4.2.3 Best Route Calculation

The algorithm chosen for determining the optimal evacuation route in this project is Dijkstra’s Algorithm, renowned for its efficiency in finding the shortest paths in weighted graphs. Dijkstra's Algorithm operates by iteratively selecting the node with the smallest tentative distance from the source node. Once a node is selected, the algorithm updates the distances to its neighbouring nodes, adjusting these values if a shorter path is found through the selected node. This node is then marked as visited, indicating that the shortest path to this node has been definitively determined. This process is repeated, with the algorithm continually selecting the next node with the smallest tentative distance, updating distances, and marking nodes as visited, until all nodes in the graph have been processed. By the end of this iterative process, Dijkstra’s Algorithm ensures that the shortest path from the source node to every other node in the graph is identified.

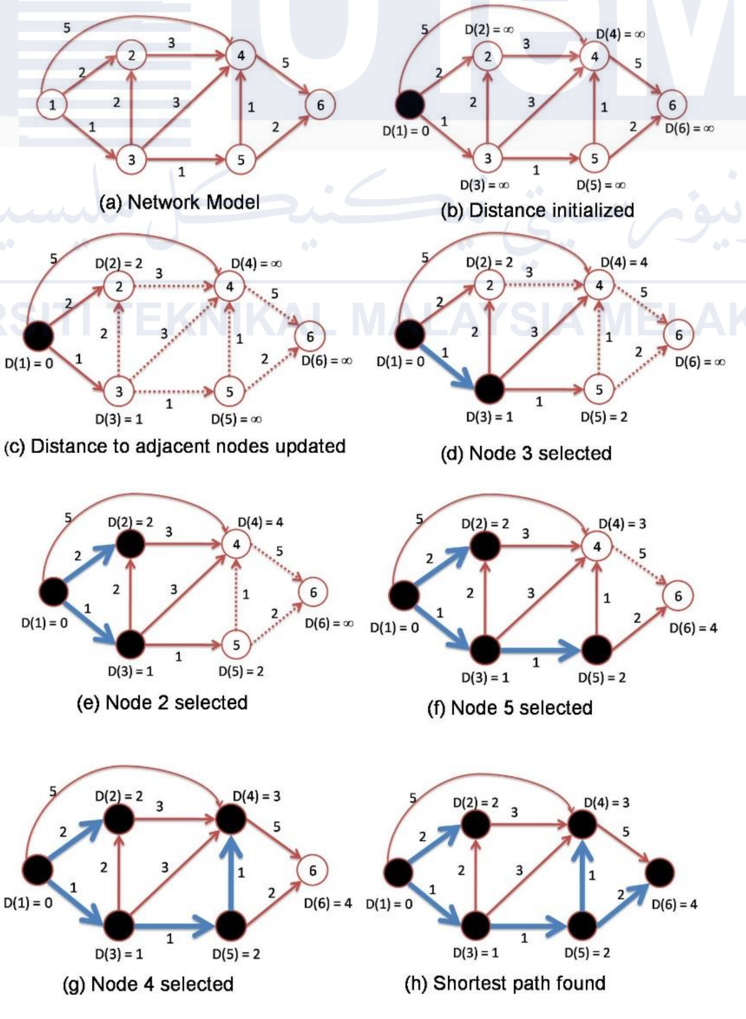

Figure 4.5 Process of Dijkstra's Algorithm (Tami, 2013)

For this project, the building plan will be meticulously analysed and marked with specific points to represent the nodes in the graph. These nodes will typically correspond to critical points in the building, such as intersections of corridors, exits, and other significant locations. The routes between these nodes will form the edges of the graph, each representing a potential path for evacuation.

The innovative aspect of this implementation lies in the values assigned to these edges. Instead of using the traditional approach of assigning distances as edge weights, this project utilizes the predicted evacuation time derived from a sophisticated estimation model. This model considers multiple variables, including the distance of the route, the width of the corridor, and the density of people, to predict the time required to traverse each edge. This approach ensures that the algorithm calculates the most efficient evacuation route based on realistic, dynamic factors rather than static distances.

To further enhance the efficiency of the algorithm, the algorithm is optimised on computing time where the algorithm stores a snapshot of the building's map upon startup. This stored map includes all nodes and edges, along with the calculated evacuation times for each edge based on the initial conditions. The algorithm then continually monitors the environment for any changes, such as the change of the density of people or a blocked corridor. The algorithm updates the map and recalculate the shortest path when the change is detected. This approach ensures that the algorithm does not waste valuable time recalculating the shortest path in every run when the situation remains unchanged.

In practical terms, Dijkstra's Algorithm will process the graph by evaluating each point and determining the edge with the predicted shortest evacuation time that connects two selected points. This means that the algorithm will not merely identify the shortest physical route, but rather the most efficient route considering the environmental variables within the building. This includes factors such as potential bottlenecks, varying corridor widths, and the expected movement speed of evacuees.

By incorporating these predictive time values, the algorithm can dynamically adapt to different scenarios within the building, ensuring that the chosen route minimizes evacuation time under various conditions. This approach provides a significant advantage in emergency situations, where traditional shortest-path algorithms might fail to account for real-time variables that impact evacuation efficiency.

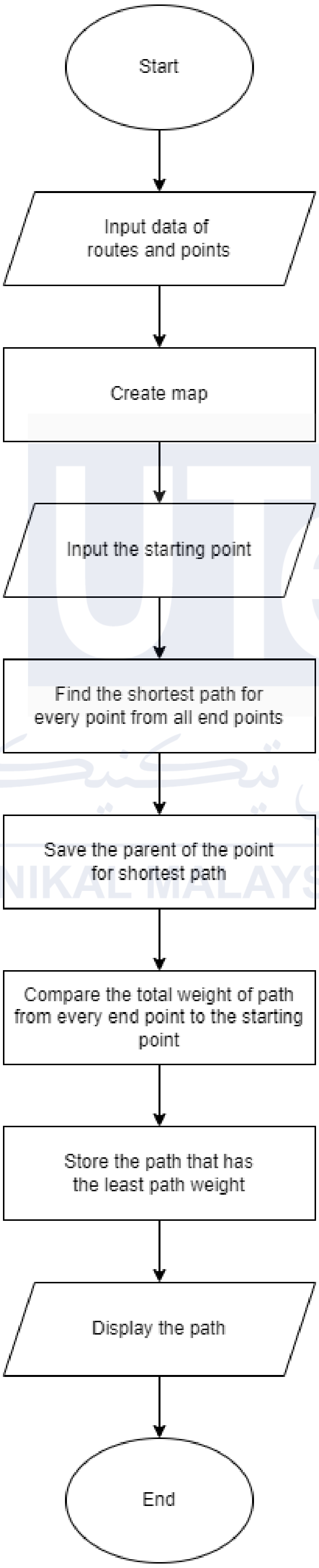

Figure 4.6 Dijkstra Algorithm Flowchart

In summary, Dijkstra's Algorithm will be applied to a graph representation of the building, with nodes and edges defined by key points and routes within the building. The edges will be weighted not by distance but by predicted evacuation times, derived from a comprehensive estimation model. This ensures that the algorithm identifies the most efficient evacuation route, optimizing for both time and safety by considering the dynamic conditions of the evacuation scenario. This sophisticated application of Dijkstra's Algorithm thus promises to enhance the effectiveness of the Emergency Routing System for Buildings Using Dijkstra's Algorithm, providing reliable and optimized routes for safe evacuation in emergency situations.

4.3 Experiment Design

To comprehensively evaluate the performance of Dijkstra's Algorithm, the system will be rigorously tested against two other well-established algorithms which are the Bellman-Ford Algorithm and the Floyd-Warshall Algorithm. These comparative tests aim to ascertain how Dijkstra's Algorithm measures up in terms of efficiency and effectiveness under various emergency evacuation scenarios.

In designing these experiments, several critical parameters and assumptions will be established. Firstly, the number of evacuees will be set to the maximum capacity that each node in the building can handle. This ensures that the tests simulate the most challenging and realistic conditions possible, where the evacuation system must manage peak loads effectively. The testing protocol will involve running the evacuation system iteratively, processing nodes sequentially from the first to the last, rather than attempting to calculate all nodes simultaneously. This sequential approach is crucial because it allows the system to dynamically update the graph of the building as the evacuation progresses. By recalculating the graph node-by-node, the system can more accurately reflect the changing conditions within the building, such as congestion and the movement of evacuees, which would otherwise be missed in a single and static calculation.

Three distinct types of building scenarios will be used to test the system. The first scenario is the multi-exit building scenario, where the system will be tested on a building with multiple exits. The number of exits will be varied, starting from one and increasing to the maximum number of exits available in the building. This will help assess how the system manages multiple evacuation routes and how efficiently it directs evacuees to the nearest safe exit under different exit configurations. The second scenario involves testing the system in a multi-floor building. The number of floors will be incrementally increased from one to the maximum number of floors in the building. This test is designed to evaluate the system's capability to handle vertical evacuations, where evacuees need to move between different floors to reach safety in high-rise buildings.

Lastly, the different area size buildings scenario will be used. In this scenario, two buildings with the same number of floors and exits but differing in area size and the number of nodes will be tested. This will examine the system's scalability and its ability to manage larger, more complex evacuation networks compared to smaller ones. The objective is to understand how the system performs in terms of processing time and route optimization in buildings of varying sizes.

