FINDING AN OPTIMAL ROUTE PATH FOR AMBULANCE

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering degree in Computer Science and Engineering

By

CHAITANYA BALAJI.I (Reg.No - 39110385)
POORNA SIVA SAI. VEJANDLA (Reg.No – 39111073)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by NAAC | 12B Status by UGC | Approved by AICTE
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI - 600119

APRIL - 2023

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by NAAC | 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **CHAITANYA BALAJI INAGANTI (39110385) and POORNA SIVA SAI (39111072)** who carried out the Project Phase-2 entitled **"Finding Optimal Route Path for an Ambulance"** under my supervision from January 2023 to April 2023.

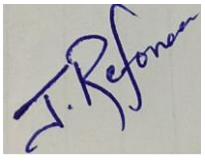
Internal Guide

Dr. Sujihelen L M.E., Ph.D.

Head of the Department

Dr. L. LAKSHMANAN, M.E., Ph.D.

Submitted for Viva voce Examination held on13.4.2023



Internal Examiner

External Examiner

DECLARATION

I, **Chaitanya Balaji (Reg.No- 39110385),** hereby declare that the Project Phase-2 Report entitled **"Finding Optimal Route Path for an Ambulance"** done by me under the guidance of **Dr. Sujihelen L, M.E., Ph. D** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in **Computer Science and Engineering**.

DATE: 13-04-2023

Chaitanya Balaji

PLACE: Chennai **SIGNATURE OF THE CANDIDATE**

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph. D**, **Dean**, School of Computing, **Dr. L. Lakshmanan M.E., Ph.D.,** Head of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. Sujihelen L M.E., Ph. D,** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my phase-2 project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

With the world's population growing, distractions can cause dangerous accidents that are difficult to avoid. Several studies have been done in an effort to find a way to stop the loss of life. The protection of the passengers is the first concern of the vehicle's designers, and airbags are designed to improve security and save the lives of occupants, but they do not stop accidents from occurring. The main reasons are general fatigue and distraction brought on by phone alerts. A wide range of techniques based on behavioral measures using machine learning techniques has been inspected to scope out driver distraction in the past. The recent growth of such technologies requires that these algorithms be improved to evaluate their accuracy in identifying distraction. There are numerous features of faces that are available to be extracted from any face to deduce the level of distraction. These include eyes being closed for longer than 5 seconds, head movements, and continuous yawning. However, the development of a shock system to push out the distraction and immediately give an alert is a challenging task as it requires accurate and robust algorithms. This study takes a novel approach by using convolution neural networks to scope out the distraction and will instantly sound a loud siren to give a sensory shock which brings back alertness. When paired with proper guidance, the said hybrid approach would produce the best solution in real-time to such issues in the future.

TABLE OF CONTENTS

.

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

In today's world, fortunately, it is possible to cure many diseases and injuries with the development of medical technology. However, obtaining timely aid is crucial for the patient, especially in accidents. Under such circumstances, an ambulance reaching the event scene in a timely manner has a significant economic and social influence. There has been news reporting the death of patients due to the untimely rescue of ambulances.Issues such as the untimely departure of the ambulance, being unfamiliar with the routes, and delays caused by congested traffic are the main reasons resulting in this. For example, statistics indicate that the chance of survival decreases by 24% for every additional minute of delay in the treatment of cardiac arrest patients. Hence, seeking an effective path, especially in congested traffic, for ambulances is one of the notably demanding research areas because of its vital and direct impact on human lives.

With the increase in city size, the emergency rescue system is playing an important role in the safety of human life and social security. The Emergency Medical Services (EMS) ambulance is designed to give medical care or treatments to patients at the emergency sites and/or directly send the patients to hospitals where intensive care by doctors can be given. The EMS ambulance provided is equipped with basic life support equipment to help whenever an emergency occurs.Following protocols and guidelines, EMS provides emergency care and ambulance to the patients and involves in the patient's life or death. Moreover, EMS also provides non-emergency standby duties with minimal charges, for instance, festivals, sports, motor racing, national and international conferences, duties during aircraft emergency landings or crashes, and transfer of patients from hospitals to other hospitals.

In order, to ensure the ambulance can arrive at the patient within the target time, ambulance availability must be ensured and the time taken to arrive must be controlled. Thus, in order to help improve the ambulance system, efficient ambulance management and system are required to increase the standard of EMS

In many instances vehicles and hence users' transit time within the network are negatively affected because of delays occasioned by traffic lock jams. In particular, in day-to-day life, an Emergency vehicle or a Fire vehicle may get stuck aggravating the precarious situation of individuals on board for treatment or early intervention at scenes where lives and properties may be saved. There exist various factors affecting the travel times of emergency vehicles (emergency vehicles).

The inevitable factors are the traffic condition at different times of the day, and random congestion (e.g. due to accidents). Traffic congestion is one of the most fundamental problems in large cities like Lagos. There is a positive connection between the delay of emergency vehicles and the ratio of fatal or serious preventable incidences. These untoward losses can be minimized by deploying intelligent decision systems and shortest path algorithms to optimize ambulance travel time to and from accidents scene to the nearest hospitals

At this point, as a sub-field of transportation in health care, seeking effective emergency medical services (EMS) management plan has been widely studied. Since those plans have a direct effect on human lives, many researchers are dedicated to optimizing them by satisfying all the constraints at the best point, such as minimizing responsiveness time, maximizing availability, and so on. Such a large number of patients in India lose their lives because the ambulance does not arrive on time or delay in the treatment. Also, in view of the increasing technology and other factors, it is necessary to be aware of such technologies to reduce this problem.

CHAPTER 2
LITERATURE SURVEY

[1] Taha Darwassh Hanawy Hussein, Mondher Frikha, Sulayman Ahmed, Javad Rahebi, “Ambulance Vehicle Routing in Smart Cities Using Artificial Neural Network”, 6th International Conference on Advanced Technologies for Signal and Image Processing (ATSIP), 2022

Taha Darwassh Hanawy Hussein proposed an AI-based neural network that helps in routing ambulances. Eight values are given as input for the neural network. These values include the time of the accident, accident location, ambulance location, hospital location, number of streets, number of people injured, accident type, and ages of the people involved in the accident. Analyzing these values, the ambulance can decide on the shortest route to the nearest hospital. Also, this paper evaluates the critical factors during an accident such as forming temporary emergency teams, the number of available ambulances, as well as the resources and responsiveness of the city.

[2] Taha Darwassh Hanawy Hussein; Mondher Frikha; Sulayman Ahmed; Javad Rahebi, “Ambulance Vehicle Routing using BAT Algorithm”, International Conference on Electrical, Communication, and Computer Engineering (ICECCE), 2021

Taha Darwassh Hanawy Hussein proposed to find an efficient route for ambulances using the BAT algorithm. The node method is used to create the map of the city. At first, the control station reports the accident location to the ambulance and the hospital. Then the driver feeds the data—node position of the accident and ambulance vehicle—into the BAT algorithm which detects the shortest path to reach the accident scene.

[3] Elgarej Mouhcine, Yassine Karouani, Khalifa Mansouri, Youssfi Mohamed, “Toward a distributed strategy for emergency ambulance routing problem”, 4th International Conference on Optimization and Applications (ICOA), 2018

Elgarej Mouhcine proposed to develop a distributed approach to the problem of finding the optimal emergency route for the ambulance. The primary factor of rescue operations in cities has always been detecting the optimal route. A distributed approach is suggested based on the model of the ant colony system algorithm.

Considering the various issues like speed limit, traffic, available ambulances, and location of the hospital, this paper introduces a distributed model to find the optimal path that reduces the time significantly.

4] Tressa Michael, Deepthy Xavier, “Intelligent Ambulance Management System with A Algorithm”, 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), 2020

A lot of hours are wasted due to traffic in contemporary society. It becomes a huge problem for ambulances and other emergency vehicles. Every minute wasted comes at the cost of precious lives. To tackle this dreadful situation, Tressa Michael has proposed using the A* algorithm to can find the shortest path for the ambulance. This algorithm takes traffic congestion into account and it produces a dynamic route map as per the intensity of the traffic. Once an accident occurs, the location of the accident is sent to the ambulance and the shortest path is shown on the ambulance dashboard. This helps to reach the location faster.

[5] Mohamed N. Ashmawy, Ahmad M. Khairy, Mohamed W. Hamdy, Anas El-Shazly, Karim El-Rashidy, Mohamed Salah, Ziad Mansour, Ahmed Khattab, “SmartAmb: An Integrated Platform for Ambulance Routing and Patient Monitoring”, 31st International Conference on Microelectronics (ICM), 2020

Mohamed N. Ashmawy proposed a single platform that integrates ambulance routing and patient monitoring. The platform aims at dual benefits: increasing the chance of survival of the patient by soon arriving at the hospital and allowing a doctor to check the patient’s biomedical data while in transit. The latter helps to do the necessary preparation before the patient arrives at the hospital. Additionally, the platform uses machine learning methods on the data gathered to assist the doctor in identifying potential medical risks.

[6] Abdel-Ghani Karkar, “Smart Ambulance System for Highlighting Emergency-Routes”, Third World Conference on Smart Trends in Systems Security and Sustainability (WorldS4), 2019

During a critical situation, paramedics must act quickly and carry patients swiftly even through heavy traffic. But the traffic and congestion affect the speed of the ambulance significantly. The problem is that drivers react only when the ambulance is near to them but they fail to give way to the ambulance when it is farther away. So

in this paper, AbdelGhani Karkar has proposed a smarter ambulance system that differs from the prevailing systems by its method of alerting drivers about the emergency routes taken by the ambulance. The system has two applications: user emergency application and paramedical application. The former shows the location of the patient and the location of the ambulance(s). And the latter locates the patient and finds the nearest hospital.

4] Tressa Michael, Deepthy Xavier, “Intelligent Ambulance Management System with A Algorithm”, 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), 2020

A lot of hours are wasted due to traffic in contemporary society. It becomes a huge problem for ambulances and other emergency vehicles. Every minute wasted comes at the cost of precious lives. To tackle this dreadful situation, Tressa Michael has proposed using the A* algorithm to can find the shortest path for the ambulance. This algorithm takes traffic congestion into account and it produces a dynamic route map as per the intensity of the traffic. Once an accident occurs, the location of the accident is sent to the ambulance and the shortest path is shown on the ambulance dashboard. This helps to reach the location faster.

[5] Mohamed N. Ashmawy, Ahmad M. Khairy, Mohamed W. Hamdy, Anas El-Shazly, Karim El-Rashidy, Mohamed Salah, Ziad Mansour, Ahmed Khattab, “SmartAmb: An Integrated Platform for Ambulance Routing and Patient Monitoring”, 31st International Conference on Microelectronics (ICM), 2020

Mohamed N. Ashmawy proposed a single platform that integrates ambulance routing and patient monitoring. The platform aims at dual benefits: increasing the chance of survival of the patient by soon arriving at the hospital and allowing a doctor to check the patient’s biomedical data while in transit. The latter helps to do the necessary preparation before the patient arrives at the hospital. Additionally, the platform uses machine learning methods on the data gathered to assist the doctor in identifying potential medical risks.

[6] Abdel-Ghani Karkar, “Smart Ambulance System for Highlighting Emergency-Routes”, Third World Conference on Smart Trends in Systems Security and Sustainability (WorldS4), 2019.During a critical situation, paramedics must act quickly and carry patients swiftly even through heavy traffic. But the traffic and

congestion affect the speed of the ambulance significantly. The problem is that drivers react only when the ambulance is near to them but they fail to give way to the ambulance when it is farther away. So in this paper, AbdelGhani Karkar has proposed a smarter ambulance system that differs from the prevailing systems by its method of alerting drivers about the emergency routes taken by the ambulance. The system has two applications: user emergency application and paramedical application. The former shows the location of the patient and the location of the ambulance(s). And the latter locates the patient and finds the nearest hospital.

7] Subash Humagain, Roopak Sinha, “Routing Autonomous Emergency Vehicles in Smart Cities Using Real-Time Systems Analogy: A Conceptual Model”, IEEE 17th International Conference on Industrial Informatics (INDIN), 2020

A short response time is crucial for emergency vehicles (EVs) like ambulances, fire, police, etc. The congestion, numerous intersections with signals, and long queues of vehicles are major hindrances to emergency vehicles. The prior solutions to route EVs failed as they do not accustom to dynamic traffic parameters. Real-time data on hindrances is crucial to reduce the time and impact of EV movement on other traffic. So Subash Humagain has proposed using autonomous emergency vehicles (AEVs) that can drive faster and safer through complex road networks by taking real-time decisions based on live data. This is made possible by taking real-time systems (RTS) where the mixed-criticality real-time system (MCRTS) is used to schedule AEVs tasks for achieving critical response time.

