VEHICLE BLACK BOX MODEL INTEGRATED WITH IoT AND DEEP LEARNING TO SAVE LIVES

A PROJECT REPORT

Submitted by

CIBI C	**(20BEC4027)**
DHARANIKUMAR K	**(20BEC4032)**
DHARNESH M	**(20BIT4017)**
GOKUL S	**(20BIT4027)**

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

ELECTRONICS AND COMMUNICATION ENGINEERING

M.KUMARASAMY COLLEGE OF ENGINEERING, KARUR

ANNA UNIVERSITY: CHENNAI 600 025

APRIL 2024

M.KUMARASAMY COLLEGE OF ENGINEERING, KARUR

BONAFIDE CERTIFICATE

Certified that this project report **"VEHICLE BLACK BOX MODEL INTEGRATED WITH IOT AND DEEP LEARNING TO SAVE LIVES" is the bonafide work of "DHARNESH M (20BIT4017), GOKUL S (20BIT4027), CIBI C (20BEC4027), DHARANIKUMAR K (20BEC4032)" who carried out the project work under my supervision in the academic year 2023-2024**.

SIGNATURE	**SIGNATURE**
Dr.A.KAVITHA, M.E.,Ph.D.,	**Mrs.P.SANMUGAVALLI, M.E.,**
HEAD OF THE DEPARTMENT	**SUPERVISOR**
Professor,	Assistant Professor,
Department of Electronics and	Department of Electronics and
Communication Engineering,	Communication Engineering,
M.Kumarasamy College of Engineering,	M.Kumarasamy College of Engineering,
Thalavapalayam, Karur-639113	Thalavapalayam, Karur-639113

This project report has been submitted for the **18ECP107L - Project Work** Viva Voce Examination held at M.Kumarasamy College of Engineering, Karur on ______________.

INTERNAL EXAMINER **EXTERNAL EXAMINER**

INSTITUTION VISION AND MISSION

Vision

To emerge as a leader among the top institutions in the field of technical education.

Mission

M1: Produce smart technocrats with empirical knowledge who can surmount the global challenges.

M2: Create a diverse, fully -engaged, learner -centric campus environment to provide quality education to the students.

M3: Maintain mutually beneficial partnerships with our alumni, industry and professional associations

DEPARTMENT VISION, MISSION, PEO, PO AND PSO

Vision

To empower the Electronics and Communication Engineering students with emerging technologies, professionalism, innovative research and social responsibility.

Mission

M1: Attain the academic excellence through innovative teaching learning process, research areas & laboratories and Consultancy projects.

M2: Inculcate the students in problem solving and lifelong learning ability.

M3: Provide entrepreneurial skills and leadership qualities.

M4: Render the technical knowledge and skills of faculty members.

Program Educational Objectives

PEO1: **Core Competence:** Graduates will have a successful career in academia or industry associated with Electronics and Communication Engineering

PEO2: **Professionalism:** Graduates will provide feasible solutions for the challenging problems through comprehensive research and innovation in the allied areas of Electronics and Communication Engineering.

PEO3: **Lifelong Learning:** Graduates will contribute to the social needs through lifelong learning, practicing professional ethics and leadership quality

Program Outcomes

PO 1: Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO 2: Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO 3: Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO 4: Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO 5: Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO 6: The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO 7: Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO 8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO 9: Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO 10: Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO 11: Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO 12: Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Specific Outcomes

PSO1: Applying knowledge in various areas, like Electronics, Communications, Signal processing, VLSI, Embedded systems etc., in the design and implementation of Engineering application.

PSO2: Able to solve complex problems in Electronics and Communication Engineering with analytical and managerial skills either independently or in team using latest hardware and software tools to fulfil the industrial expectations.

Abstract	Matching with POs, PSOs
Internet of Things, Smart Irrigation, Renewable Energy, Cloud Computing, Crop Monitoring, Precision Farming	PO1, PO2, PO3, PO4, PO5, PO6, PO7, PO9, PO10, PO11, PO12, PSO1, PSO2

S.No.	Project Domain	Mapping with POs/PSOs
1.	IoT	PO1, PO2, PO3, PO4, PO5, PO6, PO7, PO9, PO10, PO11, PO12, PSO1, PSO2

ACKNOWLEDGEMENT

Our sincere thanks to **Thiru.M.Kumarasamy, Founder** and **Dr.K.Ramakrishnan, Chairman** of **M.Kumarasamy College of Engineering** for providing extraordinary infrastructure, which helped us to complete this project in time.

It is a great privilege for us to express our gratitude to **Dr.B.S.Murugan., B.Tech., M.Tech., Ph.D., Principal** for providing us right ambiance to carry out this project work.

We would like to thank **Dr.A.Kavitha, M.E., Ph.D., Professor and Head**, **Department of Electronics and Communication Engineering** for her unwavering moral support and constant encouragement towards the completion of this project work.

We offer our wholehearted thanks to our **Project Supervisor, Mrs.P.Sanmugavalli, M.E., Assistant Professor**, Department of Electronics and Communication Engineering for her precious guidance, tremendous supervision, kind cooperation, valuable suggestions and support rendered in making our project to be successful.

We would like to thank our **Class Advisor, Dr.S.Vimalnath, M.E., Ph.D., Associate Professor,** Department of Electronics and Communication Engineering and **Project Co-ordinator, Dr.A.Maheshwari, M.E., Ph.D.,** Assistant Professor, Department of Electronics and Communication Engineering for their kind cooperation and culminating in the successful completion of this project work.

We are glad to thank all the **Faculty Members** of the **Department of Electronics and Communication Engineering** for extending a warm helping hand and valuable suggestions throughout the project.

Words are boundless to thank our **Parents** and **Friends** for their motivation to complete this project successfully.