Table 4.1 Experiment Scenario Details

Experiment Building Scenario	Number of floors	Number of exits	Area of building (m^2)	Number of cameras
1	1	1	77.44	9
2	2	1	320.40	32
3	1	3	312.12	23
4	3	8	1880.70	123

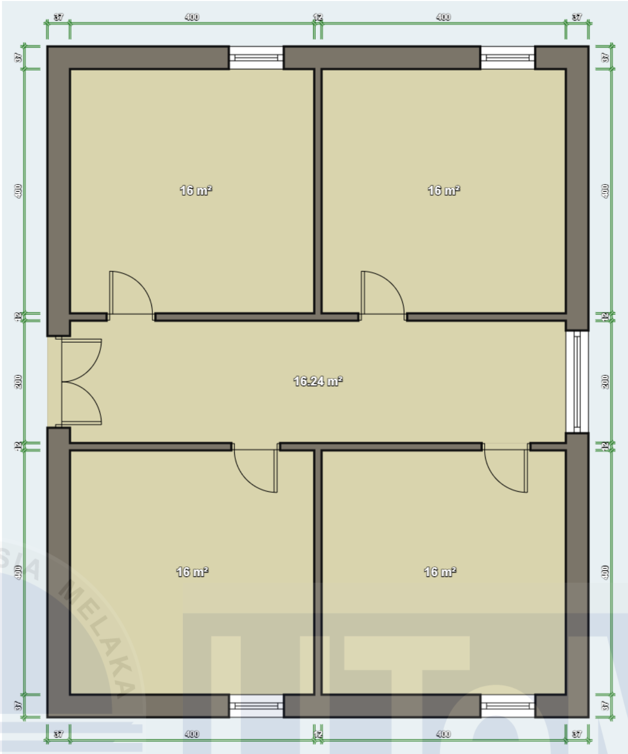

Figure 4.7 Building 1 Floor Plan

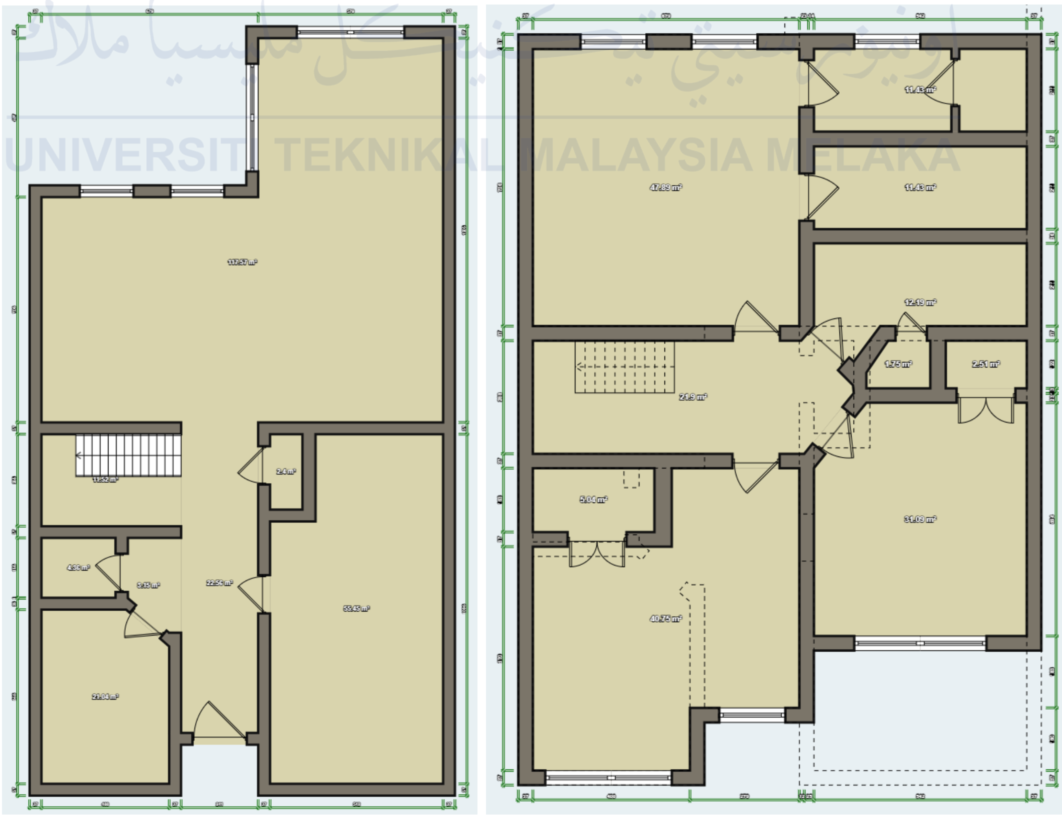

Figure 4.8 Building 2 Floor Plan

Figure 4.9 Building 3 Floor Plan

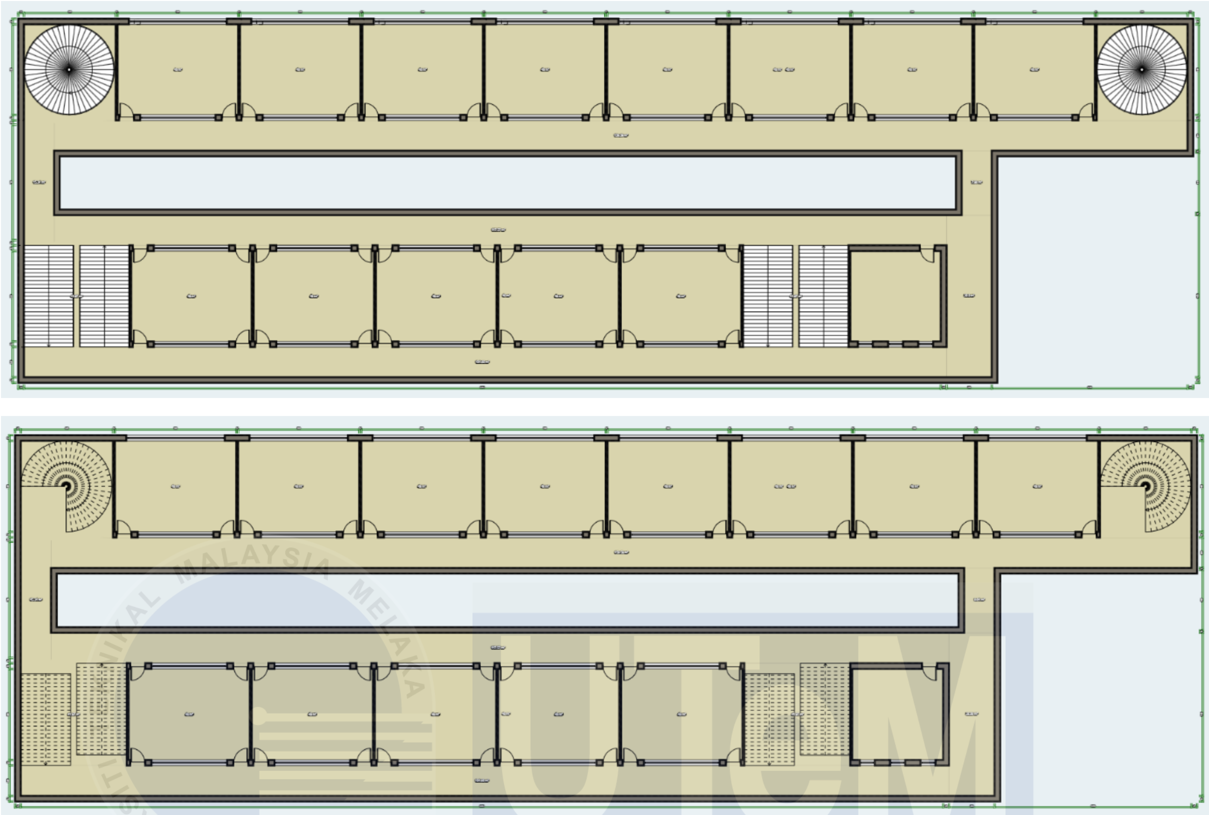

Figure 4.10 Building 4 Floor Plan

These experiment scenario examples are created using Floor Plan Creator by referencing actual floor plan of the buildings. The environment data in the buildings will be collected through cameras. The cameras will be placed in the rooms, the door of rooms, the entry of stairs and exits. The direction of cameras in the rooms is towards the interior of the rooms to monitor the situation in the rooms. The cameras at the doors monitor the entrance of rooms and the pathway around the rooms, providing the environment data around the rooms to ensure evacuates able to pass by the place. The cameras at the stairs also help to collect the data of the condition of stairs and ensure the entrance of the stairs is not blocked. The cameras also placed at the exits to ensure the escape path is existing and monitor the evacuation. The effecting range of CCTV in the experiment is defined as 30m, hence extra camera is needed in the pathways if there is not any camera within the range of 25m.

In these scenarios, the system will be evaluated based on three key performance metrics. The average escape time measures the meantime taken for all evacuees to reach safety across the different scenarios. This is a critical indicator of the system's overall effectiveness in an emergency evacuation. The average processing time assesses the computational efficiency of the algorithm, determining how quickly the system can generate evacuation routes under various conditions. Lastly, the total number of evacuees evaluates the system’s capacity to safely evacuate all individuals within the building, providing a comprehensive measure of its operational performance.