[8] Mohammad A. R. Abdeen, Mohamed Hossam Ahmed, Hafez Seliem[, Tarek Rahil Sheltami, Turki M. Alghamdi, Mustafa El-Nainay, “A Novel Smart Ambulance System—Algorithm Design, Modeling, and Performance Analysis”, IEEE Access (Volume: 10), 2022

Mohammad A. R. Abdeen proposed a novel system that uses modern communication, processing, and sensing technologies to transform emergency and ambulance services. The smart system uses road traffic conditions and hospital loading information to make the best course of action. The performance of the algorithm is analyzed both analytically and by simulation for verification. The results produced excellent consistency between simulation and analytical analysis—confirming the accuracy of the analysis. When compared with prior algorithms

reported in the literature, the smart algorithm stood superior under considered operating conditions and scenarios.

9] Myint Myint Sein, K-zin Phyo, Mau Luen Tham, Yasunori Owada, Nordin Bin Ramli, Suvit Poomrittigul, “Effective Evacuation Route Strategy for Emergency Vehicles”, IEEE 10th Global Conference on Consumer Electronics (GCCE), 2021
Rescue teams must arrive at the incident site as soon as possible to minimize risk and damage. The Dijkstra algorithm is used to develop an effective evacuation route predicting algorithm for a complex unstructured road network. The proposed system will calculate the best evacuation route for emergency vehicles and services such as fire trucks, ambulances, and police cars. It is also provided to search the nearby relief area and guide to the safe location using the best evacuation route assessment. The closest emergency service is calculated using the surround services system. The haversine distance is used to calculate the distance between two points. After estimating the nearest emergency services, the optimal route from the service center to the incident location is computed.

[10] Nikki Rathore, Pramod Kumar Jain, Manoranjan Parida, “A Routing Model for Emergency Vehicles Using the Real Time Traffic Data”, IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI), 2018
One of the several semi-government emergency medical services (EMS) that operate in India's many states is 108. The prompt response to demand and incorporation of real-time travel and traffic data into the vehicle timing and routing model are key components of EMS efficiency. This study uses the Google Maps Distance Matrix API to create an optimization technique based on real-time live traffic data that is used to calculate the routes for emergency vehicles. The vehicle routing problem is formulated using an integer programming model in the heuristic technique, and it is then optimized using Google API. The model's primary elements include dispatch optimization, shortest route finding, accident location, and vehicle tracking.

2.1 INFERENCES FROM LITREATURE SURVEY

From the above-mentioned literature works, it is clear that there has been effective research on Ambulance optimal path finding and many models have been proposed.

- It is evident that the above-mentioned systems have their own pros and cons.
- While some of the recent works involve hybrid technologies and provide better accuracies, they are still far from what is needed.
- With higher accuracy, comes the need for low computational costs, high processing speed, and most of all, convenience of use.
- After going through this refrences it is clear that some algorithms are really good at finding the evacuation route and some other are good are finding an optimal path.

YEAR	DETAIL	ALGORITHM/TECHNIQUE	FEATURES	LIMITATIONS
2022	Ambulance Vehicle Routing in Smart Cities Using Artificial Neural Network	Artificial neural network	The proposed system route detects the route faster than other systems on this issue.	Vehicle identification systems are very challenging when it comes to lowering the false positive rate for vehicles that are blurry.
2021	Ambulance Vehicle Routing using BAT Algorithm	BAT algorithm	Shortest path and quick reach time is achieved using this algorithm	The proposed system can generate better results and accuracy is not high when compared to other models
2018	Toward a distributed strategy for emergency ambulance routing problem	ant colony system algorithm	The response time taken by proposed system is less compared to others.	Quality must be raised by including more systems in the architecture
2020	Intelligent Ambulance Management System with a Algorithm	A*algorithm	We can determine the shortest path with the modified A*	It is expensive and every action associated with the a star

			method in a limited number of iterations.	algorithm has fixed cost
2020	SmartAmb: An Integrated Platform for Ambulance Routing and Patient Monitoring	Artificial neural networks	The accuracy of the proposed system is high when SVM algorithms are used and computational time is low	Large database is not integrated in the system which needs to be overcome
2019	Smart Ambulance System for Highlighting Emergency-Routes	Mobile application operation	It is different from the current systems in how it informs drivers of the ambulances emergency route	The proposed system makes no mention of its testing applicability with drivers and paramedics in actual situations
2020	Routing Autonomous Emergency Vehicles in Smart Cities Using Real Time Systems Analogy: A Conceptual Model	Mixed Criticality Real Time System Concept	In this system many highly refined scheduling algorithms are used which helps in reducing response time.	The system does not contain large no.of data sets so it could not work in critical situations
2022	A Novel Smart Ambulance System—Algorithm Design, Modeling, and Performance Analysis	SDF routing policy, event driven dynamic simulation	The results demonstrated excellent consistency between the simulation and analytical results, confirming the precision and validity of the analysis.	The response time taken by the system is quite high which needs to be taken into consideration
2021	Effective Evacuation Route Strategy for	Haversine distance	the proposed solution provides solution even	The computational time taken by the system is

	Emergency Vehicles		for complex unstructured road	quite high which needs to be reduced.
2018	A Routing Model for Emergency Vehicles Using the Real Time Traffic Data	Google Maps Distance Matrix API	A number of samples and tests are used to validate this simulation technique.	Many other complex real life situations are not considered which makes it less efficient

2.1 Literature Survey Table

2.2 OPEN PROBLEMS IN EXISTING SYSTEM

There exists a wide variety of algorithms that are capable of solving the shortest path problem.However, only several of the most popular conventional shortest path algorithms are as follows:

1. Floyd-Warshall Algorithm
2. Bellman-Ford Algorithm
3. Genetic Algorithm (GA)

•The existing systems are slightly less robust in prediction. They have not evolved according to technological advancements and time.

•The results are also a bit unsatisfactory and may lead to an erroneous prediction of the optimal path for the ambulance leading to unwanted delays in the arrival of patients to the hospital.

•A slight feature that is missing but which can be mostly significant in some situations is the audio of directions.

CHAPTER 3
REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDIES/RISK ANALYSIS OF THE PROJECT

The risk is pretty high when we don't use a secure portal for this, since it will be accessible to all. That is the reason we have developed a system where a secure web interface is built. It is highly feasible to develop this project further and it is also easy to maintain the database with pictures and details.

3.2 SOFTWARE REQUIREMENTS SPECIFICATION DOCUMENT

Hardware specifications:

- Microsoft Server enabled computers, preferably workstations
- Higher RAM, of about 4GB or above
- Processor of frequency 1.5GHz or above

Software specifications:

- Python 3.6 and higher
- Anaconda software
- Streamlit
- Google maps api

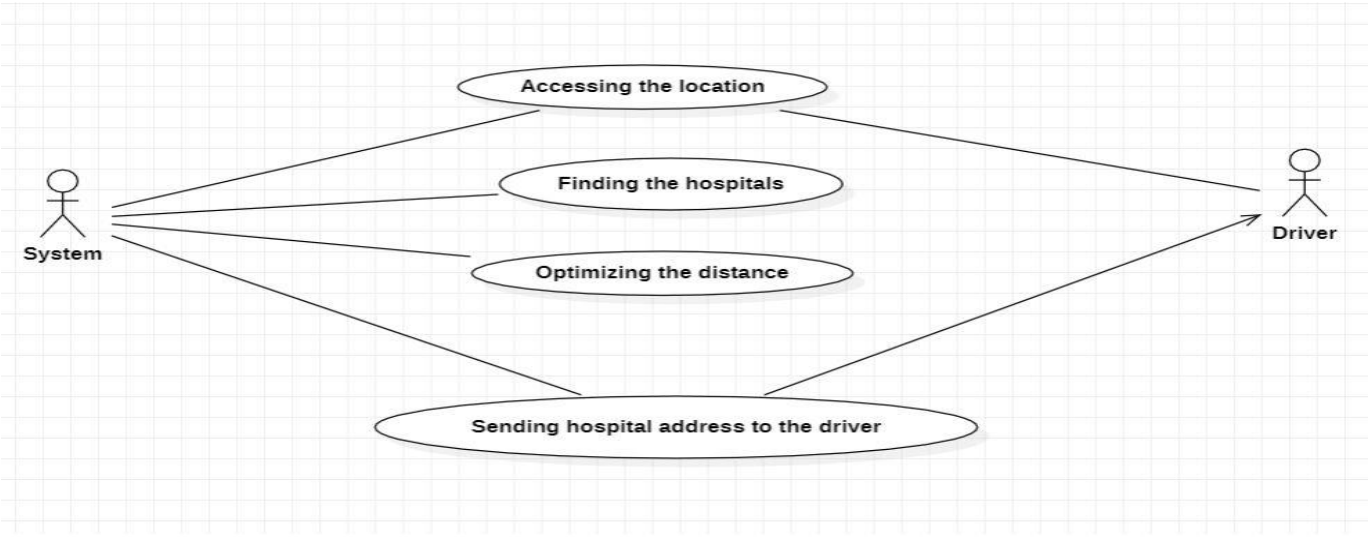
3.3 USECASE DIAGRAMS

A use case diagram is a visual representation of the interactions between a system and its actors. In the case of finding an optimal path for an ambulance and sending a notification to the driver, the actors involved may include the ambulance driver, the emergency medical services (EMS) dispatcher, and possibly even other drivers on the road who may need to be alerted of the ambulance's presence.

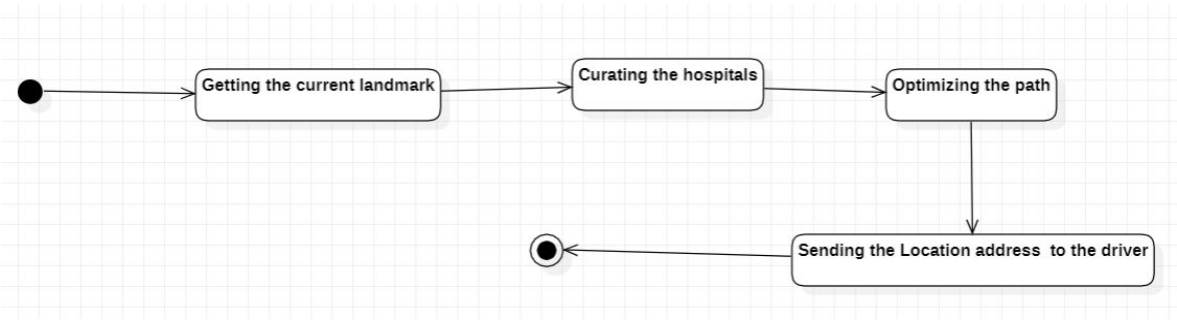
One possible use case scenario for this system might be as follows:

1.The EMS dispatcher receives a call reporting an emergency and dispatches an ambulance to the scene.
2.The ambulance driver receives a notification on their mobile device indicating the location of the emergency and the optimal route to take.
3.The ambulance driver follows the route provided by the system, which takes into account factors such as traffic congestion, road closures, and any other obstacles that may affect the journey.
4.Other drivers on the road receive a notification on their mobile devices indicating that an ambulance is approaching, and they are advised to clear the way.
5.The ambulance arrives at the scene of the emergency as quickly and safely as possible, using the most efficient route available.
In this use case diagram, the system would be represented as a central box labeled "Ambulance Routing and Notification System," with arrows pointing out to the various actors involved in the process. The EMS dispatcher, ambulance driver, and other drivers on the road would each be represented by their own boxes, labeled with their respective roles.

The use case diagram could also include additional scenarios, such as what happens if the ambulance encounters unexpected road closures or traffic congestion during its journey. These scenarios would help to ensure that the system is robust and able to handle a variety of real-world situations.