ABSTRACT

Road traffic accidents (RTAs) are becoming more common nowadays, as evidenced by the fact that the number of accidents is increasing on a daily basis. Many a times an accident goes unnoticed for hours before help comes in. Due to all these factors, there is a high rate of mortality among the accident victims. In addition to this there is delay in the ambulance reaching the hospital due to the traffic congestion between accident location and hospital which increases the chances of the death of victim. Some various methods and techniques can assist in minimizing the frequency of traffic road accidents and saving lives. There is a need of introducing a system to reduce the loss of life due to accidents and the time taken by the ambulance to reach the hospital. This will result in lives being saved and injured persons being rescued. In this project, to this end, propose an accident reporting, ambulance positioning, alert ambulance and rescue system named Vehicle BlackBox Model based on IoT and Deep Learning for Intelligent Transportation System in urban environments. The proposed project to shorten the response time for ambulance arrival at the scene of a road accident by employing an unsupervised generative clustering approach with Variational Deep Embedding (VaDE) sounds innovative and promising. The proposed method is categorized into two phases: accident detection, based on the accident detected location predict the ambulance to arrive the spot, alert the ambulance driver to reach the spot, real-time alerts to family members, hospitals, and traffic departments, facilitating route clearance for expedited ambulance travel. Another phase of the proposed system is to position the ambulance based on the data set given to the VaDE. This proposed system contributes to reducing the loss of life in road traffic accidents, minimizing response times, and improving overall emergency response efficiency.

TABLE OF CONTENTS

CHAPTER No.	CONTENTS	PAGE No.
	INSTITUTION VISION AND MISSION	iii
	DEPARTMENT VISION AND MISSION	iii
	DEPARTMENT PEOs, POs AND PSOs	iv
	ABSTRACT	ix
	LIST OF TABLES	xiii
	LIST OF FIGURES	xiv
	LIST OF ABBREVIATIONS	xv
1	**INTRODUCTION**	**1**
	1.1 OVERVIEW	1
	1.2 PROBLEM STATEMENT	5
	1.3 EMBEDDED DEVICES	8
	1.4 VARIATIONAL DEEP EMBEDDING	14
	1.5 OBJECTIVE	16
	1.6 SCOPE OF THE PROJECT	18
2	**LITERATURE SURVEY**	**20**
3	**EXISTING SYSTEM**	**28**
	3.1 EXISTING AUTOMATION	29
	3.2 DISADVANTAGES	31
4	**PROPOSED SYSTEM**	**33**
	4.1 ADVANTAGES	36
5	**SYSTEM SPECIFICATION**	**39**
	5.1 HARDWARE REQUIREMENTS	39
	5.2 SOFTWARE REQUIREMENTS	40

6	**SOFTWARE DESCRIPTION AND FUTURE WORK**	**41**
	6.1 PYTHON 3.7.4	41
	6.2 EMBEDDED C	44
	6.3 PHP 8.1	45
	6.4 MYSQL	46
	6.5 WAMPSERVER	47
	6.6 BOOTSTRAP 4	48
	6.7 FLASK	49
	6.8 SMS API	49
	6.9 GOOGLE MAP API	50
7	**SYSTEM DESIGN**	**52**
	7.1 SYSTEM ARCHITECTURE	52
	7.2 DATA FLOW DIAGRAM	53
	7.3 UML DIAGRAM	55
8	**SYSTEM TESTING**	**59**
	8.1 SOFTWARE TESTING	59
	8.2 TEST CASES	62
	8.3 TEST REPORT	67
9	**SYSTEM IMPLEMENTATION**	**70**
	9.1 SYSTEM DESCRIPTION	70
	9.2 SYSTEM FLOW	71
	9.3 MODULES DESCRIPTION	74
	9.4 BLOCK EXPLANATION	83

10	**RESULTS & DISCUSSION**	**90**
11	**CONCLUSION AND FUTURE EXHANCEMENT**	**93**
	11.1 CONCLUSTION	93
	11.2 FUTURE ENHANCEMENT	94
12	**APPENDICES**	**95**
13	**REFERENCES**	**120**

LIST OF TABLES

TABLE No.	TITLE	PAGE No.
2.1	SUMMARY OF LITERATURE SURVEY	25
8.3.1	BUG REPORT	68

LIST OF FIGURES

FIGURE No.	TITLE	PAGE No.
1.1	Road Accident	1
1.3	Embedded System in IoT	8
1.3.1	Embedded System	9
1.3.2	Embedded System Software	10
1.4.1	VAE Architecture	15
1.4.2	The model Architecture of VaDE	16
4.1	Proposed system Block diagram	38
7.1.1	System Architecture	52
7.2.1	Level 0 Architecture	53
7.2.2	Level 1 Architecture	53
7.2.3	Level 2 Architecture	54
7.3.1	Use case Diagram	55
7.3.2	Class diagram	56
7.3.3	Sequence diagram	56
7.3.4	Activity Diagram	57
7.3.5	Collaboration Diagram	58
7.3.6	Deployment Diagram	58
9.3.1	Modules Description Block Diagram	84
9.4.1	ESP8266-01 Module Pins	88
10.1	IOT Module	92
10.2	Positioning the ambulance	93

LIST OF ABBREVIATIONS

ACRONYM		ABBREVIATION
API	-	Application Program Interface
ATV	-	All-Terrain Vehicle
DNN	-	Deep Neural Networks
DUI	-	Driving Under the Influence
GMM	-	Gaussian Mixture Model
GPS	-	Global Positioning System
IC	-	Integrated Circuit
IoT	-	Internet of Things
MoRTH	-	Ministry of Road Transport and Highways
MVA	-	Motor Vehicle Act
NAC	-	National Ambulance Code
NHAI	-	National Highway Authority of India
OS	-	Operating System
PCS	-	Power Conversion System
PHP	-	Hypertext Preprocessor
RTOS	-	Real-time Operating Systems
SDLC	-	Software Development Lifecycle
SGVB	-	Stochastic Gradient Variational Bayes
VaDE	-	Variational Deep Embedding
WHO	-	World Health Organization

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

Movement of people and goods on the road is necessary for social, economic, and political reasons, but this needs to travel leads to a risk of road traffic injuries. Road accidents are the most unwanted thing to happen to a road user, though they happen quite often. A total of 4,61,312 road accidents occurred in 2022, which claimed 1,68,491 lives, while 4,43,366 people were injured, according to a new report released by the Ministry of Road Transport and Highways (MoRTH).