4.4 Summary

In conclusion, I have introduced the proposed solution designed to tackle the project's challenges effectively. It includes three main components which are the human detection model, evacuation time estimation model, and best route calculation using Dijkstra’s Algorithm. The human detection model utilizes a pretrained object detection model trained with the Tensorflow Object Detection API, specifically for accurate human detection in building environments. The evacuation time estimation Model uses building plan data from AnyLogic to train a CNN model, predicting evacuation times based on route specifics. Dijkstra’s Algorithm is employed for best route calculation, optimizing evacuation paths based on predicted time rather than distances. Comparative testing against Bellman-Ford and Floyd-Warshall Algorithms across multi-exit, multi-floor, and differently sized buildings assesses performance in terms of escape time, processing efficiency, and total evacuees. This chapter demonstrates a systematic approach to developing an Emergency Routing System for Buildings Using Dijkstra’s Algorithm, aiming to enhance emergency response in complex environments. The following chapter will provide the results of this project with analysis.

CHAPTER 5: RESULTS AND DISCUSSION

5.1 Introduction

This chapter provides a comprehensive overview of the results generated from a series of experimental scenarios conducted during the project. It delves into a detailed analysis of these outcomes, exploring the nuances and implications of the data collected. Additionally, this chapter focuses on comparing the performance of the three different algorithms implemented in the project, offering an in-depth evaluation of their strengths, weaknesses, and overall effectiveness. Through this comparative analysis, the chapter aims to identify which algorithm performs best under various conditions, determining the most suitable approach for the project's objectives.

5.2 Results

The estimation model is trained with 200 epochs and validated. Figure 5.1 shows the loss of the model which is 0.2335 and the mean accuracy error (MAE) of the model is 0.1403, shown in Figure 5.2.

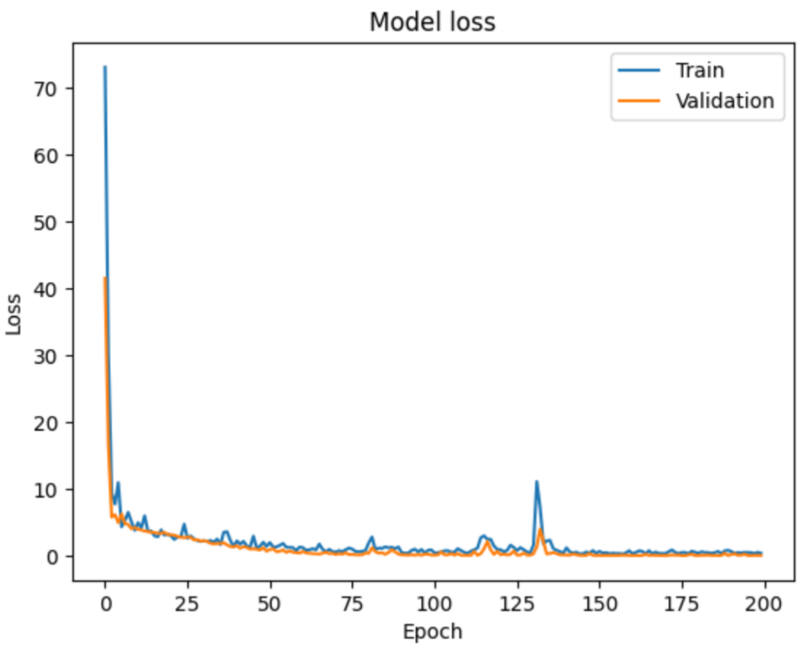

Figure 5.1 Prediction Model Loss Graph

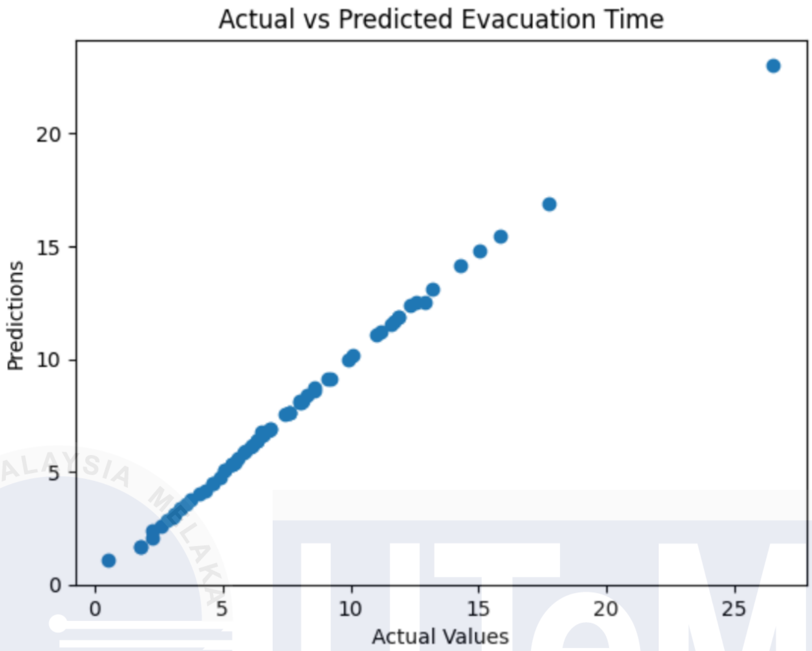

Figure 5.2 MAE of The Estimation Model Graph

Figure 5.3 and Figure 5.4 are the evacuation time and the processing time respectively, gained from using three different algorithms which are Dijkstra Algorithm, Bellman-Ford Algorithm and Floyd-Warshall Algorithm under several different random buildings. The buildings are having same number of floor and same area which is same number of nodes but different layout and the number of exits. This testing is to find out the effect of the number of exits on the performance of three algorithms by changing the number of exits from 1 to 10, with constant 80 nodes.

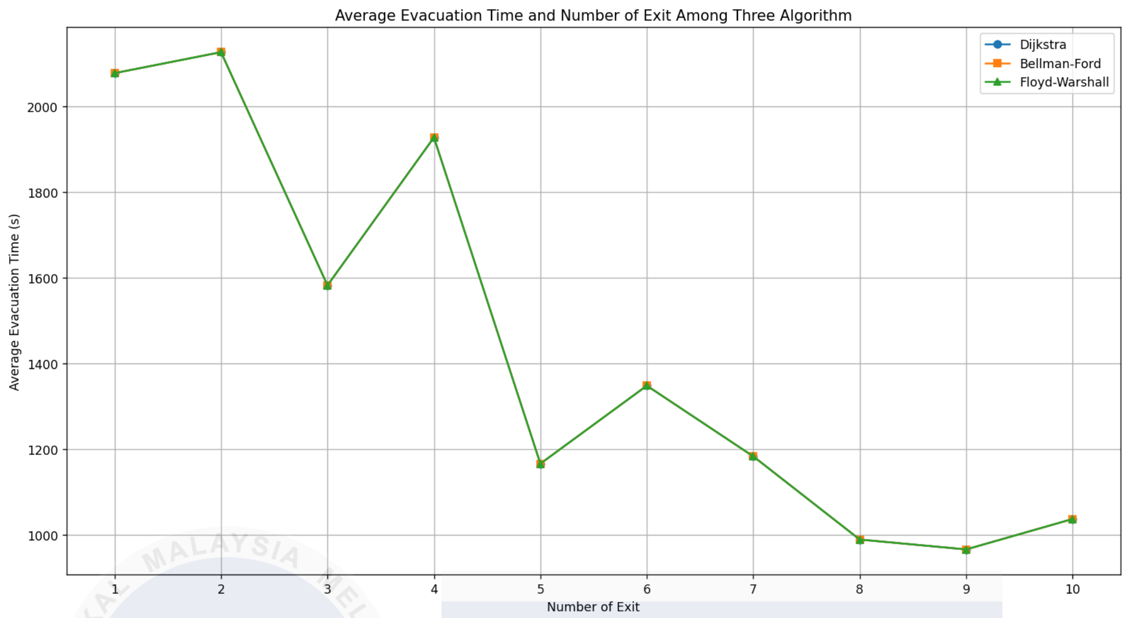

Figure 5.3 Average Evacuation Time and Number of Exit Among Three Algorithm Under Testing

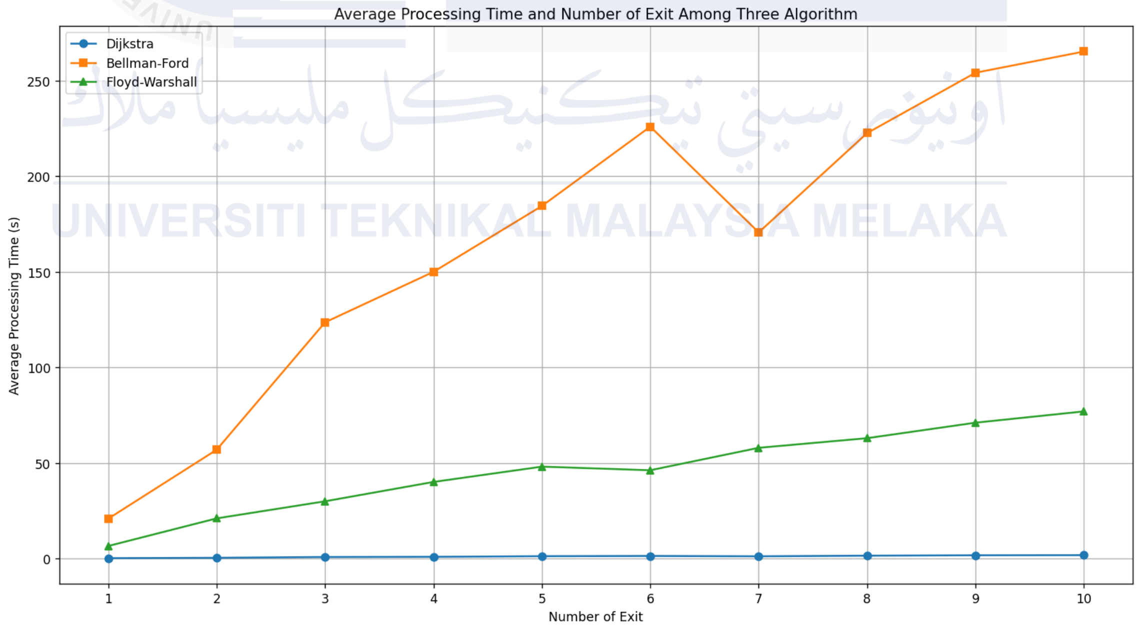

Figure 5.4 Average Processing Time and Number of Exit Among Three Algorithm Under Testing

Besides, Figure 5.5 and Figure 5.6 are the evacuation time and the processing time respectively, gained from using three algorithms for Building 4. By changing the number of exits, the performance of three algorithms is recorded.