3.3 Usecase diagram



3.3.1Activity Diagram

CHAPTER 4
DESCRIPTION OF PROPOSED SYSTEM

A system which takes the speech from the user through Streamlit which is used to record the audio, And this audio can be processed using the “recognize google” python library. By using the Geocoders and OpenStreetMap we will get the details of the nearest hospitals and their exact location and within these hospitals the nearest hospital can be addressed using the “get_nearest” function i.e. used in the programming of the code. After getting the location, for sending the hospital details to the user we have used the Twilio platform. Through this platform we can send the details of the hospital to the user in the form of text message.

4.1 SELECTED METHODOLOGY OR PROCESS MODEL:

The following methodology discussed below is being used for the model.

4.1.1 Searching using Streamlit created webpages:

Developers may easily construct interactive web apps thanks to the open-source Python framework known as Streamlit. It may be used for a variety of activities, including data exploration and visualization, machine learning, and natural language processing. It is intended to make the process of developing data-driven web apps more straightforward. This streamlit component allows to register an audio utterence from a user.The ability of Streamlit to dynamically update and refresh the user interface as data or settings are modified is one of its main advantages. Without having to manually reload the website, this enables developers to rapidly iterate and experiment with their apps.

Overall, Streamlit is a powerful tool for building data-driven web applications, particularly for data scientists and developers who want to quickly prototype and test new ideas.

4.1.2 Audio recording using mic:

The Speech Recognition API from Google is excellent. Simply said, this API transforms spoken words from a microphone into written text written as Python strings. Google API will automatically convert any spoken words you make into

printed form. For the English language, the API yields great results.By offloading the logic, a voice recognition API enables you to submit a web request to the API and receive the text that was recognized in response. Although you may perform this straight from Python code, your script will also need to have access to the internet.
The speech recognition module, which will be added on top of the application, is used to record the audio. The Google voice recognition API is then used to process the recorded speech and deliver the results.
r.recognize_google(audio) returns a string.

4.1.3. Distance key using Open StreetMap:

A collaborative initiative called OpenStreetMap (OSM) aims to provide a free, editable map of the whole planet. Since its founding in 2004, it has developed into one of the most complete and frequently utilized sources of geographic data.The OSM project is dependent on volunteers to add and change data on the roads, structures, landmarks, and other objects in their local area. Then, under the terms of an open license, this data is made available to everyone and preserved in a public database.

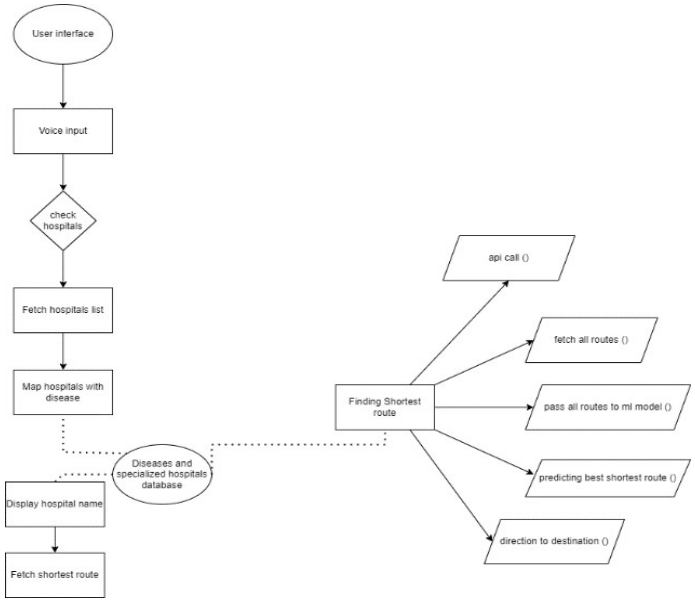
OSM offers tools and APIs that let programmers access and utilize this data in a number of ways. For instance, programmers may interact with OSM data programmatically by using the OSM API to access map data and build bespoke map apps, or they can utilize pre-built libraries like osmapi, osmnx, or overpass-api.

4.1.4. Sending message using Twilio platform:

Twilio is a cloud communication platform that enables developers to create and include communication services into their apps, such as sending and receiving emails, SMS messages, and phone calls.
A variety of APIs and SDKs are available on the Twilio platform, which makes it simple for programmers to include communication features into their apps. For instance, the Programmable Voice API from Twilio may be used to place and receive phone calls, while the Programmable SMS API from Twilio can be used to send and receive SMS messages.
Additionally, Twilio offers a selection of pre-built tools and interfaces that make it simple to include communication capabilities into well-liked web development frameworks and platforms, like Node.js, Ruby on Rails.

4.2 ARCHITECTURE / OVERALL DESIGN OF PROPOSED SYSTEM

The architectural diagram clearly shows the collection of data starts from the data collection then it maps the road network set up and visualizes its features in the map. This initialization is being done using Folium technology. Here, the geographic locations and co-ordinates are being gathered, then the hexagons in the A* algorithm are located with which a time travel API request is being generated. Then a N-path which refers to a novel path is being generated here on the map. After then, we apply the A* Algorithm to the whole data available. Here, selection of the point node happens then the shortest path is being created.



4.2 System Architecture

4.3 DESCRIPTION OF SOFTWARE FOR IMPLEMENTATION AND TESTING PLAN OF THE PROPOSED SYSTEM

The project basically combines with machine learning and web development. So, the software include here are the front end and backend along with the machine learning algorithms. The backend part includes the modelling technique used here which is the Streamlit and google maps api used to store the elements. The front-end part refers to the HTML, CSS and JavaScript languages. The language where the machine learning part takes place is the python.

4.3.1 Python:

- Python 2.6 or higher is required for the installation of the Flask. You can start by import Flask from the flask package on any python IDE. For installation on any environment, you can click on the installation link given
- To test that if the installation is working, check out this code given below.
- Python's development is conducted largely through the *Python Enhancement Proposal* (PEP) process, the primary mechanism for proposing major new features, collecting community input on issues, and documenting Python design decisions. Python coding style is covered in PEP 8. Outstanding PEPs are reviewed and commented on by the Python community and the steering council.
- Enhancement of the language corresponds with the development of the CPython reference implementation. The mailing list python-dev is the primary forum for the language's development. Specific issues are were originally discussed in the Roundup bug tracker hosted at by the foundation. In 2022, all issues and discussions were migrated to GitHub. Development originally took place on a self-hosted source-code repository running Mercurial, until Python moved to GitHub in January 2017.
- A web service is defined as a program that can transform any client application to a web-based application. The development of web services significantly depends on the scripting abilities of a programming language. An important aspect of Python language is that it can be used in scripting, implying that it is an effective development tool for web services. Web services developed by Python function using the standard messaging formats and they can be interfaced with other software development tools using the conventional Application Programming Interface (API).

- Web programming using Python entails two major paradigms: server programming and client programming (Beazley 90). Server programming entails the development of web services that run on the web server, while

client programming entails the development of web services that that run on the client side (Hetland 90). There are diverse approaches to server programming, examples include the WebFrameworks used for development of server-side services using Python; CgiScripts used to write scripting applications in Python; Webservers, which are server solutions developed using Python.

- Web services developed in Python are primarily used to offer access and functionality of the APIs via the web. On the client side, Python language can be used in a number of ways including Web Browser Programming, Web Client Programming and Web services. There are various libraries in Python for the development of web services; examples include the Simple Object Access Protocol (SOAP) and the Web Services Description Language (WSDL).

4.3.1.1 Speech Recognition using python

The speech_recognition module in Python is a powerful tool for converting human speech into text. It provides an easy-to-use API for accessing different speech recognition engines, making it possible to perform speech recognition in a variety of different languages.

- One of the main benefits of using the speech_recognition module is that it is highly customizable. Users can choose from a range of different recognition engines, adjust parameters such as the input source and the sensitivity of the recognizer, and even define their own custom recognition models.

- The module can be used for a wide range of applications, from creating voice-activated interfaces for home automation systems, to transcribing audio files for research purposes. It is also commonly used in the development of speech recognition systems for people with disabilities.

- Overall, the speech_recognition module is a powerful tool for working with human speech in Python, and is an important resource for anyone interested in developing speech recognition applications.

4.3.1.1Speech Recognition using python

- The speech recognition module in Python works by first capturing an audio input, either from a microphone or from an audio file. It then applies a recognition engine to analyze the input audio and convert it into text.The recognition engine used can be one of several available, including Google Speech Recognition, Microsoft Bing Voice Recognition, and Sphinx. The choice of engine can be determined by the user, depending on their specific requirements.

- Once the audio input has been processed by the recognition engine, the module returns a transcription of the audio as a text string. The accuracy of the transcription can be affected by a range of factors, such as the quality of the audio input and the specific recognition engine being used.

Overall, the speech recognition module in Python provides a convenient way to convert audio input into text output, making it possible to perform a range of different speech recognition tasks.

4.3.2 Streamlit:

Streamlit is a Python library that allows developers to build and deploy web applications quickly. It is designed specifically for data scientists and machine learning engineers who need to create interactive data-driven applications. The

library provides a simple and intuitive way to build web applications, allowing developers to focus on the logic of their applications rather than the design. Streamlit is an open-source library that is free to use and has a growing community of developers who contribute to its development.

- One of the main benefits of Streamlit is its ease of use. The library provides a straightforward way to create web applications without requiring developers to learn complex web development technologies. Streamlit uses a simple Python syntax to create UI elements such as buttons, sliders, and forms. The library also includes a set of pre-built widgets that can be used to create interactive data visualizations. This allows developers to quickly create web applications without spending a lot of time on design.

- Another advantage of Streamlit is its speed. The library is built on top of Python and uses a reactive programming model to update the UI in real time. This means that changes made to the code are immediately reflected in the web application without the need to restart the server. This allows developers to quickly iterate on their code and test new features in a matter of seconds.

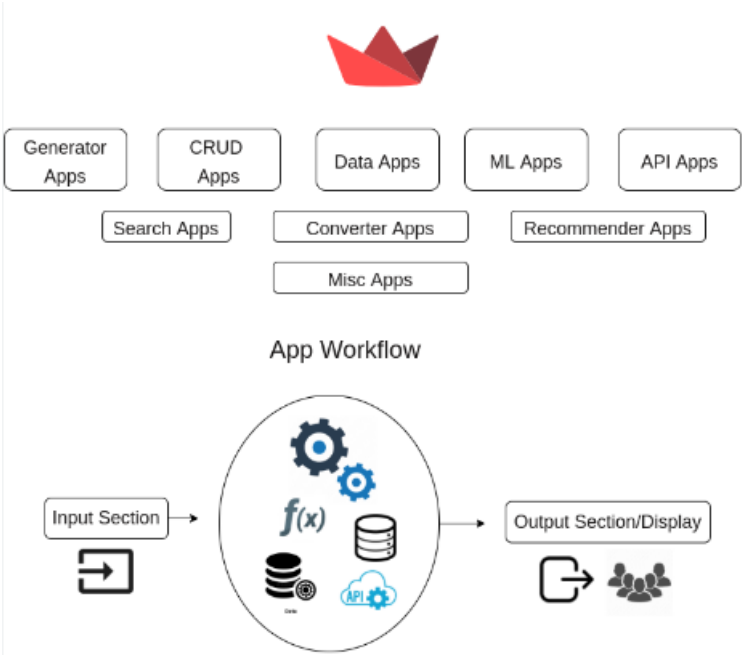
- Streamlit also supports a wide range of data visualisation libraries, including Matplotlib, Plotly, and Bokeh. This allows developers to easily create interactive visualisations and dashboards that can be used to explore and analyse data. Streamlit provides built-in support for machine learning libraries such as TensorFlow and PyTorch, making it easy to build and deploy machine learning models.

- One of the most exciting features of Streamlit is its ability to integrate with other web frameworks and services. The library can be used to create standalone web applications or integrated with existing web services such as Flask or Django. Streamlit also supports deployment to cloud platforms such as Heroku and AWS, making it easy to deploy applications to production environments.

- Streamlit provides a simple and intuitive way to create user interfaces. The library includes a set of pre-built widgets that can be used to create forms, buttons, sliders, and other UI elements. These widgets can be easily customized to suit the needs of the application. Streamlit also includes a number of layout options, allowing developers to organize their UI elements in a way that makes sense for their application. Another advantage of Streamlit is its support for collaboration. The library includes a feature called "Shareable Apps" that allows developers to share their applications with others. This makes it easy for teams to collaborate on projects and share their work with stakeholders. Shareable Apps can be deployed to the cloud or shared as standalone applications.