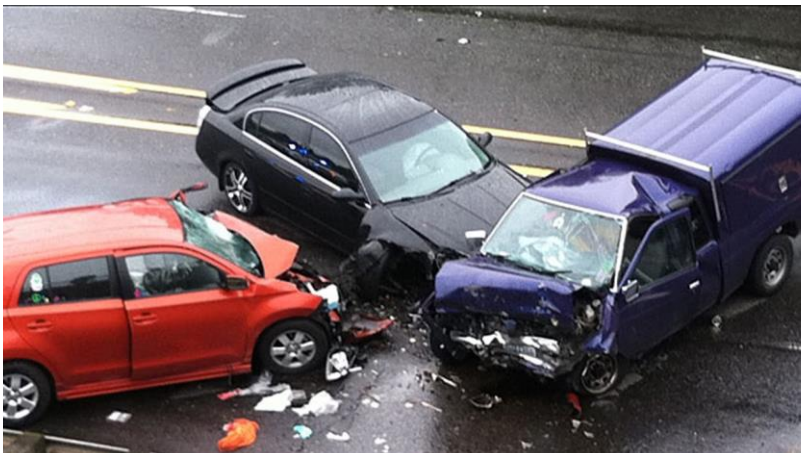


Figure 1.1 Road Accident

Main cause of accidents and crashes are due to human errors. Elaborating some of the common behavior of humans which results in accident.

- Over Speeding
- Drunken Driving
- Distractions to Driver
- Red Light Jumping
- Avoiding Safety Gears like Seat belts and Helmets

- Non-adherence to lane driving and overtaking in a wrong manner.

Various national and international researchers have found these as the most common behavior of Road drivers, which leads to accidents.

1.1.1 Types of Motor Vehicle Accidents

Nearly any type of vehicle has the potential to be involved in a traffic collision. Additionally, non-motorists, such as cyclists or pedestrians, are also at risk of being involved in an accident.

- **Car Accidents**

Car accidents can involve multiple vehicles or just a single car, and can range in severity from minor fender benders to catastrophic collisions.

- **Motorcycle Accidents**

Motorcyclists have gained a bad reputation among other drivers as being reckless or careless on the road. However, most motorcycle accidents are actually the result of another driver failing to adequately check their mirrors before changing lanes, or adverse weather and road conditions.

- **Truck Accidents**

Accidents involving large commercial trucks often result in much more serious injuries, especially for drivers of smaller vehicles. Despite the many rules and obligations truck drivers must adhere to, many go without sufficient rest, putting themselves and others at risk.

- **Bicycle Accidents**

As with motorcyclists, bicycle accidents are often the result of another driver's negligence. Some drivers feel bicyclists have no place on the road and drive aggressively, which can create a dangerous situation for all parties involved.

- **Bus Accidents**

Accidents involving buses and other mass transit vehicles commonly result in severe injuries for passengers and other motorists because of their large size and lack of safety restraints. Victims of these types of accidents require the assistance of a personal injury attorney to help determine liability for their injuries.

- **Boat Accidents**

Boating accidents are unfortunately common and can turn a pleasant vacation or outing into a painful memory for those involved. In addition to the injuries the accident can cause, there is the potential for drowning. Even for strong swimmers, an injury can prevent victims from keeping themselves afloat.

- **Passenger Van Accidents**

Accidents involving passenger vans can also typically involve injuries to numerous parties. Many schools, churches, and senior homes use vans of this type, and serious accidents could result in injuries to young children or elderly passengers.

- **Pedestrian Accidents**

Pedestrian accidents can involve any type of motor vehicle accident that results in an injury to a pedestrian. A distracted driver may not see a pedestrian until it is too late to stop, or the driver may not see the pedestrian at all. Pedestrian accidents can also include instances in which a driver breaches a sidewalk or strikes someone who is crossing the street.

- **ATV Accidents**

Like motorcycles, ATV accidents often involve severe injuries because the rider is left vulnerable without safety equipment. Most recreational vehicle drivers wear a helmet, but their bodies are still exposed in a crash.

- **DUI Accidents**

Unfortunately, DUI accidents involving injuries occur roughly every two minutes in this country. A driver under the influence of alcohol, prescription medications, or illicit drugs can be arrested for DUI. In addition to criminal charges, offending drivers can be held financially responsible for any injuries that result.

1.1.2 Ambulances in India

Ambulances in India in India, ambulances are used primarily in three types of situations: during emergencies, to prompt transfer trauma patients to the nearest medical facilities; for transporting patients to and from their residences and hospitals; and, for inter-hospital transfers. The most common mode of patient transportation is a road ambulance, which could be two three-or four-wheeler vehicles, depending on geographical location, terrain, and type of emergency. National Ambulance Code (NAC), under the aegis of the Ministry of Road Transport and Highways (MoRTH), the Government of India (GOI) classified road ambulances for registration under the provisions of the Motor Vehicle Act, 1988 (MVA, 1988).

1.1.3 Entities Providing Ambulances Services

The Government, Hospitals, Private Companies, Charitable Organizations, Religious Institutions, and Political parties are a few of the entities that provide ambulance services in India. The Government’s National Health Mission (NHM) operates National Ambulance Services (NAS) across most states and union territories.

They provide free emergency transportation with the Dial-108 model (Emergency Response System), which has one ambulance positioned for every 1,00,000 populations. The National Highway Authority of India (NHAI), GOI has deployed approximately 550 ambulances on national highways 6. Additionally, it operates the Incident Management Services, i.e., the provision of an ambulance, a patrol vehicle and a tow-away crane on national highway stretches of at least 60 km charitable institutions have been at the forefront of providing ambulance services in India. Private organizations work in partnership with state governments, hospitals and NGOs to offer ambulance services.

1.2 PROBLEM STATEMENT

In life-threatening road accidents, a call for emergency medical services (EMS) needs to go out quickly and unambiguously. Trauma is a "time-dependent disease," and basic life support may be needed soon after a crash. Any factor that might delay an EMS arrival at a road accident where an injury has occurred should be a matter of concern. However, ambulance delays have become a major obstacle for patients to receive immediate EMS assistance. Ambulance delays may worsen the patient’s injury. More than 20% of patients needing emergency treatment have died on their way to the hospital because of delays. Ambulances cannot get to emergencies fast enough. Often, the reason first responders cannot get patients to a hospital in time. The background of the project revolves around addressing the increasing challenges in emergency response due to rising traffic accidents globally. As per the World Health Organization (WHO), a significant number of people lose their lives or face injuries annually, often due to delayed medical assistance. The fundamental problem lies in the efficiency of ambulance deployment and response time, hindered by traffic congestion and suboptimal positioning. To tackle this issue, the project proposes a revolutionary approach, leveraging IoT for accident detection and intimation and deep learning based Variational Deep Embedding (VaDE) for intelligent ambulance positioning. By identifying accident-prone clusters and strategically deploying ambulances, the system aims to minimize response times, ultimately saving lives. Real-time alerts and a comprehensive NHAI Department User Interface further contribute to efficient emergency management on national highways. The project aligns with the vision of transforming emergency response strategies to meet the growing challenges of an increasing automobile population.