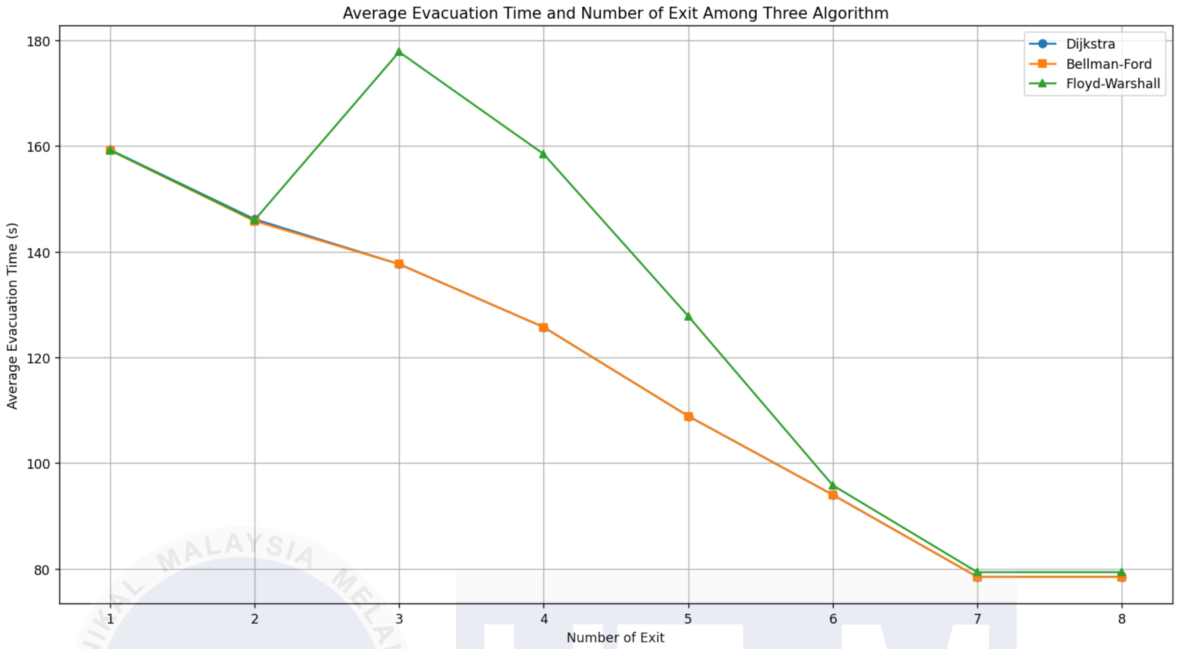

Figure 5.5 Average Evacuation Time and Number of Exit Among Three Algorithm for Building 4

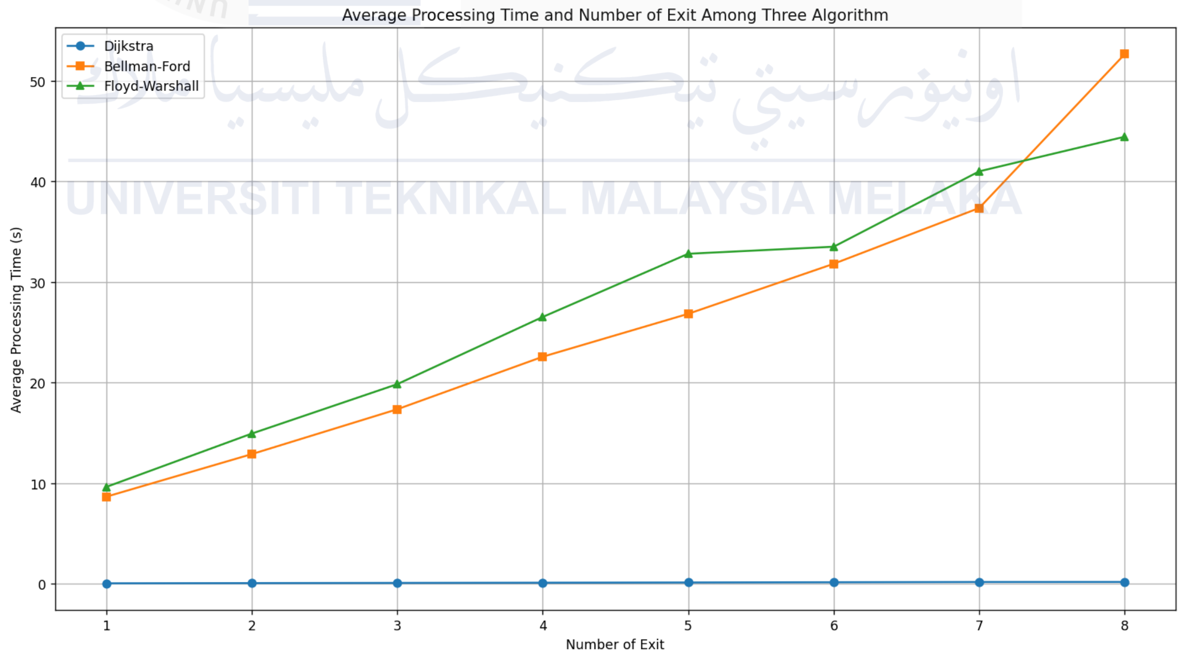

Figure 5.6 Average Processing Time and Number of Exit Among Three Algorithm for Building 4

Figure 5.7 and Figure 5.8 are the evacuation time and the processing time respectively, gained from using three different algorithms under several different random buildings. The buildings are having same number of floor but different number of exits, layout and number of nodes to represent different area of buildings. This testing is to find out the effect of the building area on the performance of three algorithms by changing the number of nodes from 1 to 80, and the number of exits varies based on the number of nodes.

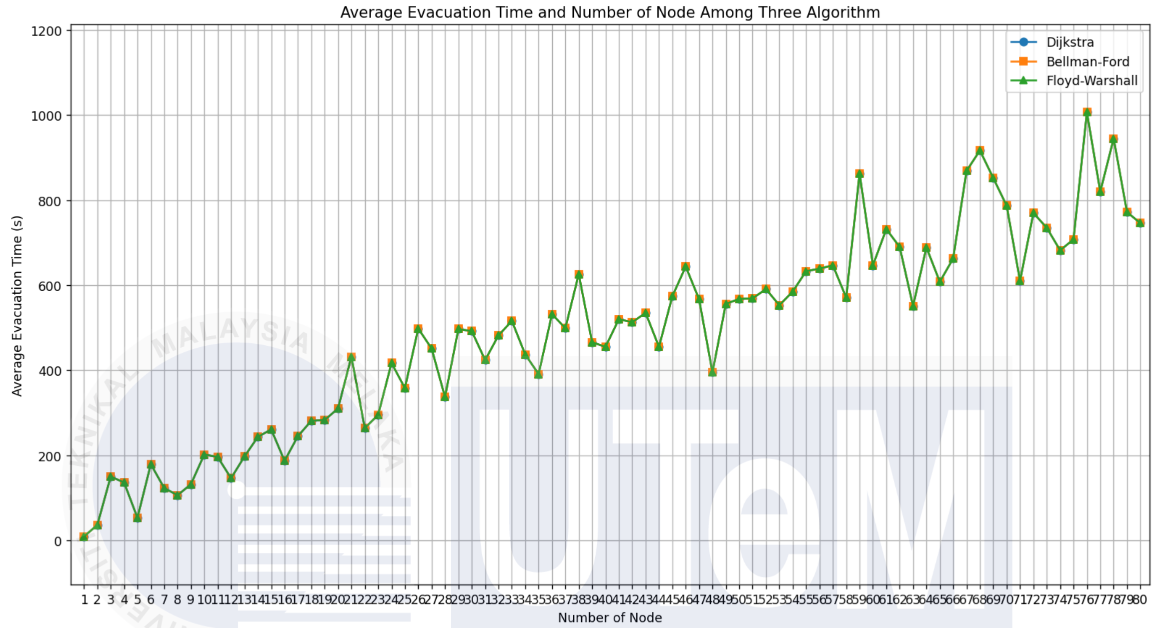

Figure 5.7 Average Evacuation Time and Number of Node Among Three Algorithm Under Testing