- The library is well-suited for data exploration and analysis, making it an ideal tool for data scientists and analysts. Streamlit provides a number of features that make it easy to deploy applications to production environments. The library includes built-in support for deployment to cloud platforms such as Heroku and AWS. Streamlit also provides a number of performance optimizations that can help to ensure that applications run smoothly in production environments.

In conclusion, Streamlit is a powerful tool for data scientists and machine learning engineers who need to quickly develop and deploy web-based data-driven applications. The library provides a simple and intuitive way to create web applications without requiring developers to learn complex web development technologies.

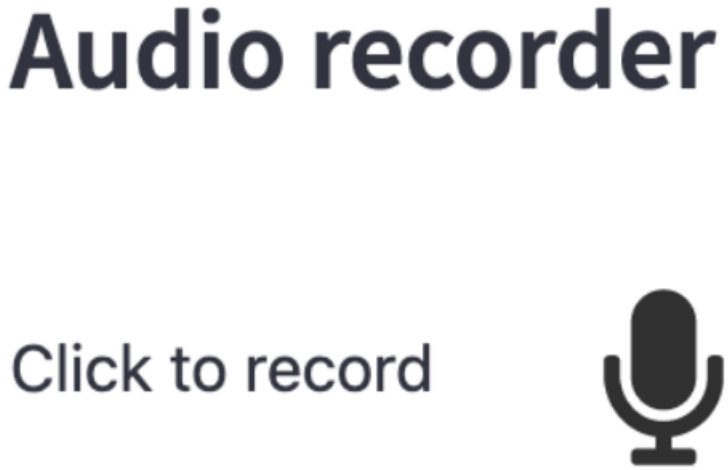


4.3.2 Streamlit

4.3.2.1Streamlit audio recorder:

It is a built-in widget of the Streamlit library that allows users to record audio directly from their web browser.It uses the Web Audio API and WebRTC technology to record audio in the user's web browser, without requiring any additional software or plugins.The recorded audio is saved to a temporary file on the server, and can be processed further using Python libraries such as SoundFile.The widget provides a simple interface for users to start and stop recording, and displays a timer indicating the length of the recording.The quality of the recorded audio may vary depending on factors such as the user's microphone and internet connection.

The widget can be customized with parameters such as the recording duration, audio format, and output file name.Streamlit Audio Recorder can be used for a variety of applications, such as speech recognition, voice analysis, and audio annotation.



4.3.2.1Streamlit Audio Recorder

As with all Streamlit widgets, the Audio Recorder can be easily integrated into Streamlit apps and shared online for collaboration and deployment. When using the Streamlit Audio Recorder, the following steps occur:

1. Import the necessary packages: To use the Streamlit Audio Recorder, you need to import the streamlit and soundfile packages.
2. Create a Streamlit Audio Recorder widget: To create the Audio Recorder widget, you can use the st.audio() function with the record=True parameter. This creates an Audio Recorder widget that allows the user to record audio.
3. Record audio: Once the Audio Recorder widget is displayed, the user can click on the "Record" button to start recording audio. The widget will display a timer indicating the length of the recording.
4. Save the recorded audio: Once the user clicks on the "Stop" button to stop recording, the recorded audio is saved to a temporary file. You can then use the soundfile.read() function to read the saved audio data and process it as needed.
5. Display the recorded audio: To display the recorded audio, you can use the st.audio() function again, but this time passing in the recorded audio data as the data parameter.

Overall, the Streamlit Audio Recorder provides an easy way to record and

process audio within a Streamlit app.

4.3.3 Geopy:

Geopy is a Python library that provides geocoding and reverse geocoding capabilities. Geopy is built on top of popular geocoding services such as Google Maps API, Bing Maps API, OpenStreetMap Nominatim, and others, and offers a simple and consistent interface for geocoding and reverse geocoding.

4.3.3 Geopy

Here are some key features of Geopy:

- Geocoding: Geopy can convert addresses or location descriptions into geographic coordinates (latitude and longitude) using various geocoding services. Developers can specify which geocoding service to use, and Geopy will automatically handle the API requests and responses.

- Reverse Geocoding: Geopy can also perform reverse geocoding, which is the process of converting geographic coordinates into a human-readable address or location description. This can be useful for displaying location names on a map, for example.

- Multiple Geocoding Services: Geopy supports multiple geocoding services, including Google Maps API, Bing Maps API, OpenStreetMap Nominatim, and others. This allows developers to choose the geocoding service that best suits their needs.

- Consistent Interface: Geopy provides a consistent interface for geocoding and reverse geocoding, regardless of the underlying

geocoding service. This makes it easy for developers to switch between different services without having to modify their code.

- Extensible: Geopy is extensible, allowing developers to add support for additional geocoding services or customize the behavior of existing services.

- Open Source: Geopy is open source and free to use, making it accessible to developers of all levels.

Overall, Geopy is a powerful and flexible geocoding library for Python that simplifies the process of adding geocoding and reverse geocoding capabilities to Python applications.

eopy is a Python client for several popular geocoding web services.

geopy makes it easy for Python developers to locate the coordinates of addresses, cities, countries, and landmarks across the globe using third-party geocoders and other data sources.

geopy includes geocoder classes for the OpenStreetMap Nominatim, Google Geocoding API (V3), and many other geocoding services. The full list is available on the Geocoders doc section. Geocoder classes are located in geopy.geocoders.

geopy is tested against CPython (versions 3.5, 3.6, 3.7, 3.8, 3.9) and PyPy3. geopy 1.x line also supported CPython 2.7, 3.4 and PyPy2.

Installation
Install using pip with:

pip install geopy
Or, download a wheel or source archive from PyPI.

Geocoding

To geolocate a query to an address and coordinates:

```
>>> from geopy.geocoders import Nominatim
>>> geolocator = Nominatim(user_agent="specify_your_app_name_here")
>>> location = geolocator.geocode("175 5th Avenue NYC")
>>> print(location.address)
Flatiron Building, 175, 5th Avenue, Flatiron, New York, NYC, New York, ...
>>> print((location.latitude, location.longitude))
(40.7410861, -73.9896297241625)
>>> print(location.raw)
{'place_id': '9167009604', 'type': 'attraction', ...}
```

To find the address corresponding to a set of coordinates:

```
>>> from geopy.geocoders import Nominatim
>>> geolocator = Nominatim(user_agent="specify_your_app_name_here")
>>> location = geolocator.reverse("52.509669, 13.376294")
>>> print(location.address)
Potsdamer Platz, Mitte, Berlin, 10117, Deutschland, European Union
>>> print((location.latitude, location.longitude))
(52.5094982, 13.3765983)
>>> print(location.raw)
{'place_id': '654513', 'osm_type': 'node', ...}
```

Measuring Distance

Geopy can calculate geodesic distance between two points using the geodesic distance or the great-circle distance, with a default of the geodesic distance available as the function geopy.distance.distance.

Here's an example usage of the geodesic distance, taking pair of (lat, lon) tuples:

```
>>> from geopy.distance import geodesic
>>> newport_ri = (41.49008, -71.312796)
>>> cleveland_oh = (41.499498, -81.695391)
>>> print(geodesic(newport_ri, cleveland_oh).miles)
```

538.390445368
Using great-circle distance, also taking pair of (lat, lon) tuples:

```
>>> from geopy.distance import great_circle
>>> newport_ri = (41.49008, -71.312796)
>>> cleveland_oh = (41.499498, -81.695391)
>>> print(great_circle(newport_ri, cleveland_oh).miles)
536.997990696
```

4.3.3.1 Geo Coders

Geocoders are tools used to convert addresses or other location descriptions into geographic coordinates (latitude and longitude), which can then be used to display the location on a map or perform geospatial analysis. Here are some important points about geocoders:

- Types of geocoders: There are different types of geocoders, including online geocoders that use APIs to convert addresses into coordinates, and desktop geocoders that can be installed on a computer or server and used offline.

- Geocoding accuracy: The accuracy of geocoding depends on the quality and completeness of the address or location description provided. Incomplete or inaccurate addresses may result in incorrect or imprecise geocoding.

- Geocoding data sources: Geocoders use a variety of data sources, including geospatial data from government agencies, commercial providers, and crowdsourced data.

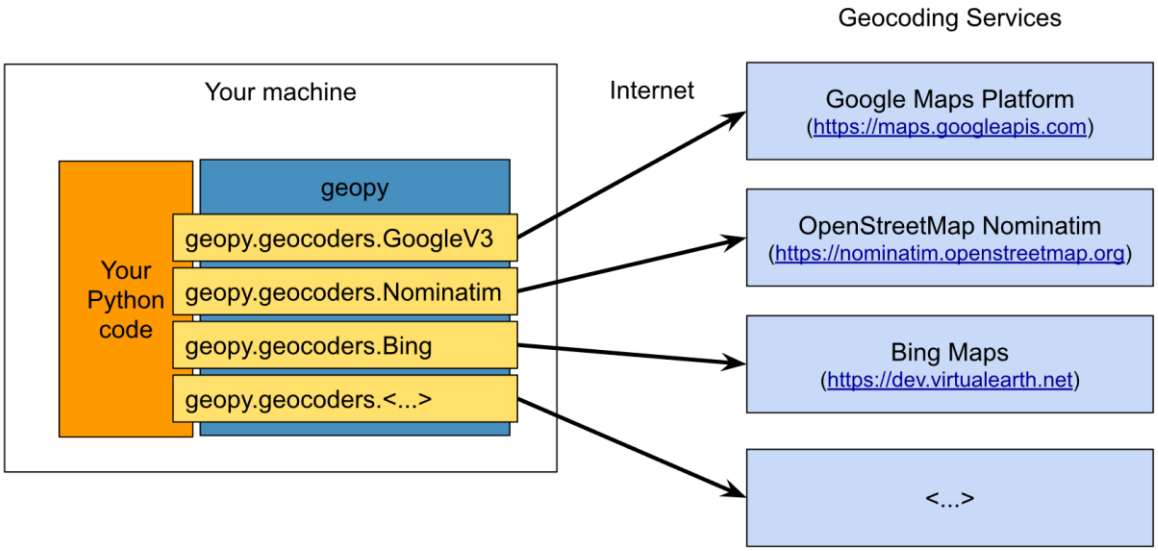
- Reverse geocoding: Reverse geocoding is the process of converting geographic coordinates into an address or location description. Reverse geocoding is useful for displaying a location name or address on a map based on its coordinates.

- Batch geocoding: Batch geocoding is the process of geocoding a large

number of addresses at once, typically using a spreadsheet or database. Batch geocoding can be useful for analyzing or visualizing large datasets.

- Geocoding APIs: Many online geocoders offer APIs that developers can use to integrate geocoding functionality into their applications. Some popular geocoding APIs include Google Maps API, Bing Maps API, and OpenStreetMap Nominatim.

- Geocoding standards: Geocoding standards, such as the Open Geospatial Consortium (OGC) and Geographic Markup Language (GML), help ensure interoperability and consistency across different geocoding tools and applications.



4.3.3.1 GeoCoders illustration

Overall, geocoders play an important role in geospatial analysis, mapping, and location-based services, and their accuracy and reliability are critical for ensuring accurate results.The working of geocoders involves converting addresses or location descriptions into geographic coordinates (latitude and longitude), which can then be used to display the location on a map or perform geospatial analysis. Here is an overview of how geocoders work:

1. Address Parsing: The first step in geocoding is parsing the input address into its individual components such as street number, street name, city, state, and

country. This step is necessary to identify the correct location accurately.

2. Matching the Address to the Database: The next step is to match the parsed address to the geocoding database. Geocoding databases contain spatial data such as street networks, building polygons, and points of interest. The geocoder uses this data to match the input address with the nearest spatial feature in the database.

3. Spatial Analysis: Once the input address has been matched to the nearest feature in the database, the geocoder performs spatial analysis to calculate the geographic coordinates of the location. This involves using mathematical algorithms to estimate the latitude and longitude of the input address based on the spatial data in the database.

4. Geocoding Results: After the geocoding process is complete, the geocoder returns the geographic coordinates of the input address in the form of latitude and longitude. These coordinates can then be used to display the location on a map or perform geospatial analysis.