1.2.1 Factors influencing post-crash injury outcomes

Death is potentially preventable in a proportion of cases of people who die as a result of road crashes before they reach hospital. The potential help towards recovery that victims can receive may be viewed as a chain with several links:

- actions, or self-help, at the scene of the crash, by the victims themselves, or more frequently by bystanders.
- access to the emergency medical system.
- help provided by rescuers of the emergency services.
- delivery of medical care before arrival at the hospital.
- hospital trauma care.
- rehabilitative psychosocial care.

Inadequate post-crash care

Delays in detecting and providing care for those involved in a road traffic crash increase the severity of injuries. Care of injuries after a crash has occurred is extremely time-sensitive: delays of minutes can make the difference between life and death. Improving post-crash care requires ensuring access to timely prehospital care and improving the quality of both prehospital and hospital care, such as through specialist training programs.

Factors influencing post-crash injury outcomes, such as delay in detecting the crash and providing life-saving measures and psychological support. Road traffic injuries cause considerable economic losses to individuals, their families, and to nations. These losses arise from the cost of treatment as well as lost productivity for those killed or disabled by their injuries, and for family members who need to take time off work or school to care for the injured. Road traffic crashes cost most countries 3% of their gross domestic product.

Road traffic injuries can be prevented. Governments need to take action to address road safety in a holistic manner. This requires involvement from multiple sectors such as transport, police, health, education, and actions that address the safety of roads, vehicles, and road users.

Effective interventions include designing safer infrastructure and incorporating road safety features into land-use and transport planning, improving the safety features of vehicles, improving post-crash care for victims of road crashes, setting, and enforcing laws relating to key risks, and raising public awareness.

Systems approach

A perspective that considers the various parts and their relationships as they contribute to the totality of a phenomenon. In the case of road traffic injury prevention, this calls for a comprehensive understanding of the risk factors, determinants, impacts and interventions, as well as consideration of the role of different agencies and stakeholders in prevention.

This lack of rapid care and first aid might result in life loss in a matter of minutes. To address all these challenges, an intelligent system is necessary. Although several information communications technologies-based solutions for accident detection and rescue operations have been proposed, these solutions are not compatible with all vehicles and are also costly.

1.3 EMBEDDED DEVICES

It is essential to know about the embedded devices while learning the IoT or building the projects on IoT. Embedded devices are the objects that build the unique computing system. These systems may or may not connect to the Internet. An embedded device system generally runs as a single application. However, these devices can connect through the internet connection, and are able to communicate through other network devices.

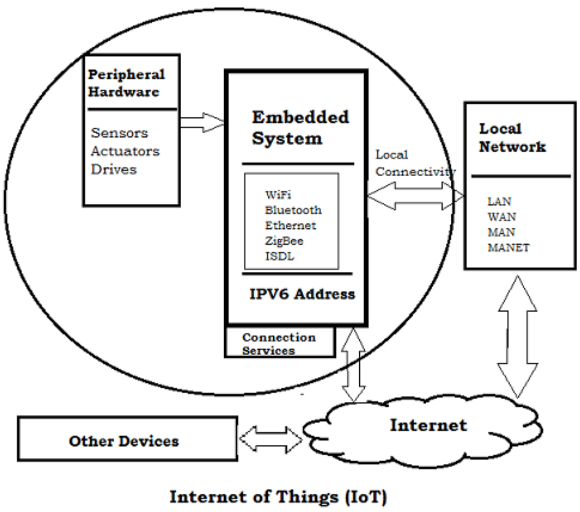


Figure 1.3 Embedded System in IoT

Embedded systems will play an important role in Internet of Things (IoT) due to their unique characteristics and features such as real time computing, low power consumption, low maintenance and high availability are becoming the key enabler of IoT. Major players in embedded system hardware and software developments are aiming to bring these transformations into their products to take advantage of the growing IoT market.

1.3.1 Embedded System Hardware

The embedded system can be of type microcontroller or type microprocessor. Both types contain an integrated circuit (IC). The essential component of the embedded system is a RISC family microcontroller like Motorola 68HC11, PIC 16F84, Atmel 8051, ATmega328 and many more. The most important factor that differentiates these microcontrollers with the microprocessor like 8085 is their internal read and writable memory. The essential embedded device components and system architecture are specified below.

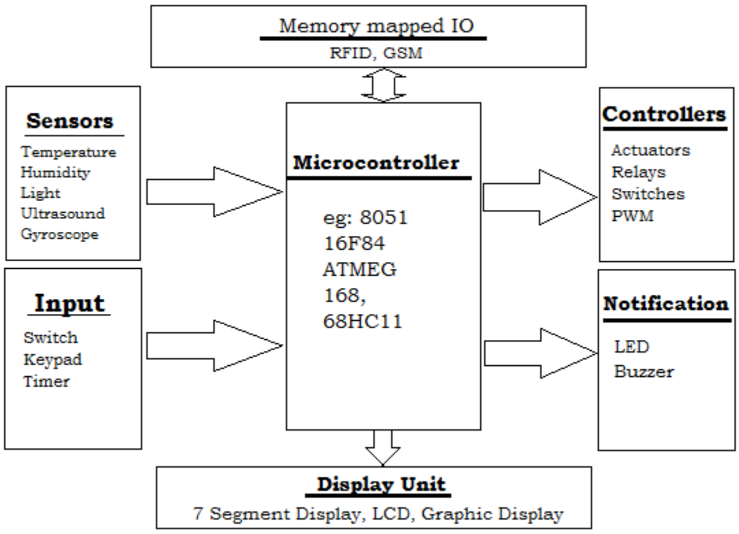


Figure 1.3.1 Embedded System

1.3.2 Embedded System Software

Embedded software is written for the device drivers, operating system, and applications, as well as for error handling and debugging. Software components of embedded system include:

- Device Driver: A device driver is an embedded code written for a specific piece of hardware.
- Operating System (OS) or MicroOS: Embedded systems have a gamut of operating systems, including RTOS (Real-time Operating Systems), mobile embedded, stand-alone and network embedded systems.