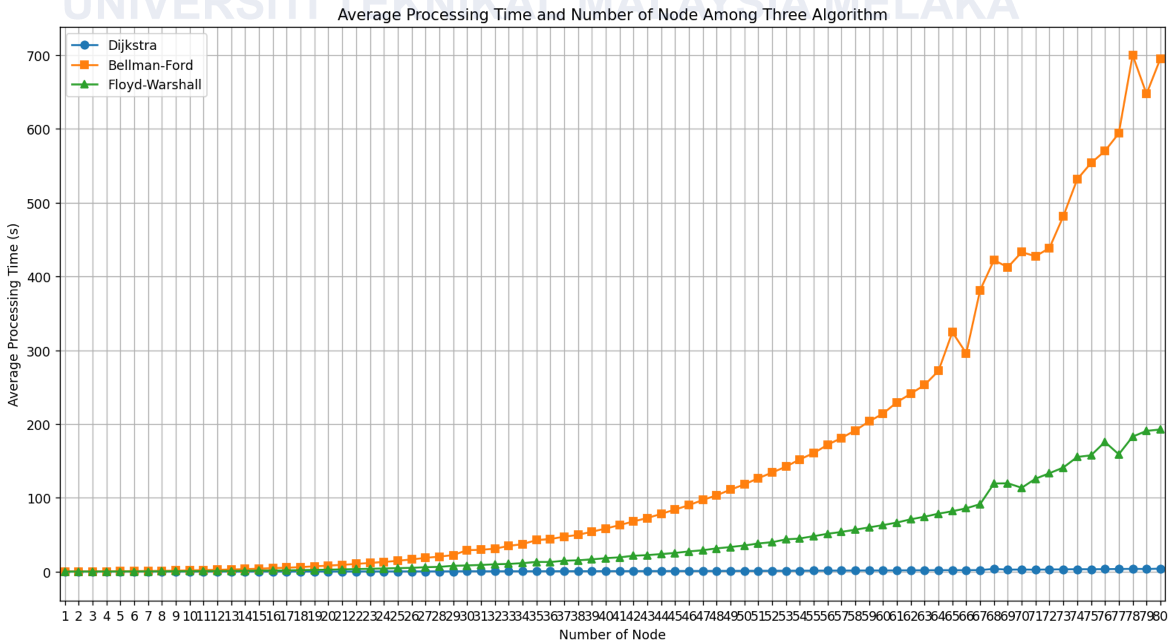

Figure 5.8 Average Processing Time and Number of Node Among Three Algorithm Under Testing

Apart from that, Figure 5.9 is the evacuation time and the processing time gained from using three different algorithms for Building 1 and Building 3. The performance of three algorithms is recorded based on Building 1 with 77.44m^2 and Building 3 with 312.12m^2.

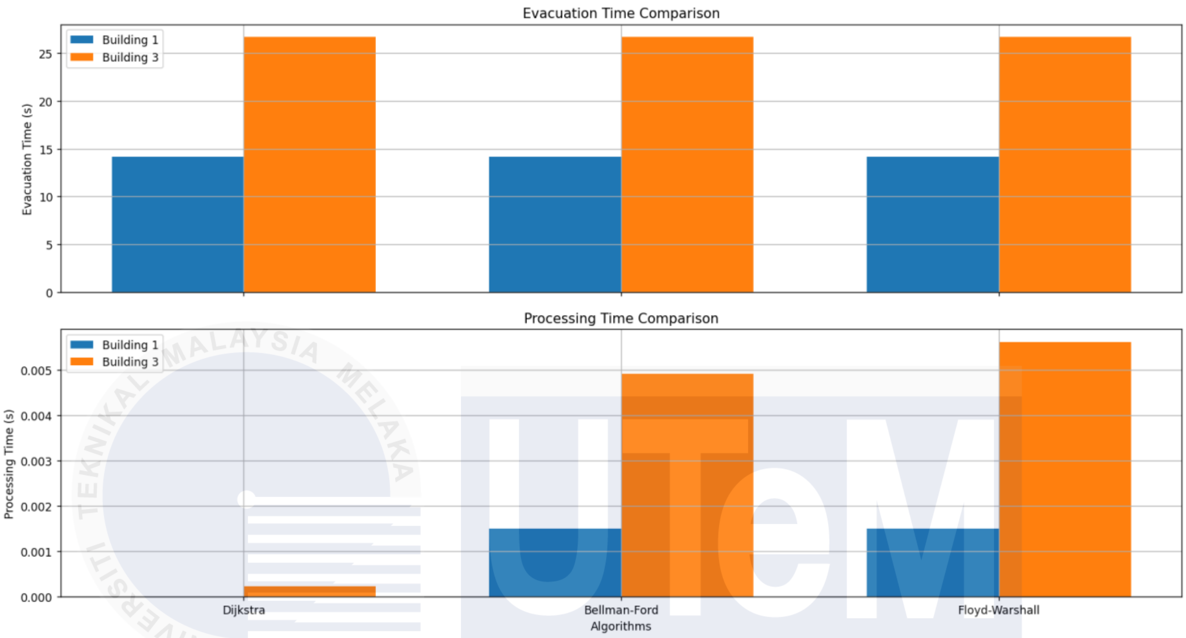

Figure 5.9 Average Evacuation Time and Processing Time Among Three Algorithm for Building 1 and Building 3

The evacuation time and the processing time gained from using three different algorithms under Building 4 with different number of exits scenario are represented as Figure 5.10 and Figure 5.11 respectively. The number of exits in the building 4 is changed and the performance are compared for the building with different number of floors. This testing is to find out the effect of the number of floors on the performance of three algorithms by changing the number of floors of building 4 from 1 to 3.

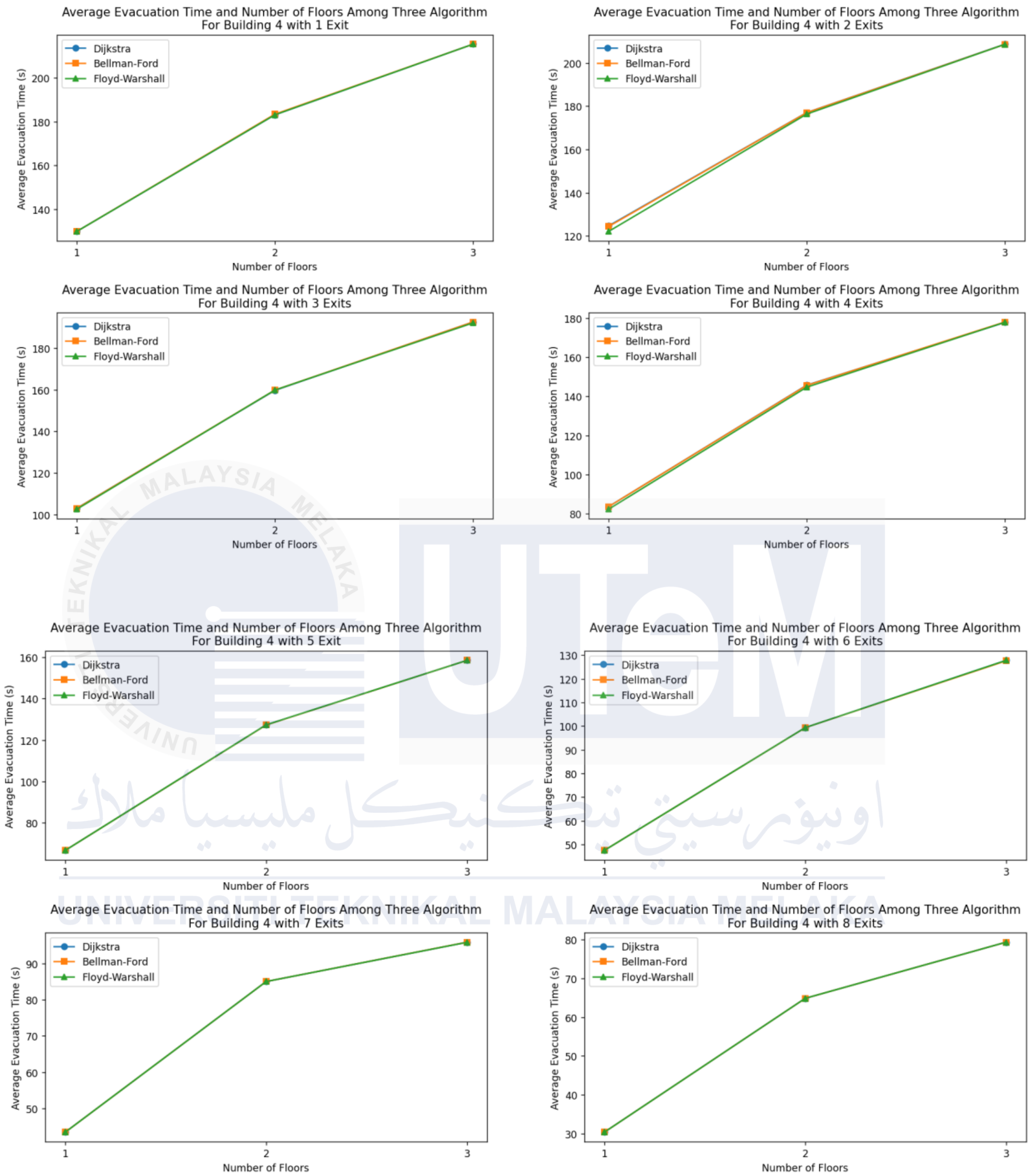

Figure 5.10 Average Evacuation Time and Number of Floors Among Three Algorithm for Building 4 with Different Number of Exits