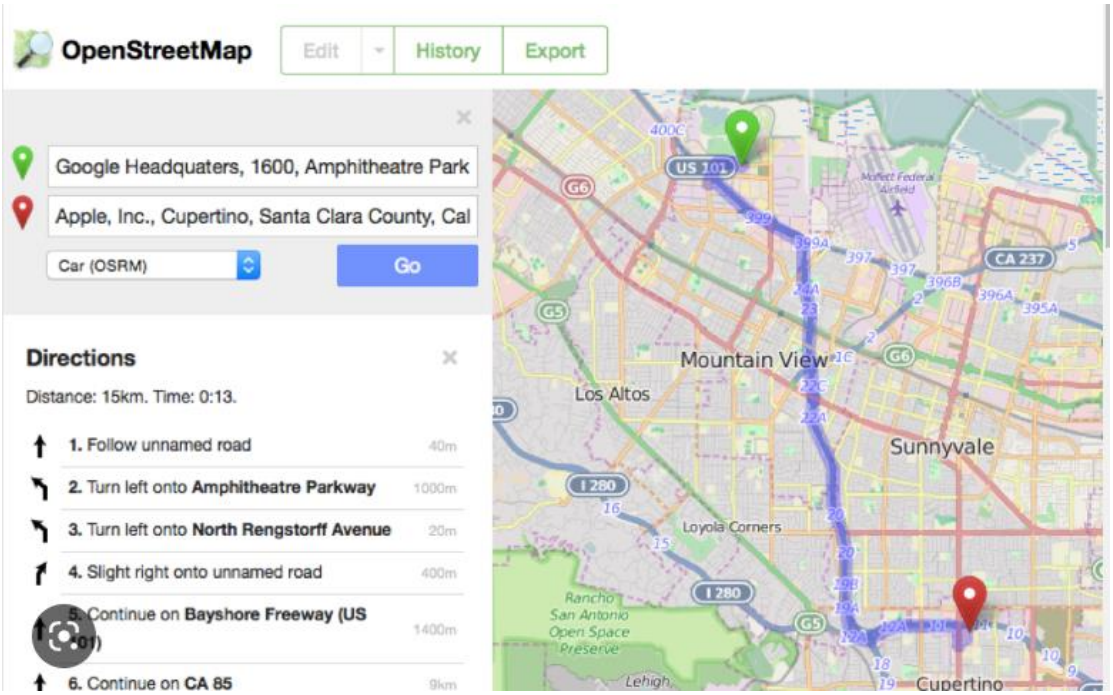
It is important to note that the accuracy of geocoding depends on the quality and completeness of the input address and the geocoding database. Incomplete or inaccurate addresses may result in incorrect or imprecise geocoding results. Similarly, outdated or incomplete geocoding databases may also result in inaccurate geocoding results.

4.3.4OpenStreetMaps:

OpenStreetMap (OSM) is a collaborative mapping project that aims to create a free and open map of the world. As a result, virtually any kind of matter can be added to OpenStreetMap as long as it is legal and fits within the guidelines and rules set by the OpenStreetMap community. Here are some examples of what can be added to OpenStreetMap:

- Geographic features: OSM is primarily used for mapping geographic features like roads, buildings, parks, rivers, lakes, and other natural and man-made features.

- Points of interest: OpenStreetMap can also be used to map points of interest, such as businesses, landmarks, tourist attractions, and public facilities like hospitals, schools, and libraries.
- Transportation networks: OSM is frequently used to map transportation networks, including roads, railways, airports, and public transit systems.
- Historical landmarks: OpenStreetMap can also be used to map historical landmarks and other significant cultural or historical sites.
- Humanitarian mapping: OSM is often used for humanitarian mapping projects, such as mapping disaster-stricken areas, refugee camps, and other areas where mapping is critical to support relief efforts.
- Local knowledge: OSM relies heavily on local knowledge, and community members are encouraged to add any relevant information that they possess about their local area, including local street names, local businesses, and other details that might not be included on official maps.



4.3.3OpenStreetMaps illustration

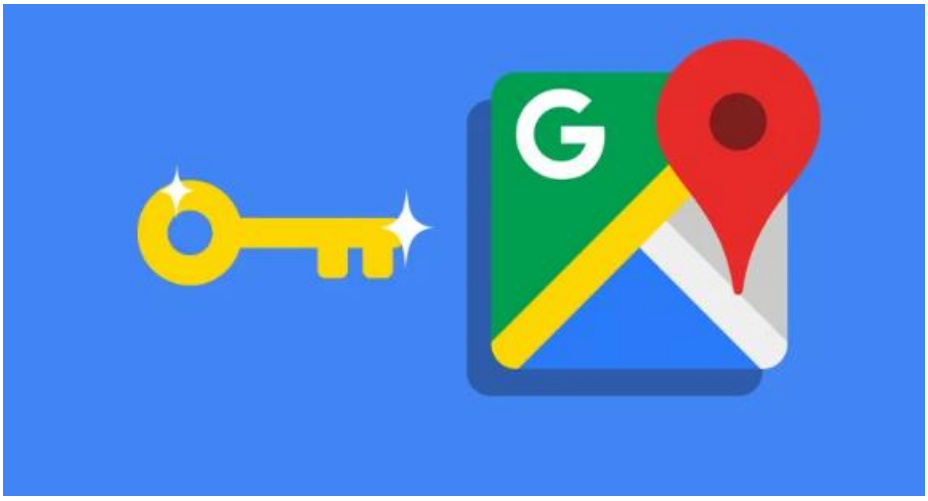
In general, OpenStreetMap is a highly versatile mapping tool that can be used for a wide range of applications. If you're unsure whether something can be added to OSM, it's always best to consult the community guidelines and reach out to other members of the OpenStreetMap community for guidance.

4.3.5 Google Map API:

Google Maps API (Application Programming Interface) is a set of tools and services provided by Google that allow developers to embed Google Maps into their own applications and websites. Here is a brief explanation of how Google Maps API works:

1. Sign up for a Google Maps API key: Before using Google Maps API, developers need to sign up for a Google Maps API key, which allows them to access the Google Maps API services.
2. Embed a map on your website: Once you have an API key, you can embed a Google Map on your website or application by using a simple code snippet. This code specifies the size and location of the map on the page.
3. Customize the map: Developers can customize the map by adding markers, overlays, and other features to the map. They can also adjust the zoom level and map type to match the needs of their application.
4. Use the Directions API: The Directions API allows developers to integrate directions and travel times into their applications. This API provides route information for various modes of transportation, including driving, walking, and cycling.
5. Use the Geocoding API: The Geocoding API allows developers to convert addresses into geographic coordinates, which can be used to display a location on a map or to search for nearby points of interest.
6. Use the Places API: The Places API allows developers to search for places of interest near a given location, such as restaurants, hotels, and tourist attractions.
7. Use the Street View API: The Street View API allows developers to embed 360-degree street-level imagery into their applications, providing a more immersive experience for users.

In summary, Google Maps API provides developers with a range of tools and services to create customized maps and location-based applications. Developers can integrate directions, geocoding, and points of interest search into their applications using Google Maps API.



4.3.5 Google Maps Api Key

Google Maps requires users to obtain an API key to access their maps and services.

To obtain an API key for Google Maps, you need to follow these steps:

- Go to the Google Cloud Console and sign in or create an account.
- Create a new project (if you don't have one already) by clicking on the "Select a project" dropdown menu on the top navigation bar and clicking on "New Project".
- Once you have a project, enable the Google Maps JavaScript API by navigating to the "APIs & Services" tab on the left-hand side menu and selecting "Dashboard". Then click on the "+ ENABLE APIS AND SERVICES" button, search for "Google Maps JavaScript API", and enable it.
- After enabling the API, you need to create credentials to access it. From the "APIs & Services" tab, click on "Credentials" and then "Create credentials". Choose "API key" as the credential type and copy the generated API key.

Finally, use the API key in your Google Maps JavaScript API calls by including it as a parameter in the API request URL.Note that Google Maps has usage limits and charges fees for high usage levels, so make sure to review their pricing and usage policies before using the API extensively.

4.3.6 Twilio

Twilio is a cloud communications platform that allows developers to add messaging, voice, and video to their applications. With Twilio, developers can easily integrate real-time communications into their software, allowing users to communicate via phone, SMS, chat, or video, all without leaving the app.Twilio provides a set of APIs that developers can use to build communication capabilities into their applications. These APIs include:

- Programmable SMS: Allows developers to send and receive SMS messages, including group messaging and delivery status tracking.
- Programmable Voice: Allows developers to make and receive voice calls, as well as manage call control, conferencing, and call recording.
- Programmable Video: Allows developers to embed real-time video chat and screen sharing into their applications.
- Twilio Chat: Allows developers to add in-app chat functionality to their applications.

4.3.6 Twilio Platform

Twilio also provides a number of pre-built communication solutions that developers can use, such as:

- Authy: A two-factor authentication service that provides strong security for user accounts.

- Flex: A fully programmable contact center platform that allows businesses to build and customize their own contact center workflows.
- TaskRouter: A cloud-based routing engine that allows developers to build complex routing workflows for their applications.

Twilio is used by businesses of all sizes, from startups to Fortune 500 companies, across a range of industries including healthcare, finance, e-commerce, and more. It offers scalable, reliable, and secure communication solutions, making it a popular choice for developers looking to add communication functionality to their applications.

To send messages from the Twilio platform using Python, you can use the Twilio REST API and the Twilio Python helper library. Here are the steps you can follow:

Install the Twilio Python helper library using pip:

Copy code

```
pip install twilio
```

Import the twilio.rest module in your Python script:

arduino

Copy code

```
from twilio.rest import Client
```

Create a Client object by providing your Twilio account SID and auth token:

makefile

Copy code

```
account_sid = 'your_account_sid'
auth_token = 'your_auth_token'
client = Client(account_sid, auth_token)
```

Use the client.messages.create() method to send a message:

lua

Copy code

```
message = client.messages.create(
    to='recipient_number',
```

```
    from_='your_twilio_number',
    body='Your message here'
)
```

Replace recipient_number with the phone number you want to send the message to, and your_twilio_number with your Twilio phone number.

Print the sid attribute of the message object to confirm that the message was sent successfully:

scss
Copy code

```
print(message.sid)
```

Here's the complete code:

python
Copy code

```
from twilio.rest import Client

account_sid = 'your_account_sid'
auth_token = 'your_auth_token'
client = Client(account_sid, auth_token)

message = client.messages.create(
    to='recipient_number',
    from_='your_twilio_number',
    body='Your message here'
)

print(message.sid)
```

Note that Twilio charges for each message sent, so make sure you have sufficient credit in your Twilio account.

4.4 PROJECT MANAGEMENT PLAN:

A project management plan for the optimal path finding for an ambulance project would typically include the following key elements:

- Project scope: This section outlines the project's goals, objectives, deliverables, and constraints. For example, the project scope might include the development of a software system that can find the fastest and safest route for an ambulance to an emergency location, taking into account factors such as traffic congestion, road conditions, and weather.

- Project schedule: This section outlines the timeline for the project, including key milestones, deadlines, and dependencies. It should include a detailed Gantt chart or other visual representation of the project schedule.

- Project budget: This section outlines the project budget, including estimated costs for personnel, equipment, materials, and other expenses. It should also include a contingency plan in case of unexpected costs or delays.

- Project team: This section outlines the roles and responsibilities of the project team members, including project manager, software developers, quality assurance testers, and other stakeholders. It should also include a plan for communication and collaboration among team members.

- Risk management: This section outlines potential risks and how they will be managed and mitigated throughout the project. For example, risks might include technical issues, delays in delivery of equipment or materials, or changes in project requirements.

- Quality management: This section outlines how the project team will ensure that the software system meets high quality standards. It should include a plan for testing and validation of the system, as well as a plan for ongoing maintenance and support.

- Project communication: This section outlines how the project team will

communicate progress and updates to stakeholders, including project sponsors, customers, and end-users. It should include a plan for regular project status reports and other forms of communication.

Overall, a project management plan for optimal path finding for an ambulance project should be comprehensive, detailed, and flexible enough to accommodate changes and unexpected challenges. By following a well-designed plan, the project team can ensure that the software system is delivered on time, within budget, and meets the needs of its users.

CHAPTER 5
IMPLEMENTATION DETAILS

5.1 DEVELOPMENT AND DEPLOYEMENT SETUP

The development of finding an optimal path for an ambulance to identify nearby hospitals involves several stages and a specific developmental setup. This setup involves the use of advanced technologies and algorithms that can help identify the most efficient route to a hospital while taking into account factors such as traffic, road conditions, and the location of available medical facilities.