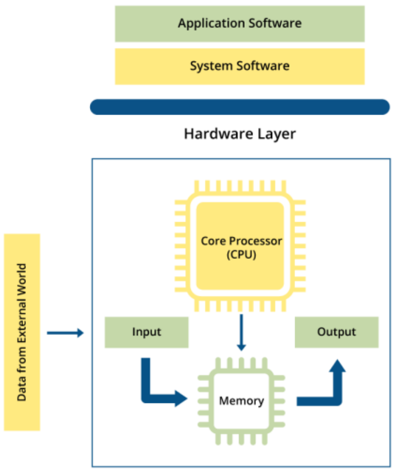


Figure 1.3.2 Embedded System Software

Most of the embedded software uses any of the two languages: C and C++. C++ has features, like enhanced security and closeness to real-world applications, whereas C is more reliable and has better performance by directly interacting with the hardware.

1.3.3 Role of Embedded systems on the Internet of Things

Internet of Things is the concept of connecting devices via the internet to exchange data. It is the most trending technology in this modern world as we can control embedded devices from any location using the Internet of things. In sum, the Internet of Things (IoT) is a process in which objects are equipped with sensors, actuators, and processors that involve hardware board design and development, software systems, web APIs, and protocols, which together create a connected environment of embedded systems.

1.3.4 Applications for IoT

Building and home automation

From enhancing security to reducing energy and maintenance costs, TI offers a wide range of innovative nano power, signal chain, sensing, power management and wireless connectivity devices that address key system challenges in intelligent building and smart home applications such as predictive maintenance, video surveillance, smart lighting and control and HVAC.

- Air quality & gas detection
- Door & window sensor
- HVAC gateway
- Electronic smart lock
- Thermostat
- Smart speaker
- Motion detector (PIR, uWave, etc.)
- Wireless environmental sensor
- Video doorbell
- Water leak detector

Smart cities

Reduce cost and resource consumption with TI's products for IoT-related designs for outdoor lighting, surveillance, long range wireless connectivity, centralized & integrated system control and more.

- AC charging (pile) station
- Data concentrator
- DC charging (pile) station
- Drones
- Electricity meter
- Gas meter
- Micro inverter
- Energy storage power conversion system (PCS)
- String inverter
- IP network camera
- Water meter
- Wireless communications

Smart manufacturing

Smart factory and Industry 4.0 projects require secure and robust IoT designs. TI offers a wide range of innovative technologies to enable a smarter, safer, more efficient factory environment.

- Access control
- Actuator
- Machine vision camera
- CNC control
- Communication modules
- Communication switch
- Condition monitoring sensor
- CPU (PLC controller)
- Flow transmitter
- Industrial communication
- Industry 4.0
- Lighting

- Portable monitor
- Position sensor
- Proximity sensor
- Process analytics
- Robotics
- Temperature transmitter

Automotive

From headlights to taillights and all systems in between, TI offers a wide range of innovative technologies for the modern automobile.

- Automotive gateway
- Automotive head unit
- Emergency call (eCall)
- Powertrain temperature sensor
- On-board (OBC) & wireless sensor
- Passive entry passive start (PEPS)
- Telematics control unit

Wearables

With the broadest portfolio in the industry, TI provides highly efficient ultra-low power solutions for the wearables market.

Augmented reality glasses

- GPS personal navigation device
- Smartwatch
- Wearable fitness & activity monitor

Healthcare

TI is shaping technology to improve the quality and accessibility of digital products that are revolutionizing the health and fitness industries.

- Multiparameter patient monitor
- Telehealth systems

Agriculture

Reduce time to market in your precision agriculture design with TI devices and reference designs.

- Industrial transport (non-car & non-light truck)

1. Drones
2. Off-highway vehicles sensor systems

1.4 VARIATIONAL DEEP EMBEDDING (VaDE)

Variational autoencoders (VAE) are used as one of the most important concepts in this thesis. They were first introduced by Deidrick P. Kingma and Max Welling in 2014. Applications using VAE include aircraft turbomachinery design, anomaly detection, image analysis and more. The model is different from other autoencoders in two ways. First, it is a probabilistic autoencoder. Second, it is a generative autoencoder. First, it is a probabilistic autoencoder. Second, it is a generative autoencoder. The network of the autoencoder is shown in Figure 2.5. It is the same as other autoencoders with an encoder followed by a decoder. The difference is that it outputs μ the mean coding, and σ the standard deviation. In practice, the output is γ where $\gamma = \log(\sigma^2)$. From the μ and σ, the actual latent space is then sampled by $z = \mu + \sigma * €$, where € is a random variable drawn from a normal distribution N (0, I), where I is the identity matrix. After that, the decoder decodes the latent space normally, the output resembles the training input.