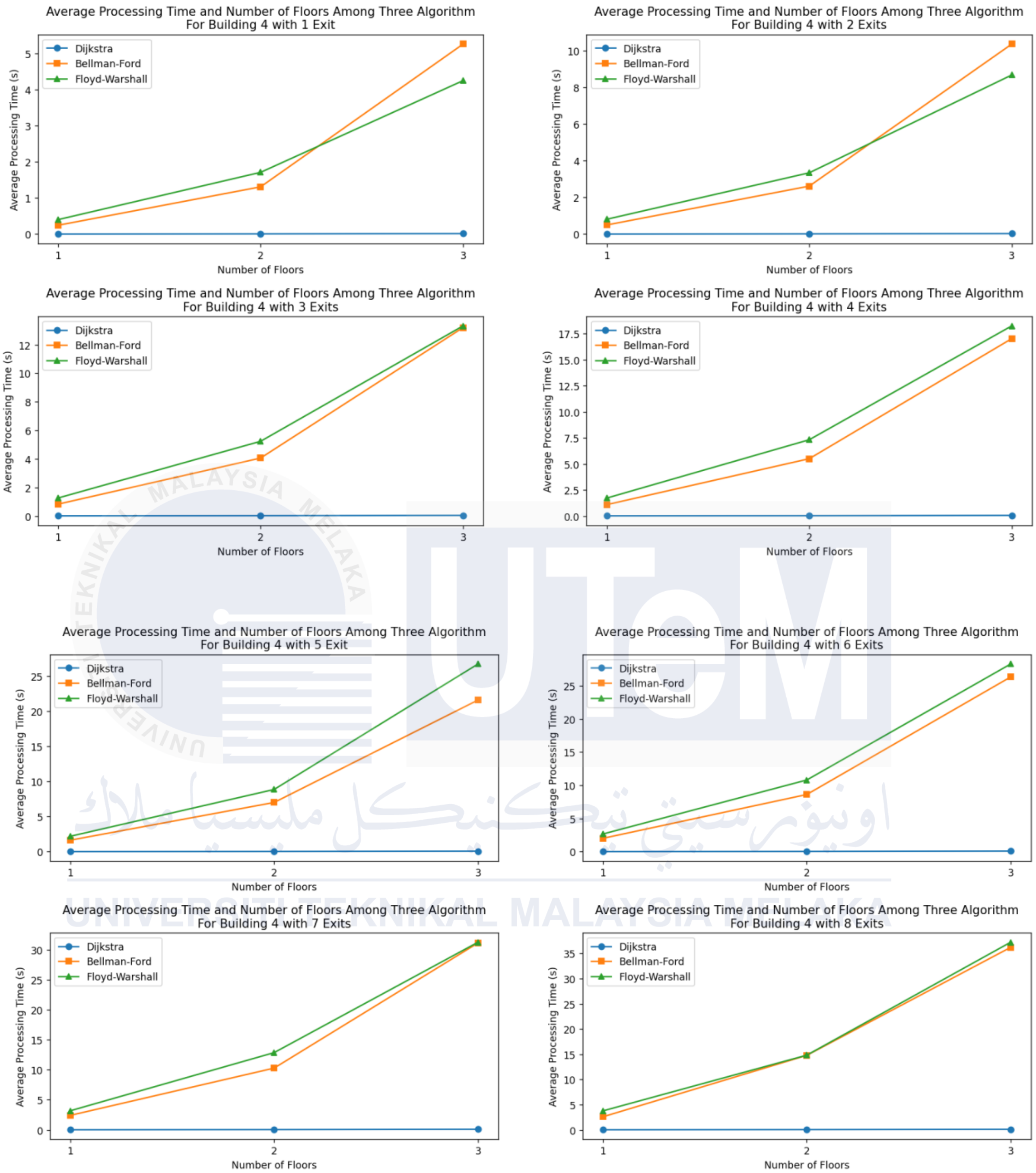

Figure 5.11 Average Processing Time and Number of Floors Among Three Algorithm for Building 4 with Different Number of Exits

Next, Figure 5.12 and Figure 5.13 is the evacuation time and the processing time respectively, gained from using three different algorithms for Building 4. The performance of three algorithms is recorded by changing the included number of floors for Building 4.

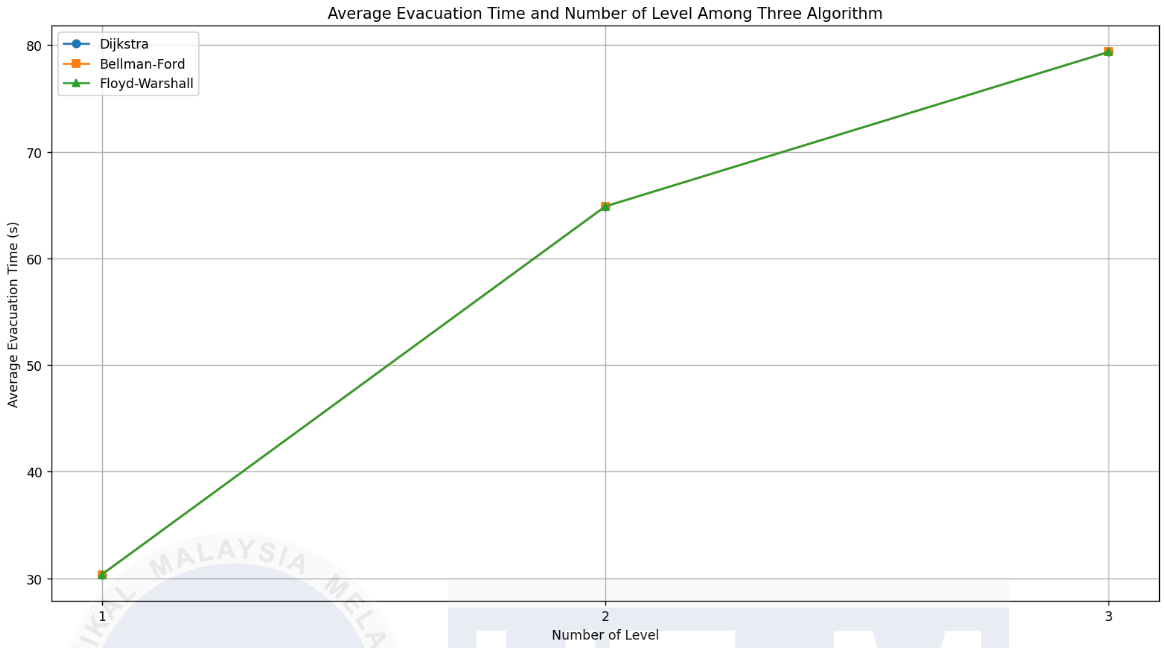

Figure 5.12 Average Evacuation Time and Number of Level Among Three Algorithm for Building 4

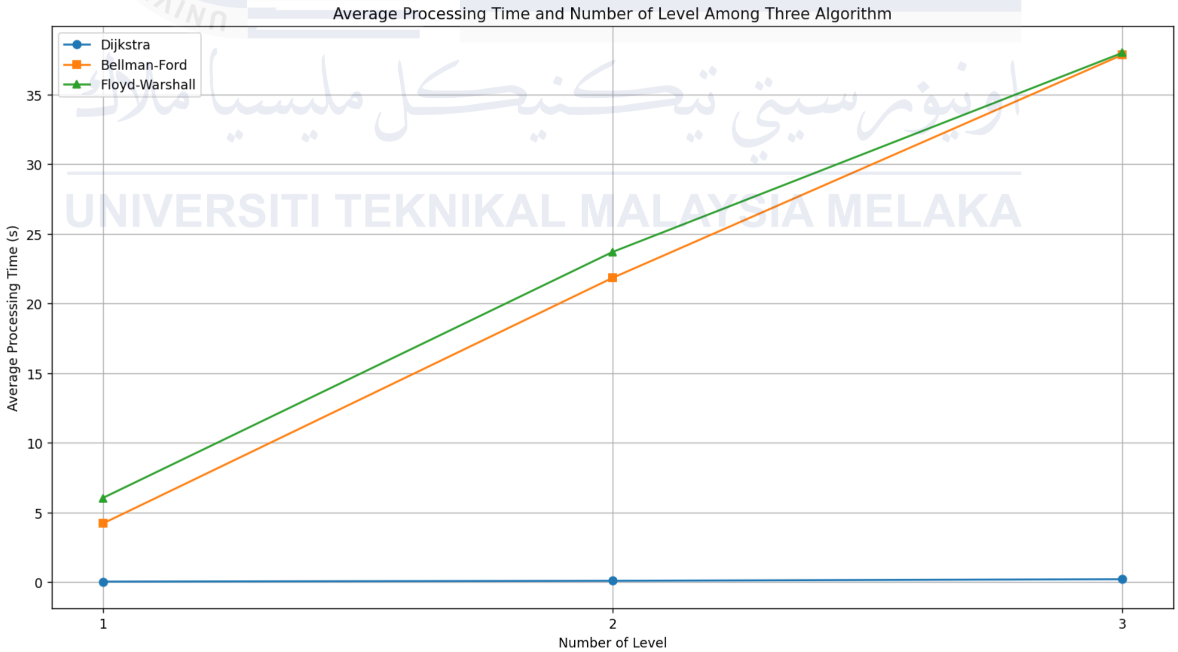

Figure 5.13 Average Processing Time and Number of Level Among Three Algorithm for Building 4

5.3 Analysis and Discussion

The results indicate that the evacuation time estimation model achieved a low loss value of 0.2335 and a Mean Absolute Error (MAE) of 0.1403. However, these values fall short of the ideal requirements, where the loss value should be below 0.1 and the MAE should be under 0.001. In emergency situations, precise time estimation is crucial for rescuers to efficiently evacuate people from a building. A higher MAE introduces greater uncertainty in time predictions, which becomes more significant as the evacuation duration increases. For example, in scenarios where evacuation might take up to two hours, an MAE of 0.14 could result in an uncertainty of 15-20 minutes, a margin that is considerably large in critical situations.