The first stage of this development involves the collection of data related to the road network, hospitals, and other relevant information. This data can be obtained from various sources, such as maps, GPS data, and real-time traffic updates. The data is then fed into an algorithm that can process this information and identify the optimal path to a nearby hospital.

5.1.1 Data Collection

• All data location and information related to the routing of ambulances are collected from the libraries online and these libraries are imported

• Based on the data collected, a road network is developed, where all major routes used to send a patient to designated hospital are constructed in the developed road network.

• Road network developed involved all major roads connected to the designated hospital. The road network consists of nodes and edges, which will be the directional links that connect the two nodes between them.

• The visuals are set for the map

• The information related to the latitudes and longitudes of all the major hospitals in a given area is collected.

• An initial map is created using Folium. The hospital coordinates are loaded on the map using JSON. The data is then sent to the map client.

5.1.2 Algorithm Development:

- The next step is to develop an algorithm that can find the optimal path from the ambulance location to the nearest hospital.

- There are several algorithms that can be used for this purpose, such as Dijkstra's algorithm, A* algorithm, and the Floyd-Warshall algorithm.
- The algorithm should take into account factors such as road conditions, traffic congestion, and the location of hospitals to determine the most efficient route.

5.1.3 Creating a model:

• The next step is creating hexagons for the implementation of algorithm.

• Once the incident site is identified, we developed a detailed road network that shows complete distances

• Here, the algorithm will be applied to determine the shortest path of ambulance distance from a given location to other locations.

• Referring to the model, the idea of this algorithm is to avoid expanding paths that are already far or expensive.

• This algorithm chooses the next node n whose g(n) + h(n) is minimal. This process repeats until the goal node is reached.

• The algorithm chooses its node based on the cost from the start node plus an estimate of the goal node.

• The shortest paths are based on the incident site and current ambulance station

• The algorithm is applied to the whole data and the results in the form of a shortest optimal path are stored in HTML files.

5.1.4 Deployment Setup:

- The final step is to set up the deployment environment for the algorithm.
- This involves creating a server or a cloud-based platform that can handle user requests and run the algorithm in real-time.
- The deployment setup should also include a user interface that allows ambulance drivers to input their location and receive the optimal path.
- The message is being sent by twilio platform.

5.2 ALGORITHMS

Transfer learning is a machine learning technique where a pre-trained model is used as a starting point for a new task or problem. Instead of training a new model from scratch, the pre-trained model's weights and learned features are used as a basis for the new model. This approach can significantly reduce the amount of training data and computational resources required to achieve good performance on the new task.

The general steps involved in transfer learning are as follows:

- Select a pre-trained model: The first step is to select a pre-trained model that has been trained on a large dataset for a similar task as the new task. For example, a pre-trained model trained on the ImageNet dataset for image classification can be used as a starting point for a new image classification task.

- Fine-tuning: The pre-trained model is then modified by replacing the final layer(s) with new layers that are specific to the new task. These new layers are then trained on the new dataset while the pre-trained layers are kept fixed or are slightly adjusted. This process is called fine-tuning.

- Training: The new model is then trained on the new task data. The pre-trained weights and learned features are used as a starting point to speed up the training process and improve the performance of the new model.

- Evaluation: The new model is evaluated on a separate test dataset to determine its performance.

Transfer learning can be used in a variety of applications, including computer vision, natural language processing, and speech recognition. It can also be used to transfer knowledge from one domain to another, such as transferring knowledge learned from text data to image data.

Overall, transfer learning is a powerful technique that allows for faster and more efficient training of machine learning models by leveraging pre-existing knowledge

from pre-trained models.

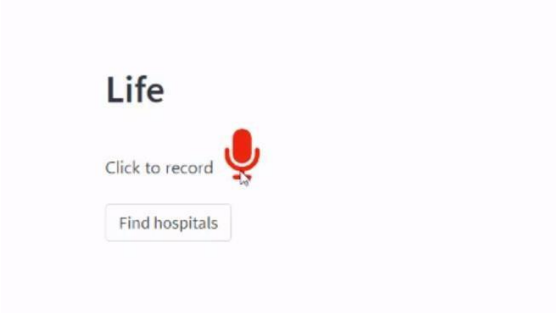
5.3 TESTING

To test a deep learning image processing algorithm, there are several steps you can follow:

- Collect and prepare your dataset: Collect a dataset that is relevant to your problem and prepare it in a format that can be used by your algorithm. This might involve cleaning the data, resizing images, and converting to a suitable file format.
- Split the dataset into training and testing sets: Divide your dataset into two separate sets: a training set, which will be used to train your model, and a testing set, which will be used to evaluate the performance of your model.
- Define the architecture of your model: Choose the appropriate deep learning architecture for your problem. This may involve selecting an existing pre-trained model or designing your own architecture.
- Train your model: Train your model using the training set. This involves feeding the training set through the model and adjusting the model's parameters to minimize the error between the predicted and actual outputs.
- Evaluate the performance of your model: Use the testing set to evaluate the performance of your model. Calculate metrics such as accuracy, precision, recall, and F1 score.
- Iterate and refine your model: Based on the performance of your model, make adjustments to your algorithm and repeat the process until you achieve satisfactory results.
- Test your model on new data: Once you are satisfied with the performance of your model, test it on new data to ensure that it generalizes well to new images.

It's important to note that testing a deep learning image processing algorithm can be a complex and time-consuming process. It may require a significant number of computational resources and expertise in deep learning.

CHAPTER 6
RESULTS AND DISCUSSIONS



6.1Web Interface

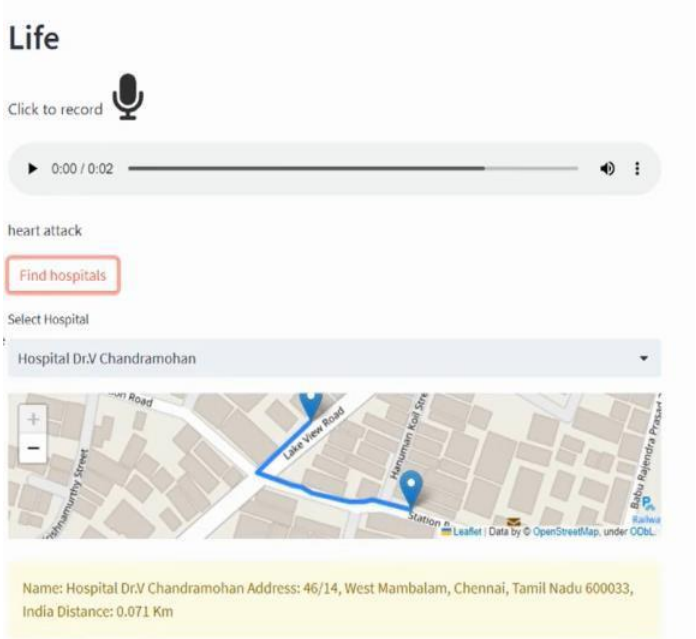
The web interface here displayed is the front page of the entire project .As you can see it is named as Life which is symbolized for the hospital emergency purposes and is significant to the life a patient who is in critical condition.The web page displayed here is being created with the help of Streamlit.

The web interface displays options like click to record for the activation of the mic.The

Mic is used for recording the voices incase if a person doesn't know how to type .

6.2 Record our audio using Streamlit

6.3 Recognizing the speech it will show the nearest hospitals



6.4 Showing the Curated hospitals

By using Geocoders and Open Street Map, The location of the hospital will be shown

6.4 Address as Message

Text Message for the phone regarding the location of hospital by Twilio platform after processing

CHAPTER 7
CONCLUSION

7.1 Conclusion

With this we are coming to an end our project, our ultimate aim is to reduce the deaths caused by the road accidents with our project. In summary, determining the best route for an ambulance is an essential part of emergency response systems. We can assist emergency vehicles in getting to their location as fast and securely as possible by utilizing the power of real-time traffic data and pathfinding algorithms. In the long run, this may result in quicker response times and better results for individuals in need of emergency medical treatment. A crucial step in attaining this aim is the creation of sophisticated ambulance routing tools and algorithms, such polygonal path finding and folium mapping.

7.2 Future Work:

Our future goal is to build a software with more features which will be accessible for all the people. here are several future developmental plans that could be implemented to further enhance the project of finding the optimal path for an ambulance. Some of these plans are:

- Real-Time Traffic Information: Integration of real-time traffic information to the algorithm can help identify congested areas and suggest alternate routes. By using real-time data, the algorithm can dynamically adjust the route to ensure the fastest possible travel time.

- Multi-Objective Optimization: The algorithm can be expanded to optimize multiple objectives such as minimizing travel time and minimizing distance traveled. This can provide greater flexibility to ambulance drivers to choose their preferred route based on their individual needs.

- Integration with Emergency Services: Integration of the algorithm with emergency services can help prioritize ambulance routes based on the severity of the emergency. This can help ensure that the most critical cases receive priority and the fastest possible response.

- Machine Learning: Incorporation of machine learning algorithms can help the system learn from previous trips and adapt to changing traffic patterns. This can further improve the accuracy of the algorithm and reduce response times.

- Mobile Application: Development of a mobile application for ambulance drivers to input their location and receive the optimal path to the nearest hospital. The application can also provide real-time updates on traffic conditions, road closures, and other relevant information.

Overall, these future developmental plans can significantly improve the accuracy and efficiency of the system, ensuring faster response times and better patient outcomes.

7.3 Research Issues

At the initial stage of our research, we found that there were a very low number of references available on our topic. This was concerning to us, as we wanted to ensure that our research was thorough and that we had a comprehensive understanding of the topic. To address this issue, we had to put in extra effort to find additional references that we could use to support our work. We started by conducting a more extensive literature search using a variety of databases and search engines.

We also reached out to experts in the field and asked for their input and suggestions for additional sources. This allowed us to find more relevant research papers, reports, and other materials that we could use to support our work. In addition, we had to carefully evaluate each reference to ensure that it was high-quality and relevant to our research question. This involved reading each paper or report in detail and assessing its strengths and weaknesses. We also had to ensure that we cited each reference correctly and included all relevant information in our bibliography.

Overall, while the initial lack of references was a concern, we were able to overcome this issue by putting in extra effort to find additional sources. This allowed us to ensure that our research was comprehensive and well-supported by relevant literature.

7.4 Implementation Issues

Finding an optimized path for an ambulance project using Streamlit involves a few implementation issues that need to be considered. Here are some key points to keep in mind like the

- Data Sources :The data sources used for the project. You will need to decide on the data sources that will be used for the project. This could include traffic data, road network data, ambulance station locations, and hospital locations. You will need to ensure that the data is up to date and accurate.

- Another thing is Optimization Algorithm. You will need to choose an appropriate algorithm to optimize the ambulance path. This could include algorithms such as Dijkstra's algorithm or A* algorithm. You will also need to ensure that the algorithm is suitable for real-time applications.

- User Interface design Streamlit provides a simple and user-friendly interface for data visualization and interaction. You will need to design an intuitive user interface that allows the user to input the location of the incident and the destination hospital. The user interface should also display the optimized ambulance path and estimated time of arrival.

- Integration with Mapping APIs is another difficult task we need to integrate the application with mapping APIs such as Google Maps or OpenStreetMap to display the road network and the ambulance path. This will require an API key and familiarity with the API documentation.

- Performance of the application should be designed to perform well even with a large amount of data. You may need to optimize the algorithm and use caching to improve performance.

- Testing and Debugging of the project is important to thoroughly test the application and debug any issues before deploying it for real-world use. This may involve running test scenarios and collecting user feedback.

By considering these implementation issues, you can create an effective and efficient ambulance path optimization application using Streamlit.