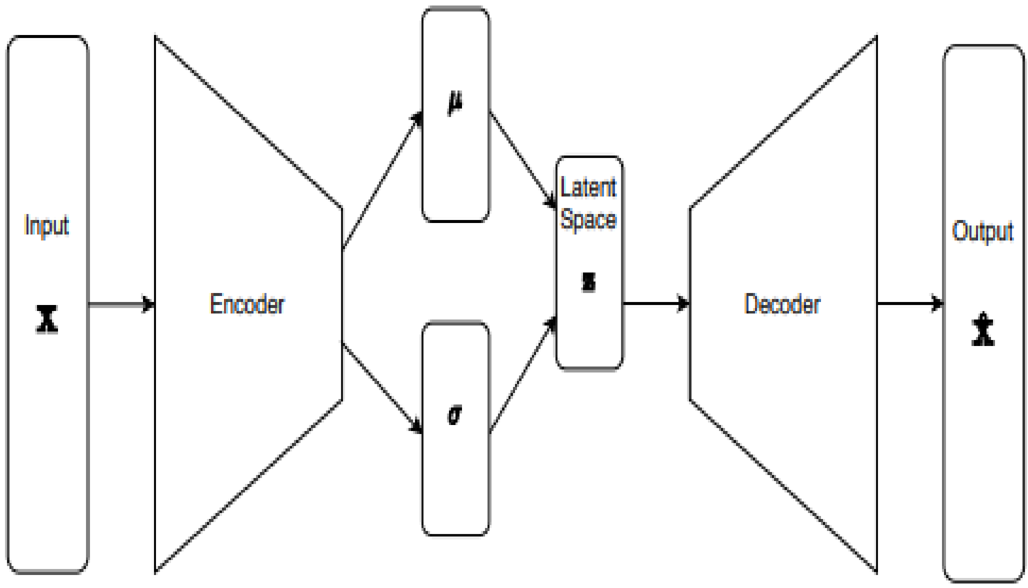


Figure. 1.4.1 VAE Architecture

Variational Deep Embedding (VaDE), a novel unsupervised generative clustering approach within the framework of Variational Auto-Encoder (VAE). Specifically, VaDE models the data generative procedure with a Gaussian Mixture Model (GMM) and a deep neural network (DNN): 1) the GMM picks a cluster; 2) from which a latent embedding is generated; 3) then the DNN decodes the latent embedding into an observable. Inference in VaDE is done in a variational way: a different DNN is used to encode observables to latent embeddings, so that the evidence lower bound (ELBO) can be optimized using Stochastic Gradient Variational Bayes (SGVB) estimator and the reparameterization trick.

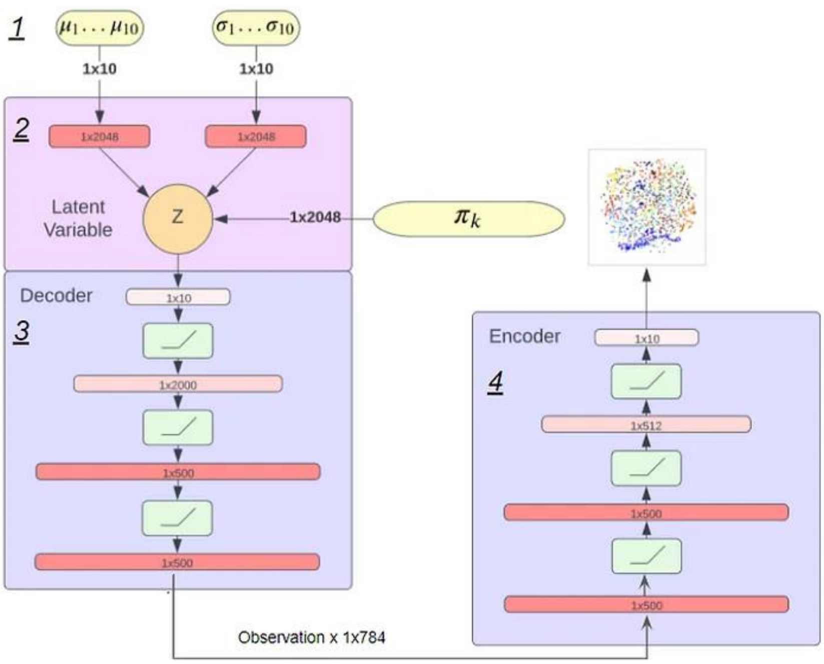


Figure 1.4.2 The model Architecture of VaDE

Variational Deep Embedding shows capability in solving the clustering problem using an architecture similar to VAE. It models the data generative process using a GMM model and a neural network.

1. A Cluster is selected from a Gaussian Mixture Model, and it's mean, and log variance is fed into the network.
2. A latent embedding is generated based on the picked cluster.
3. A DNN decodes the latent embedding into an observable x.
4. An Encoder network is used to maximize the ELBO of VaDE

1.5 OBJECTIVE

The primary object of the project is to significantly reduce the impact of road traffic accidents by leveraging advanced technologies, including IoT and deep learning, to create an intelligent and responsive system. This system is designed to expedite emergency response times, optimize ambulance positioning, and improve overall coordination for effective accident management.

- **To Achieve Real-Time Accident Detection**

Develop sensors and algorithms for immediate accident identification.

Utilize accelerometers, collision sensors, and over-speed sensors.

- **To Optimize Ambulance Positioning**

Implement algorithms predicting optimal ambulance locations.

Utilize unsupervised generative clustering (VaDE) for positioning efficiency.

- **To Establish Communication and Alerts**

Develop a real-time alert system for family, hospitals, and traffic.

Enable effective communication between victims and emergency services.

- **To Manage Route Clearance and Traffic**

Collaborate for route clearance with traffic departments.

Provide real-time traffic updates and alternative routes.

- **To Implement Accident Reporting and Rescue System**

Set up a reporting mechanism for detailed accident information.

Develop an intuitive user interface for victims to request help and report details.

- **To Apply VaDE for Optimization**

Utilize VaDE for unsupervised clustering to optimize ambulance positioning.

Train the model with historical data for improved accuracy.

- **To Develop User Interface**

Create an accessible interface for victims, family, and responders.

Include real-time tracking and reporting features.

- **To Conduct Testing and Iteration**

Conduct rigorous testing in simulated and real-world scenarios.

Iterate based on user and stakeholder feedback.

- **To Deploy and Ensure Scalability**

Deploy in urban areas with high traffic.

Ensure scalability for increased users and emergencies.