Additionally, a comparison of three algorithms which are Dijkstra’s Algorithm, Bellman-Ford’s Algorithm, and Floyd-Warshall’s Algorithm, demonstrates that all three can identify nearly the same evacuation time, indicating that they can efficiently determine the most effective evacuation route in any situation, given accurate environmental variables. However, it is noted that Floyd-Warshall’s Algorithm produces a slightly higher evacuation time when the number of exits changes in Building 4. This suggests that, while all three algorithms are capable of identifying the most efficient evacuation path, Floyd-Warshall’s Algorithm may be less optimal under specific conditions.

In terms of processing time, there are significant differences among the three algorithms. Dijkstra’s Algorithm consistently exhibits the lowest processing time, even as the number of exits, building area, and number of building levels increase. In contrast, both Bellman-Ford’s and Floyd-Warshall’s Algorithms show a marked increase in processing time as these parameters rise, with processing times becoming extremely high under more complex conditions. Low processing time is critical, as the system must recalculate routes whenever the situation in the building changes which occurs frequently, especially during emergencies. A shorter processing time ensures that the system can provide timely feedback on new routes.

Based on the research, each of these three algorithms has its own specific use case, and they perform well in their respective scenarios. For finding the optimal route from a location to an exit in a building, which can be represented as a network of nodes, Dijkstra’s Algorithm is particularly well-suited to this project. This algorithm is designed for scenarios like this, where the goal is to identify the best path from one point to another. The other two algorithms which are Bellman-Ford’s Algorithm and Floyd-Warshall’s Algorithm, are better suited for more complex scenarios or specific conditions, often resulting in higher processing times and potentially more detailed outputs than just the shortest path.

The results indicate that Dijkstra's Algorithm is the most suitable choice for this project. It offers acceptable performance in determining the best route, as all three algorithms produce nearly identical evacuation times, while also significantly outperforming the others in terms of processing time. If an alternative route is needed, Bellman-Ford's Algorithm may offer better performance; if there is a need to determine the shortest path between every pair of points in the building, Floyd-Warshall's Algorithm might be more suitable. However, due to the purpose of this system, Dijkstra's Algorithm is the preferred choice based on its demonstrated efficiency in testing.

5.4 Summary

In short, I evaluated the performance of the evacuation time estimation model and compared the efficiency of three different pathfinding algorithms which are Dijkstra's Algorithm, Bellman-Ford's Algorithm, and Floyd-Warshall's Algorithm. The results demonstrate that while the model achieves a relatively low loss and MAE, but it falls short of the ideal benchmarks required for high-stakes emergency scenarios, where precision in time estimation is paramount. The analysis of the three algorithms revealed that all could determine efficient evacuation routes under various conditions, though Dijkstra's Algorithm proved to be the most suitable due to its consistently low processing time across different building configurations. This efficiency is critical for real-time applications where frequent updates are necessary to adapt to dynamic emergency situations. While the project successfully identified the most appropriate algorithm, the findings also underscore the need for further refinement of the evacuation time estimation model to enhance its accuracy and reliability.

CHAPTER 6: PROJECT CONCLUSION

6.1 Introduction

In this final chapter, the project is wrapped up on Emergency Routing System for Buildings Using Dijkstra's Algorithm by summarizing the key findings, discussing the advantages and disadvantages of the project, and presenting avenues for future research and development. Throughout this project, the objective was to find out the performance and suitability of Dijkstra's Algorithm on emergency routing system for buildings.

6.2 Project Summarization

The targeted goals were successfully accomplished. Two more methods known as Bellman-Ford Algorithm and Floyd-Warshall Algorithm are used to further enhance the model, and each algorithm is examined to see which produces the best performance outcomes. The final model generates an escape route for everyone in the building and adjust it based on the real time condition after testing and comparison.

Table 6.1 Project Summarization

Problem Statement	Objective	Conclusion
Current emergency routing systems often fail to adequately accommodate the diverse needs of victims and adapt to dynamic evacuation scenarios.	To assess the effectiveness of existing emergency routing systems in accommodating diverse victim needs and adapting to dynamic evacuation scenarios.	Human Detection Model and Evacuation Time Estimation Model are applied in the system to improve the effectiveness of the system and understand the environment requirements.

During emergencies, determining the shortest and safest evacuation routes while minimizing evacuation time and congestion within buildings remains a significant challenge.	To develop an emergency routing system for buildings using Dijkstra's Algorithm.	I successfully developed a Dijkstra’s Algorithm based emergency routing system for real time evacuation.
Evaluating the performance of emergency routing systems that utilize algorithms like Dijkstra's is crucial to ensure their efficiency, reliability, and scalability in various emergency scenarios and building environments.	To evaluate the performance of the system using statistical methods.	The performance of the system is successfully evaluated based on the evacuation time and the processing time in various scenario such as the change of the number of the exit, the change of the number of the floor and the change of the building area.

6.3 Project Contribution

This project plays an important role in improving the capabilities of building safety officers by using information from the sensors and detectors in the building to real time figure out the best route for the building evacuation. The emergency responders able to quickly identify the safest evacuation routes, enhancing their ability to manage emergencies effectively. It equips first responders with real-time data to guide evacuees efficiently and provides rescuers with precise information to reach those in need. Apart from that, it helps building staff to deliver clear and timely instructions to occupants, ensuring a coordinated and swift evacuation for everyone involved. As a result, this project will significantly reduce the risk of fatalities and injuries during building emergencies, increasing the chances of safely rescuing more occupants from unexpected disasters.

6.4 Project Limitation

6.4.1 Human Detection Model

The current model has limitations that can impact its accuracy in detecting people during an emergency situation. One of the issues is its inability to distinguish between individuals who are closely overlapped in the image; the model will count them as a single person. This can lead to an underestimation of the number of occupants in a particular area, which may affect the effectiveness of evacuation planning and resource allocation.

Additionally, the model has a detection range constraint. People who are positioned too far from the camera may not be detected because they appear too small in the image for the model to recognize. This limitation could result in certain individuals being missed entirely in the evacuation planning, potentially leading to delayed assistance or rescue efforts.

These limitations highlight the need for further refinement of the model to improve its ability to accurately detect and count all individuals in a given space, regardless of proximity or distance, ensuring a more reliable and comprehensive evacuation process.

6.4.2 Evacuation Time Estimation Model

The model does not consider the arrangement of evacuation order, meaning it doesn’t account for the optimal sequence in which people should exit the building based on their location or the severity of the situation. This oversight could lead to inefficiencies, such as bottlenecks or unnecessary delays. Additionally, the model neglects the varying conditions of evacuees. It assumes that everyone can evacuate at the same pace, overlooking the fact that elderly individuals, people with disabilities, or those in a state of panic might require more time and assistance to evacuate safely.

Besides, the model operates with a predefined speed for all occupants, assuming uniform movement throughout the evacuation process. This assumption does not accurately reflect the diverse range of human behaviour and movement in emergencies. People move at different speeds based on their physical condition, the density of the crowd, or the urgency of the situation. As a result, the model's estimate may not fully capture the real dynamics of an evacuation. While useful for providing a rough estimate, the model lacks the complexity needed to offer a truly accurate prediction of evacuation time, highlighting the need for more sophisticated models that incorporate these critical factors.

6.4.3 Model Algorithm

The algorithm currently lacks the ability to manage the order in which people should evacuate or to direct the flow of evacuees in a way that optimizes the evacuation process. Without this guidance, the evacuation may not be as efficient as it could be, potentially leading to longer evacuation times. Moreover, the algorithm does not have mechanisms to prevent people from crowding together, which can create bottlenecks and increase the risk of panic or chaos during the evacuation.

In a situation where large groups of people are trying to exit simultaneously, the lack of structured guidance can lead to confusion and disorganization. This could result in people crushing together at choke points, such as narrow corridors or exits, further complicating the evacuation and potentially causing injuries. The algorithm's current design does not address these challenges, which are critical for ensuring a smooth and orderly evacuation process.

Additionally, the algorithm does not incorporate strategies to separate evacuees into smaller, more manageable groups. By not dividing people into several groups with designated routes, the system misses an opportunity to enhance evacuation efficiency. Grouping and directing evacuees systematically could significantly reduce congestion and ensure that everyone exits the building in a safer and more orderly manner, ultimately shortening the overall evacuation time.