REFERENCES

[1] Taha Darwassh Hanawy Hussein, Mondher Frikha, Sulayman Ahmed, Javad Rahebi, *"**Ambulance Vehicle Routing in Smart Cities Using Artificial Neural Network**"*, 6th International Conference on Advanced Technologies for Signal and Image Processing (ATSIP), 2022

[2] Taha Darwassh Hanawy Hussein; Mondher Frikha; Sulayman Ahmed; Javad Rahebi, ***"Ambulance Vehicle Routing using BAT Algorithm",*** International Conference on Electrical, Communication, and Computer Engineering (ICECCE), 2021

[3] Elgarej Mouhcine, Yassine Karouani, Khalifa Mansouri, Youssfi Mohamed, ***"Toward a distributed strategy for emergency ambulance routing problem",*** 4th International Conference on Optimization and Applications (ICOA), 2018

[4] Tressa Michael, Deepthy Xavier, ***"Intelligent Ambulance Management System with A Algorithm***", 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), 2020

[5] Mohamed N. Ashmawy, Ahmad M. Khairy, Mohamed W. Hamdy, Anas ElShazly, Karim El-Rashidy, Mohamed Salah, Ziad Mansour, Ahmed Khattab, "***SmartAmb: An Integrated Platform for Ambulance Routing and Patient Monitoring",*** 31st International Conference on Microelectronics (ICM), 2020

[6] [6] AbdelGhani Karkar, ***"Smart Ambulance System for Highlighting Emergency-Routes"***, Third World Conference on Smart Trends in Systems Security and Sustainability (WorldS4), 2019

[7] Subash Humagain, Roopak Sinha, "***Routing Autonomous Emergency Vehicles in Smart Cities Using Real Time Systems Analogy: A Conceptual Model"***, IEEE 17th International Conference on Industrial Informatics (INDIN), 2020

[8] Mohammad A. R. Abdeen, Mohamed Hossam Ahmed, Hafez Seliem[, Tarek Rahil Sheltami, Turki M. Alghamdi, Mustafa El-Nainay, "***A Novel Smart Ambulance System—Algorithm Design, Modeling, and Performance Analysis***", IEEE Access (Volume: 10), 2022

[9] Myint Sein, K-zin Phyo, Mau Luen Tham, Yasunori Owada, Nordin Bin Ramli, Suvit Poomrittigul, ***"Effective Evacuation Route Strategy for Emergency Vehicles"***, IEEE 10th Global Conference on Consumer Electronics (GCCE), 2021 [10] Nikki Rathore, Pramod Kumar Jain, Manoranjan Parida, "A Routing Model for Emergency Vehicles Using the Real Time Traffic Data", IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI), 2018.

[10] Taha Darwassh Hanawy Hussein, Mondher Frikha, Sulayman Ahmed, Javad Rahebi, ***"Ambulance Vehicle Routing in Smart Cities Using Artificial Neural Network",*** 6th International Conference on Advanced Technologies for Signal and Image Processing (ATSIP), 2022

[11] Mondher Frikha; Sulayman Ahmed; Javad Rahebi***, "Ambulance Vehicle Routing using BAT Algorithm",*** International Conference on Electrical, Communication, and Computer Engineering (ICECCE), 2022

[12] Elgarej Mouhcine, Yassine Karouani, Khalifa Mansouri, Youssfi Mohamed, "***Toward a distributed strategy for emergency ambulance routing problem"***, 4th International Conference on Optimization and Applications (ICOA), 2018

[13] Tressa Michael, Deepthy Xavier, ***"Intelligent Ambulance Management System with A Algorithm",*** 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), 2020

[14] Mohamed N. Ashmawy, Ahmad M. Khairy, Mohamed W. Hamdy, Anas El-Shazly, Karim El-Rashidy, Mohamed Salah, Ziad Mansour, Ahmed Khattab, ***"SmartAmb: An Integrated Platform for Ambulance Routing and Patient Monitoring",*** 31st International Conference on Microelectronics (ICM), 2020

[15] AbdelGhani Karkar, ***"Smart Ambulance System for Highlighting Emergency-Routes",*** Third World Conference on Smart Trends in Systems Security and Sustainability (WorldS4), 2019

[16] Subash Humagain, Roopak Sinha***, "Routing Autonomous Emergency Vehicles in Smart Cities Using Real Time Systems Analogy:*** A Conceptual Model", IEEE 17th International Conference on Industrial Informatics (INDIN), 2020.

[17] Mohammad A. R. Abdeen, Mohamed Hossam Ahmed, Hafez Seliem[, Tarek Rahil Sheltami, Turki M. Alghamdi, Mustafa El-Nainay, "***A Novel Smart Ambulance System—Algorithm Design, Modeling, and Performance Analysis",*** IEEE Access (Volume: 10), 2022

[18] Myint Myint Sein, K-zin Phyo, Mau Luen Tham, Yasunori Owada, Nordin Bin Ramli, Suvit Poomrittigul, ***"Effective Evacuation Route Strategy for Emergency Vehicles",*** IEEE 10th Global Conference on Consumer Electronics (GCCE), 2021

[19] Nikki Rathore, Pramod Kumar Jain, Manoranjan Parida, ***"A Routing Model for Emergency Vehicles Using the Real Time Traffic Data"***, IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI), 2018

APPENDIX

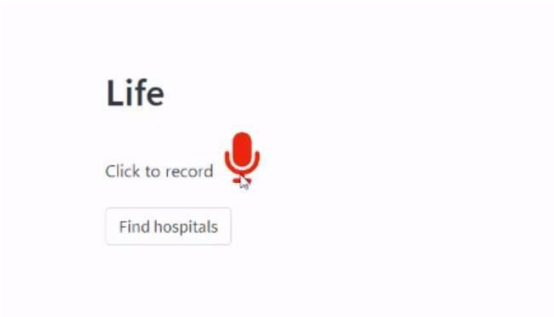
A. SOURCE CODE

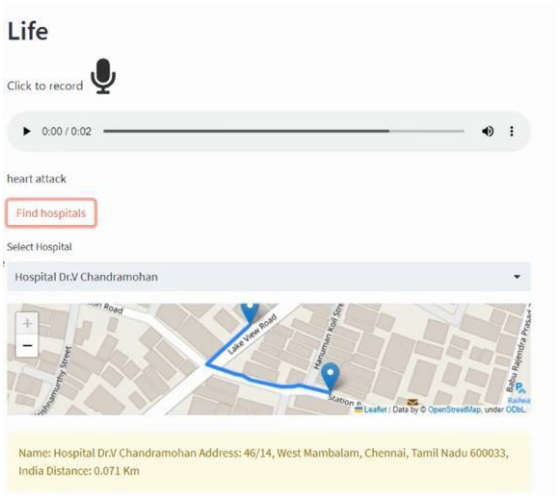
```
import streamlit as st

from utils import *

import streamlit.components.v1 as components

from audio_recorder_streamlit import audio_recorder

import speech_recognition as sr

import pyttsx3

from mail import send_sms

from geopy.geocoders import Nominatim

locator = Nominatim(user_agent="myGeocoder")

r = sr.Recognizer()

def STT():

    try:

        with sr.AudioFile("audio.wav") as source2:

            audio2 = r.listen(source2)

            MyText = r.recognize_google(audio2)

            MyText = MyText.lower()


            return MyText


    except sr.RequestError as e:

        print("Could not request results; {0}".format(e))


    except sr.UnknownValueError:

        print("unknown error occurred")
```

```
def main():
    st.header('Life')
    #condition = st.text_input('Enter Your Status')
    audio_bytes = audio_recorder()
    if audio_bytes:
        st.audio(audio_bytes, format="audio/wav")
        wav_file = open("audio.wav", "wb")
        wav_file.write(audio_bytes)
        type_res = STT()
        st.write(type_res)



    ch = st.checkbox('Find hospitals')
    if ch:
        my_lat, my_long = get_my_loc()
        location = locator.reverse((my_lat, my_long))
        address = location.address
        nearest_hospitals = get_nearest(my_lat, my_long)


        names, address, location, distance = get_names(nearest_hospitals)


        choice = st.selectbox('Select Hospital', options=names, index=0)
        if choice:
            indx = names.index(choice)
            route = get_route(my_lat, my_long, *location[indx])


            m = create_map(route)
```

```
                m.save('map.html')


                HtmlFile = open("map.html", 'r', encoding='utf-8')
                source_code = HtmlFile.read()
                components.html(source_code)
                st.warning(
                    '''
                    Name: {}
                    Address: {}
                    Distance: {} Km
                    '''.format(names[indx], address[indx], distance[indx]/1000)
                )
                try:
                    msg = '''
                    Location: {}
                    Type: {}
                    Address: {}
                    '''.format((my_lat,my_long), type_res, address[0])
                    send_sms(msg)
                except:
                    print('SMS NOT SENT')



main()
```

B. SCREENSHOTS






2:05
Thu, Jan 19 PM
Messages · Now
503501
Location: (13.0389, 80.2258)
Type: heart attack
Address: 46/14, West Mambalam, Chennai, Tamil Nadu 600033, India
Mark as read
Delete

C.RESEARCH PAPER

Finding an Optimal Route Path for Ambulance

CHAITANYA BALAJI I
Student
Department of CSE,
Sathyabama Institute of Science and Technology.
Chaitanyabalaji10@gmail.com

POORNA SIVA SAI.V
Student
Department of CSE,
Sathyabama Institute of Science and Technology.
Poorna vejandla27@gmail.com

Sujihelen. L
Assistant Professor
Department of CSE,
Sathyabama Institute of Science and Technology.
sujihelen.cse@sathyabama.ac.in

Abstract-

The first worry in emergency situations is how to get to the scene of the occurrence in the shortest amount of time. In such circumstances, we are most in need of a guide to assist us in locating the ambulance vehicles to the nearest hospitals. In this project we are introducing a type of web application which helps to reduce deaths caused by road accidents . In this application we have introduced a voice recognizer which can acknowledge the speech and share the details of the nearest hospitals which suits best according to the emergency through a message containing the exact longitude and latitude of the hospital. We have used python as our programming language within this we have used speech recognition for recognizing our client request and Geocoders for getting the latitudes and longitudes of the nearest hospital

Keywords-Shortest Distance, Speech Recognition, Geocoders, StreamLit.

I INTRODUCTION

Medical advancements have allowed for the treatment of a wide variety of illnesses and injuries in today's globe. In the event of an accident, however, it is essential that the victim receive help immediately. The quick arrival of an ambulance at the scene of such an event has enormous social and economic implications. A number of victims have been reported dead after ambulances were rescued too late.

The main causes of this are things like the ambulance not leaving at the scheduled time, the driver not being familiar with the routes, and traffic congestion. A patient in cardiac arrest, for instance, has a 24% lower probability of life for every minute that passes before they receive care. As a result of the critical and immediate influence it has on people's lives, finding an efficient route for ambulances is one of the most challenging study subjects.

As cities expand, the function of emergency rescue systems in ensuring the safety of human life and societal stability grows in significance. The purpose of the ambulance service provided by EMS is to either provide immediate medical attention to patients at the scene of an emergency or transport them to a hospital for more extensive medical care. When an emergency arises, the provided EMS ambulance will be there to help with basic life support measures.

Emergency medical services (EMS) provide lifesaving emergency care and ambulance transportation to patients in need in accordance with established protocols and recommendations. In addition to providing emergency medical services, EMS also offers standby services for events across national and international conferences, responsibilities during emergency landings

or accidents of airplanes, as well as the transport of patients from one hospital to another for a small fee.

Availability of ambulances and the time it takes to get there must be guaranteed to meet the goal time for treatment. Therefore, effective ambulance management and system are crucial for raising EMS quality.

II LITERATURE REVIEW

[1] Taha Darwassh Hanawy Hussein suggested an artificial intelligence-powered neural network to aid in allocating emergency medical care. There are eight values given to the neural network. The data contain the date and time of the accident, its place, the geography of the ambulance and hospital, streets involved, the number of wounded people, the nature of the accident, and the ages of those involved. They also include the number of injured people. The ambulance may choose the most practical path to the closest medical centre using this information. This study evaluates the city's resources and response to an accident, the efficiency of current emergency teams, and the availability of ambulances..

[2] Hussein recommended utilising the BAT algorithm to determine the ambulances' most direct route. Node-based construction is used to create the city map. The control centre will initially alert the hospital and ambulance to the accident's location. The driver then enters the data into the BAT algorithm, which uses the node coordinates of the accident and the ambulance vehicle to calculate the quickest path to the accident spot..

[3] In order to determine the optimum route for an ambulance to go in the event of an emergency, Elgarej Mouhcine suggested a distributed technique. Urban rescue operations have traditionally been primarily focused on determining the best course of action. A decentralised approach is suggested that draws inspiration from the algorithmic organisation of ant colonies. This study provides a distributed model that considers aspects including speed limits, traffic, the availability of ambulances, and the proximity of the hospital to choose the best route that significantly reduces time..