1.6 SCOPE OF THE PROJECT

The scope of the project is to revolutionize emergency response strategies in road traffic accidents. It encompasses real-time accident detection through advanced sensors, optimizing ambulance positioning using unsupervised clustering, establishing a robust communication and alert system, managing route clearance and traffic in collaboration with relevant authorities, implementing a comprehensive accident reporting and rescue system with a user-friendly interface, leveraging deep learning for continuous optimization, developing an intuitive user interface, rigorous testing with iterative improvements, and deploying the system in urban areas with a focus on scalability and continuous enhancement. Designing and implementing a system to collect data from various sensors within the vehicle, such as accelerometers, GPS, cameras, and other relevant sensors. Integrating IoT technology to enable real-time data transmission from the vehicle to a centralized system. Developing the core functionality of the vehicle black box, including data logging, event detection, and storage capabilities. Implementing mechanisms to record critical events such as accidents, sudden braking, swerving, or other irregular driving behavior. Developing deep learning models to analyze the data collected by the black box in real-time. Training models for various tasks such as driver behavior analysis, road condition assessment, object detection, and collision prediction. Integrating the deep learning models with the black box system to provide real-time safety alerts and warnings to drivers. Implementing mechanisms to alert drivers of potential hazards, drowsiness, distraction, or aggressive driving behavior. Designing a user-friendly interface to visualize the data collected by the black box and the insights generated by the deep learning models. Providing detailed reports and analytics to users, fleet managers, and authorities for proactive safety measures and accident prevention. Implementing remote monitoring capabilities to allow authorities or designated personnel to access real-time data from vehicles. The scope of integrating a vehicle black box with IoT and deep learning to save lives encompasses several key areas. These include data collection and sensor integration, IoT connectivity, development of deep learning models for analyzing driving behavior, real-time analysis and alert generation, driver feedback and intervention mechanisms, data logging and reporting, scalability and compatibility considerations, regulatory compliance and ethical considerations, testing and validation, and deployment and adoption strategies. The project aims to leverage these technologies to enhance vehicle safety, prevent accidents and promote safer driving behavior, ultimately saving lives. By encompassing these aspects, the project aims to leverage IoT and deep learning technologies to enhance vehicle safety, potentially saving lives by preventing accidents and promoting safer driving behavior.

CHAPTER 2

LITERATURE SERVEY

T.D.Hanawy Hussein, M.Frikha and J.Rahebi proposed the ambulance routing problem is one of the capacitated ambulance routing problem forms. It deals with injuries and their requests for saving. Therefore, the main aim of the ambulance routing problem is to determine the minimum (i.e., optimum) required distances between 1) accident places and the ambulance station; 2) the location of the nearest hospital and the accident places. Despite the efforts proposed in the literature, determining the optimum route is crucial. In our system, let's use the node approach to produce a city map. Initially, the control station receives accident site information and sends it to the hospital and the ambulance. The HHO vehicle routing algorithm receives data from the driver; the data includes the location of the accident and the node position of the ambulance vehicle. Then, the driver's shortest route to the accident scene by the HHO. The locations of the accident and hospital are updated by the driver once the car reaches the accident site. The fastest route (which results in the least travel time) to the hospital is then determined. The HHO can provide offline information for a potential combination of the coordinates of destination and source. Extensive simulation experiments demonstrated that the HHO can provide optimal solutions. Furthermore, performance evaluation experiments demonstrated the superiority of the HHO algorithm over its counterparts (SAODV, TVR, and TBM methods). Furthermore, for ten malicious nodes, the PDF of the algorithm was 0.91, which is higher than the counterparts [1].

Chowdhury, S.Kaisar, M.E.Khoda, R.Naha, M.A.Khoshkholghi and M.Aiash Proposed Emergency Management System (EMS) is an important component of Intelligent transportation systems, and its primary objective is to send Emergency Vehicles (EVs) to the location of a reported incident. However, the increasing traffic in urban areas, especially during peak hours, results in the delayed arrival of EVs in many cases, which ultimately leads to higher fatality rates, increased property damage, and higher road congestion. Existing literature addressed this issue by giving higher priority to EVs while traveling to an incident place by changing traffic signals (e.g., making the signals green) on their travel path. A few works have also attempted to find the best route for an EV using traffic information (e.g., number of vehicles, flow rate, and clearance time) at the beginning of the journey. However, these works did not consider congestion or disruption faced by other non-emergency vehicles adjacent to the EV travel path. The selected travel paths are also static and do not consider changing traffic parameters while EVs are en route. To address these issues, this article proposes an Unmanned Aerial Vehicle (UAV) guided priority-based incident management system to assist EVs in obtaining a better clearance time in intersections and thus achieve a lower response time. The proposed model also considers disruption faced by other surrounding non-emergency vehicles adjacent to the EVs' travel path and selects an optimal solution by controlling the traffic signal phase time to ensure that EVs can reach the incident place on time while causing minimal disruption to other on-road vehicles [2].

U. Mittal and P. Chawla Proposed The temporal and spectral structure is possessed in the time-frequency domain by sound events. Analyzing and classifying acoustic environment using sound recording is an emerging research area. Convolutional layers can quickly extract high-level features and shift-invariant features from the time-frequency domain. In this work, emergency vehicle detection (EVD) like fire brigades, ambulances, and police cars is done based upon their siren sounds. Dataset from Google Audio set ontology was collected and features are extracted by Mel-frequency Cepstral Coefficient (MFCC). Three deep neural networks (DNN) models (dense layer, Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN)) with different configurations and parameters have been investigated. Then, an ensemble model has been designed with optimum selected models by performing experimental tests on various configurations with hyper-parameter tuning. The proposed ensemble model provides the highest accuracy of 98.7%, while the recurrent neural network (RNN) model provides an accuracy of 94.5%. Also, performance analysis of deep learning models is done with various machine learning models like Perceptron, SVM, decision tree etc [3].

T.Darwassh, Hanawy Hussein, M.Frikha, S.Ahmed and J.Rahebi Proposed an ambulance vehicle routing approach in smart cities. The approach is based on the bat algorithm and convolutional neural network (BA-CNN). It aims to transfer the patients confidentially, accurately, and quickly. The type of CNN used in this research is a residual network (ResNet). The node method is responsible for creating the city map. In the beginning, information about the accident place is received by the control station and forwarded to both the hospital and the ambulance. The driver feeds the data that contains the ambulance vehicle's node position and the accident location to the BA-CNN vehicle routing algorithm. The algorithm then obtains the shortest path to reach the location of the accident by the driver. When the vehicle arrives at the accident location, the driver updates the algorithm with hospital and accident positions. Then, the shortest path (which leads to the fast reach time) to the hospital is calculated. The bat algorithm provides offline data for a possible combination of different source and destination coordinates. The offline data are then trained by utilizing a neural network. The neural network is used for finding the shortest routes between source and destination. The performance evaluation of the BA-CNN algorithm is based on the following metrics: end-to-end delay (EED), throughput, and packet delivery fraction (PDF). This BA-CNN is compared with counterparts, including three different existing methods such as TBM, TVR, and SAODV. The experiments demonstrate that the PDF of our method is 0.90 for 10 malicious nodes, which is higher than in the TBM, TVR, and SAODV [4].