6.4.4 Environment

Due to the limitations of my device has only one camera, testing must be conducted on a one-by-one basis, rather than being able to evaluate multiple cameras simultaneously. This constraint limits the ability to simulate a fully integrated system where multiple cameras could work together to provide comprehensive coverage. As a result, the testing process might not fully capture the complexities of a real-world scenario where multiple cameras would be deployed.

Apart from it, this project assumes that a camera is positioned at every critical node in the building, such as rooms and labelled points to collect enough data from environment. Nevertheless, the reality is that most of the buildings may not be enough CCTV cameras available to cover every area effectively. This limitation leads to potential blind spots where critical data may not be captured, affecting the overall accuracy of the system. The reliance on a high number of cameras for complete coverage can also be seen as a constraint, as it may not be feasible to install such extensive surveillance in real-world applications.

6.5 Future Works

To improve the system's capabilities, I will prioritize on improving the human detection model to enhance both accuracy and the amount of information extracted from the detected individuals. This involves refining the algorithm to better distinguish between overlapping people and those at varying distances, ensuring a more precise count and a clearer understanding of crowd dynamics. The improvements will be critical for providing real-time data that accurately reflects the situation in the building during an emergency.

Another important future development is the segmentation of the building's areas. By dividing the structure into distinct zones, the system can more effectively manages and directs evacuees based on their location within the building. The segmentation helps in identifying the safest and quickest routes for different areas of people and distributing the flow of evacuees in the most efficient way that minimizes the congestion and reduces the possibility of bottlenecks at critical exit points.

Furthermore, I will focus on improving the algorithm to manage the evacuation order more strategically. By controlling the sequence in which different groups of people are instructed to evacuate, the system can optimize the routes calculated by the algorithm, ensuring a more orderly and efficient evacuation. This involves prioritizing certain groups, such as those in danger or those with disabilities, and directing them through the safest and quickest paths available. These advancements will significantly improve the system's ability to coordinate evacuations, impressively improving the safety and reducing the evacuation times during emergencies.

Last but not least, I plan to address the issue of the limited number of cameras in the project by finding a solution to mitigate blind spots caused by the shortage of devices. Instead of placing cameras at every location, I will strategically deploy them at critical points. By utilizing a human detection model, the system will monitor evacuees across different routes by counting the number of people entering and exiting specific areas, ensuring accurate tracking. This approach greatly reduces deployment costs and system workload while retaining effective monitoring.

6.6 Summary

In a nutshell, the Emergency Routing System for Buildings Using Dijkstra’s Algorithm has successfully achieved its objectives by finding out the most efficient escape route based on the real time situation. These results underscore the performance of Dijkstra’s Algorithm on finding the best route timely. However, ongoing refinement and adaptation to evolving more environment variables and evacuees’ condition will be crucial for further elevating the system's performance in a changing building environment during emergency. By pushing the boundaries of Artificial Intelligence (AI) and computer vision, I pave the way for a safer and more efficient future. As I finalize this project, I look ahead to the exciting possibilities that lie in further research, refined designs, and enhanced implementations within the realm of AI based technologies.

REFERENCES

Buchwalow, I. B., and Böcker, W. (2010). Immunohistochemistry: basics and methods. Berlin: Springer Verlag.

Caamaño-Tubío, R. I., Pérez, J., Ferreiro, S., and Aldegunde, M. (2007). Peripheral serotonin dynamics in the rainbow trout (Oncorhynchus mykiss) Comparative Biochemistry and Physiology Part C: Toxicology & Pharmacology. 145(2): 245-255.

Cameron, A. A., Plenderleith, M. B. and Snow, P. J. (1990). Organization of the spinal cord in four species of elasmobranch fishes: cytoarchitecture and distribution of serotonin and selected neuropeptides. The Journal of Comparative Neurology. 297: 201-218.

Desa, M.I (1995). Bus fleet maintenance modeling in a developing country. Ph.D Thesis, University of Salford.

Improve indigenous housing now, government told, 2007. accessed 8 February 2009, <http://www.architecture.com.au/i-cms?page=10220>.

International Narcotics Control Board 1999, United Nations, accessed 1 October 1999, <http://www.incb.org>.

Theusen, G. J. and Fabrycky, W. J. (1984). Engineering Economy. 6th. Ed. Englewood Cliffs, N. J.: Prentice Hall. 150-178.

1) https://jtec.utem.edu.my/jtec/article/view/4331
2) https://iopscience.iop.org/article/10.1088/1742-6596/1803/1/012018/pdf
3) https://www.mdpi.com/2220-9964/9/1/46
4) https://ijpsat.org/index.php/ijpsat/article/download/5172/3218
5) http://refbase.cidis.espol.edu.ec/files/angelyoyola/2017/55_AngelyOyola_etal2017.pdf
6) https://iopscience.iop.org/article/10.1088/1757-899X/854/1/012035
7) https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=b2c1d384fd9b0595b62c7bcfe92110a6ca88890b
8) https://www.myweb.ttu.edu/slim/lab/papers/min_syscon_16.pdf
9) https://www.hindawi.com/journals/mpe/2021/3017607/
10) https://thescipub.com/pdf/jcssp.2015.330.336.pdf
11) https://www.ijeat.org/wp-content/uploads/papers/v9i5/A1775109119.pdf
12) https://www.tandfonline.com/doi/pdf/10.1080/23311916.2024.2319394
13) https://iopscience.iop.org/article/10.1088/1742-6596/1185/1/012022/pdf
14) https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8507583/
15) https://journal.ugm.ac.id/v3/JNTETI/article/view/2201
16) https://www.e3s-conferences.org/articles/e3sconf/pdf/2023/46/e3sconf_icmsem2023_06009.pdf
17) https://www.researchgate.net/publication/362769286_Optimal_Evacuation_Route_Planning_of_Urban_Personnel_at_Different_Risk_Levels_of_Flood_Disasters_Based_on_the_Improved_3D_Dijkstra's_Algorithm
18) https://www.proquest.com/openview/c11c56dfe072e3c28d01c7ef808a80d4/1?pq-origsite=gscholar&cbl=2032387
19) https://people.cs.nycu.edu.tw/~yushuen/data/Evacuation17.pdf
20) https://www.mdpi.com/2227-7390/10/17/3146
21) https://www.researchgate.net/publication/349942676_Deep_adaptive_learning_for_safe_and_efficient_navigation_of_pedestrian_dynamics
22) https://www.mdpi.com/2076-3417/13/9/5812
23) https://github.com/datitran/object_detector_app/tree/master/object_detection
24) https://www.researchgate.net/publication/221417598_Real-Time_Pedestrian_Evacuation_Planning_During_Emergency
25) https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9320030/

26) https://ieeexplore.ieee.org/document/9306403
27) https://www.sciencedirect.com/science/article/pii/S0379711221001727
28) https://ieeexplore.ieee.org/document/10130099
29) https://www.freecodecamp.org/news/dijkstras-shortest-path-algorithm-visual-introduction/
30) https://www.researchgate.net/publication/221417598_Real-Time_Pedestrian_Evacuation_Planning_During_Emergency
31) https://www.simplilearn.com/tutorials/data-structure-tutorial/bellman-ford-algorithm
32) https://www.intechopen.com/online-first/87285
33) https://www.geeksforgeeks.org/a-search-algorithm/
34) https://www.geeksforgeeks.org/floyd-warshall-algorithm-dp-16/
35) https://towardsdatascience.com/introduction-to-genetic-algorithms-including-example-code-e396e98d8bf3
36) https://www.mdpi.com/2075-5309/13/12/2907

APPENDIX

Sample of Data

1	Num of ppl	Area	Distance	Max ppl	Ppl density	Width	Time
2	12	4	5	4	3	0.8	5.5
3	16	4	5	4	4	0.8	6.5
4	6	4	5	4	1.5	0.8	4
5	5	4	5	4	1.25	0.8	3.75
6	3	2.1	1.4	2	1.5	1.5	3.51
7	19	4	2	4	4.75	2	12.88
8	4	3.4	1.7	3	1.33333	2	4.18
9	20	4	2	4	5	2	13.5
10	13	11.76	4.9	11	1.18182	2.4	6
11	8	5.94	2.2	5	1.6	2.7	6.5
12	33	12.72	5.2	12	2.75	2.44615	11.01
13	16	8.76	7.3	8	2	1.2	6.65
14	2	1.8	1.5	1	2	1.2	3.75
15	3	1.8	1.2	1	3	1.5	6.22
16	106	29.04	12.1	29	3.65517	2.4	17.02
17	6	11.76	4.9	11	0.54545	2.4	4.09
18	15	4.29	3.3	4	3.75	1.3	7.74
19	1	0.91	0.7	1	1	1.3	1.98
20	50	11.28	9.4	11	4.54545	1.2	11.52
21	10	8.16	6.8	8	1.25	1.2	5.27
22	3	3.24	2.7	3	1	1.2	2.85
23	48	10.8	9	10	4.8	1.2	11.7
24	48	11.16	9.3	11	4.36364	1.2	11.2
25	3	1.36	1.7	1	3	0.8	3.85
26	12	8.4	5.6	8	1.5	1.5	5.61
27	45	9.84	8.2	9	5	1.2	11.6
28	18	12.24	10.2	12	1.5	1.2	7.35
29	23	8.7	5.8	8	2.875	1.5	8.29
30	0	7.8	6.5	7	0	1.2	3.25
31	6	4.44	3.7	4	1.5	1.2	4.1