[4] Time is wasted sitting in traffic a lot in the fast-paced world of today. This is a serious problem for ambulances and other emergency vehicles. The time that is wasted may have been used to save lives. In order to deal with this dangerous situation, Tressa Michael has suggested using the A* algorithm to choose the ambulance's best course of action. This programme considers the flow of traffic and creates a dynamic route based on the volume of traffic. The ambulance receives the coordinates of the accident, and the fastest path is shown on the dashboard. This makes it easier to get there faster..

[5] Mohamed N. Ashmawy suggested a central hub as a remedy that could manage ambulance dispatching and tracking concurrently. The platform aims to enhance the patient's survival prospects by getting them to the hospital as soon as possible, and it also enables a physician to analyse the patient's biological data while they are still travelling. The latter is advantageous since it gives the hospital time to prepare before the patient shows there. To assist the doctor in identifying any potential health hazards, the obtained data is subsequently analysed using machine learning techniques..

[6] Paramedics must act quickly and rationally in life-or-death situations, especially in crowded locations. Unfortunately, the high traffic and congestion significantly reduce the ambulance's pace. Drivers frequently fail to pull aside for an ambulance when it is farther away because they only do it when

the ambulance is directly in their path. AbdelGhani Karkar suggests a more sophisticated ambulance system in this study that distinguishes itself from existing ones by informing drivers in advance of the ambulance's emergency routes. The system offers both a medical application and a user emergency application. The first shows both the location of the patient and the ambulance (s). Contrarily, the latter not only locates the.

[7] vehicles such as fire engines, Ambulance, cop vehicles and other vehicles rely on their drivers' prompt replies. Traffic jams, a high number of intersections with traffic lights, and protracted vehicle wait times provide considerable challenges for emergency vehicles. Due to the inability of earlier attempts to route EVs to handle continuously shifting traffic circumstances, they were unsuccessful. Real-time information on obstructions is crucial to reducing the wait time for EV mobility and its impact on other traffic. AEVs should be used, according to Subash Humagain, since they can make decisions in real time based on current data, which enables them to negotiate intricate road networks more swiftly and safely. In order to achieve critical reaction time inside a real-time system, AEVs' tasks are specifically planned using the mixed-criticality realtime system (MCRTS)..

[8[Mohammad A. R. Abdeen introduced a revolutionary technology that will use data gathering, processing, and sensing capabilities to revolutionise 911 and ambulance service. In order to choose the best course of action, the intelligent system considers various factors such as the flow of traffic on roads and count of patients at the hospital . To confirm the algorithm's validity, analytical and simulated studies of its performance are employed. The outcomes, which demonstrated exceptional agreement between the simulation and analytical methodologies, supported the correctness of the analysis. The smart algorithm performed better than other algorithms that had been previously published in the literature under the settings and situations that were taken into consideration.

[9] If lives and property are to be saved, rescue crews must be at the scene of a disaster quickly. The Dijkstra algorithm is used to create an effective device for predicting safe routes over a complicated, unstructured road network. With the help of the anticipated technologies, the evacuation route for police vehicles, ambulances, and fire trucks will be determined. We evaluate the best evacuation route, look for neighbouring relief zones, and guide you to a safe location. The location of the closest emergency service may be found using the neighbouring services system. The haversine distance can be used to calculate the separation between two locations. Once an incident has occurred, the best route from the service centre to the incident area is made.

[10] 108 is one of a number of alternate emergency medical services available in India's several states (EMS). Real-time travel and traffic data will be included into the vehicle scheduling and routing model in order to quickly meet demand., efficient EMS depends by a number of interrelated elements. In order to find the optimal routes for emergency vehicles, this study uses the Maps Matrix API to construct an optimization approach based on real-time live traffic data. The truck routing problem is modelled as an integer programming problem in the heuristic technique, and it is then optimised using the Google API. The model's key features include the capacity to optimise dispatching, identify the fastest routes, locate the scene of accidents, and track the whereabouts of vehicles..

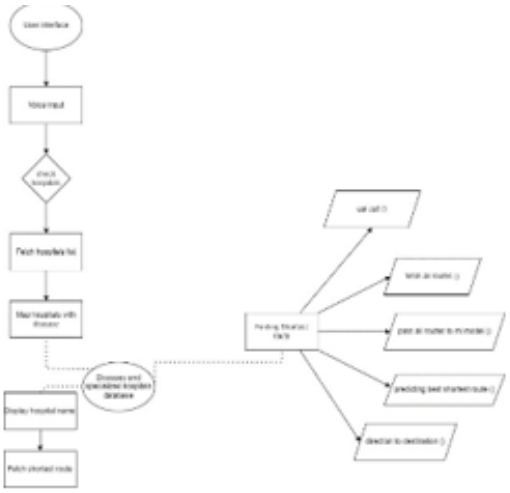
Existing system

When utilizing a standard disease risk model to make a prediction, the process typically incorporates machine learning and supervised learning algorithms. These algorithms employ training data that is labeled for the purpose of training the models. Patient classification into high-risk and low-risk categories is carried out using group test sets. There is already a great deal of applications that have been developed for use in the area of hospitals, physicians, and medical care. Among the tools that are at your disposal are medical recommendations, an activity log, and drugs, information regarding the procedures, directions to the hospital, and patient reviews of the facility.

Proposed system

In the process of completion of this project we have used a system which takes the speech from the user through **Streamlit** which is used to record the audio, And this audio can be processed using the "**recognize google"** python library. By using the **Geocoders** and **OpenStreetMap** we will get the details of the nearest hospitals and their exact location and within these hospitals the nearest hospital can be addressed using the "**get_nearest"** function i.e. used in the programming of the code. After getting the location, for sending the hospital details to the user we have used the **Twilio** platform.Through this platform we can send the details of the hospital to the user in the form of text message.

III ARCHITECTURE DIAGRAM



IV DESCRIPTION OF THE PROPOSED MODEL/SYSTEM:

Module 1: Streamlit

Developers may easily construct interactive web apps thanks to the open-source Python framework known as Streamlit. It may be used for a variety of activities, including data exploration and visualization, machine learning, and natural language processing. It is intended to make the process of developing data-driven web apps more straightforward.

"This streamlit component allows to register an audio utterence from a user."

The ability of Streamlit to dynamically update and refresh the user interface as data or settings are modified is one of its main advantages. Without having to manually reload the website, this enables developers to rapidly iterate and experiment with their apps.

Overall, Streamlit is a powerful tool for building data-driven web applications, particularly for data scientists and developers who want to quickly prototype and test new ideas.

Module 2: Recognize Google

The Speech Recognition API from Google is excellent. Simply said, this API

transforms spoken words from a microphone into written text written as Python strings. Google API will automatically convert any spoken words you make into printed form. For the English language, the API yields great results.

By offloading the logic, a voice recognition API enables you to submit a web request to the API and receive the text that was recognized in response. Although you may perform this straight from Python code, your script will also need to have access to the internet.
The speech recognition module, which will be added on top of the application, is used to record the audio. The Google voice recognition API is then used to process the recorded speech and deliver the results.

r.recognize_google(audio) returns a string.

Module 3: OpenStreetMap

A collaborative initiative called OpenStreetMap (OSM) aims to provide a free, editable map of the whole planet. Since its founding in 2004, it has developed into one of the most complete and frequently utilized sources of geographic data.

The OSM project is dependent on volunteers to add and change data on the roads, structures, landmarks, and other objects in their local area. Then, under the terms of an open license, this data is made available to everyone and preserved in a public database.

OSM offers tools and APIs that let programmers access and utilize this data in a number of ways. For instance, programmers may interact with OSM data programmatically by using the OSM API to access map data and build bespoke map apps, or they can utilize pre-built libraries like osmapi, osmnx, or overpass-api.

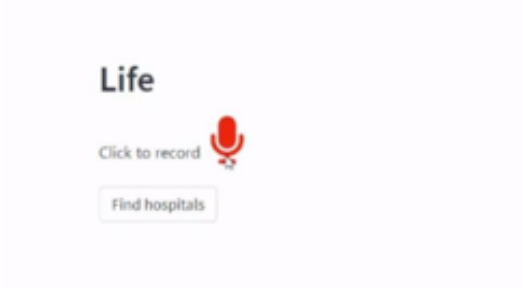
Module 4: Twilio platform

Twilio is a cloud communication platform that enables developers to create and include communication services into their apps, such as sending and receiving emails, SMS messages, and phone calls.

A variety of APIs and SDKs are available on the Twilio platform, which makes it simple for programmers to include communication features into their apps. For instance, the Programmable Voice API from Twilio may be used to place and receive phone calls, while the Programmable SMS API from Twilio can be used to send and receive SMS messages.
Additionally, Twilio offers a selection of pre-built tools and interfaces that make it simple to include communication capabilities into well-liked web development frameworks and platforms, like Node.js, Ruby on Rails, and WordPress.

V RESULTS:

This is the interface of our project created by using Streamlit.



We will record our audio using Streamlit as shown below:

After recognizing the speech it will show the nearest hospitals

By using Geocoders and Open Street Map, The location of the hospital will be shown as follows:

Text Message for the phone regarding the location of hospital by Twilio platform:

CONCLUSION:

With this we are coming to an end our project, our ultimate aim is to reduce the deaths caused by the road accidents with our project. In summary, determining the best route for an ambulance is an essential part of emergency response systems.

We can assist emergency vehicles in getting to their location as fast and securely as possible by utilizing the power of real-time traffic data and pathfinding algorithms. In the long run, this may result in quicker response times and better results for individuals in need of emergency medical treatment. A crucial step in attaining this aim is the creation of sophisticated ambulance routing tools and algorithms, such polygonal path finding and folium mapping..

Our future goal is to build a software with more features which will be accessible for all the people.

REFERENCES:

[1] Taha Darwassh Hanawy Hussein, Mondher Frikha, Sulayman Ahmed, Javad Rahebi, "Ambulance Vehicle Routing in Smart Cities Using Artificial Neural Network", 6th International Conference on

Advanced Technologies for Signal and Image Processing (ATSIP), 2022

[2] Taha Darwassh Hanawy Hussein; Mondher Frikha; Sulayman Ahmed; Javad Rahebi, "Ambulance Vehicle Routing using BAT Algorithm", International Conference on Electrical, Communication, and Computer Engineering (ICECCE), 2021

[3] Elgarej Mouhcine, Yassine Karouani, Khalifa Mansouri, Youssfi Mohamed, "Toward a distributed strategy for emergency ambulance routing problem", 4th International Conference on Optimization and Applications (ICOA), 2018

[4] Tressa Michael, Deepthy Xavier, "Intelligent Ambulance Management System with A Algorithm", 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), 2020

[5] Mohamed N. Ashmawy, Ahmad M. Khairy, Mohamed W. Hamdy, Anas ElShazly, Karim El-Rashidy, Mohamed Salah, Ziad Mansour, Ahmed Khattab, "SmartAmb: An Integrated Platform for Ambulance Routing and Patient Monitoring", 31st International Conference on Microelectronics (ICM), 2020 [6] AbdelGhani Karkar, "Smart Ambulance System for Highlighting Emergency-Routes", Third World Conference on Smart Trends in Systems Security and Sustainability (WorldS4), 2019

[7] Subash Humagain, Roopak Sinha, "Routing Autonomous Emergency Vehicles in Smart Cities Using Real Time Systems Analogy: A Conceptual Model", IEEE 17th International Conference on Industrial Informatics (INDIN), 2020

[8] Mohammad A. R. Abdeen, Mohamed Hossam Ahmed, Hafez Seliem[, Tarek Rahil Sheltami, Turki M. Alghamdi, Mustafa El-Nainay, "A Novel Smart Ambulance System—Algorithm Design, Modeling, and Performance Analysis", IEEE Access (Volume: 10), 2022

[9] Myint Sein, K-zin Phyo, Mau Luen Tham, Yasunori Owada, Nordin Bin Ramli, Suvit Poomrittigul, "Effective Evacuation Route Strategy for Emergency Vehicles", IEEE 10th Global Conference on Consumer Electronics (GCCE), 2021 [10] Nikki Rathore, Pramod Kumar Jain, Manoranjan Parida, "A Routing Model for Emergency Vehicles Using the Real Time Traffic Data", IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI), 2018