G. Liu, J. Qu, X. Li and J. Wu Proposed to solve the problems of a slow solving speed and easily falling into the local optimization of an ore-blending process model (of polymetallic Mult objective open-pit mines), an efficient ore-blending scheduling optimization method based on multiagent deep reinforcement learning is proposed. Firstly, according to the actual production situation of the mine, the optimal control model for ore blending was established with the goal of minimizing deviations in ore grade and lithology. Secondly, the open-pit ore-matching problem was transformed into a partially observable Markov decision process, and the ore supply strategy was continuously optimized according to the feedback of the environmental indicators to obtain the optimal decision-making sequence. Thirdly, a multiagent deep reinforcement learning algorithm was introduced, which was trained continuously and modeled the environment to obtain the optimal strategy. Finally, taking a large open-pit metal mine as an example, the trained multiagent depth reinforcement learning algorithm model was verified via experiments, with the optimal training model displayed on the graphical interface. The experimental results show that the ore-blending optimization model constructed is more in line with the actual production requirements of a mine. When compared with the traditional Mult objective optimization algorithm, the efficiency and accuracy of the solution have been greatly improved, and the calculation results can be obtained in real-time. A Survey of Artificial Intelligence and its Application in Collision Avoidance Systems for Intelligent Transportation Systems. Transportation Research Part C: Emerging Technologies. Pham, H. T., Nguyen, T. H., & Le, D. N. (2019). Deep Learning in Vehicular Ad-Hoc Networks: A Review. Journal of Network and Computer Applications. Deep Learning for Computational Intelligence in Autonomous Vehicles: A Review. Neural Networks. Driving Style Recognition using Deep Learning: Recent Advances and Future Directions. IEEE Transactions on Intelligent Transportation Systems. A Survey on Deep Learning for Intelligent Vehicles: Past, Present, and Future. IEEE Transactions on Intelligent Transportation Systems [5].

2.1 SUMMARY OF LITERATURE SURVEY

S. No	Title of the Paper	Author & Year	Methodology	Demerits
1.	Optimal Ambulance Positioning for Road Accidents With Deep Embedded Clustering	Sandeep Kumar Satapathy & 2023	This paper proposes an approach for the automatic placement of paramedic help using Deep Embedded Clustering (DEC).	The complexity of the model may make it challenging to train on large datasets or with limited computational resources.
2.	Automatic Accident Detection and Reporting System using NodeMCU	Arnikaa. A & 2023	The process is completed when an accident occurs, the MEMS sensor analyses the signal, and the microcontroller receives this signal. To alert the notification is transmitted via the GSM	Integrating a GSM module into an IoT device may involve dealing with SIM cards, mobile carrier configurations, and additional complexities.

3.	Smart Accident Detection And Alert System	Arnav Chaudhari & 2022	At the point when the vehicle is in a mishap, the vehicle's sensor distinguishes it promptly and sends an SMS to the crisis contacts along with Location.	While GSM networks have extensive coverage in many urban and suburban areas, there can still be coverage gaps, especially in remote or rural locations.
4.	Real-time road accident reporting system with location detection using cloud-based data analytics	Melissa Chong Jia Ying & 2022	The system then uses location detection and Global System for Mobile Communications (GSM) to accurately report accidents to the relevant authorities.	GSM relies on cellular networks, and the strength of the signal may vary depending on the location.

5.	Smart Accident Detection System	Murukurthi Lokesh Sai Kumar & 2021	The problem is approached by taking advantage of mobile communication networks and the Internet of Things. The paper proposes a simulation model that detects accidents using sensors.	GSM was primarily designed for voice and text communication. While it supports basic data services, it is not optimized for multimedia applications, such as video streaming or high-quality audio.

CHAPTER 3

EXISTING SYSTEM

Existing accident detection and intimation system typically relies on conventional methods for identifying and reporting road incidents. These systems may lack the advanced technologies found in modern solutions but serve the purpose of notifying relevant authorities and facilitating emergency responses. Key components of a traditional system include:

- **Manual Reporting**

In traditional systems, accident detection often relies on eyewitnesses or individuals involved in the incident to manually report the occurrence to emergency services or law enforcement.

- **Emergency Hotlines**

Users or witnesses can make use of dedicated emergency hotlines to report accidents. These hotlines are managed by emergency response agencies, allowing individuals to communicate details about the incident.

- **Law Enforcement Patrols**

Traditional accident detection involves law enforcement patrols actively patrolling roadways. They rely on visual observation and reports from witnesses to identify accidents and respond accordingly.

- **Traffic Cameras**

Some traditional systems deploy traffic cameras at strategic locations to monitor road conditions. These cameras may capture incidents, but they lack the advanced analysis capabilities of modern technologies.

- **Public Reporting**

Members of the public may report accidents to nearby police stations or emergency services. This reporting is often done through direct visits or phone calls.

3.1 EXISTING AUTOMATION

- Location-based Vehicle Accident Safety Measure (LVASM) system developed using the android platform.
- Vehicle tracking and accident detection system using Microcontroller, Raspberry Pi and Accelerometer.
- Eye Blink Monitoring system (EBM).
- IoT-based accident detection systems are being proposed which utilize cloud resources for the processing, storage, and transmission of data.
- Fog-based delay-aware accident management system (we call it ERDMS). The proposed system aims to decrease response and rescue time.
- The accident was detected through GPS speed.
- Accelerometer is used to detect accidents.
- Rescue system with accident detection.
- Detection of accidents at point of intersection
- Accelerometer is used to detect accidents.
- Mobile phones are used to detect accidents.
- GPS and Accelerometer are used to detect accidents.
- Alarm system with accident detection.
- The position of the vehicle is used for accident detection.
- Detection of shortest path and accident
- Reporting system with accident detection

- Vector machines are used to detect accidents.
- Detect the closed emergency point.
- Two sensors are used to detect the accident.
- The speed factor is used to detect accidents.
- Air bags are used to detect accidents.
- Reporting system with accident detection
- Collision detection and information system

Existing Clustering Algorithms

- **K-Means Clustering**

K-Means is a centroid-based clustering algorithm that partitions data into k clusters based on similarity. It works by iteratively assigning data points to the nearest centroid and updating the centroid's position. While effective for spherical clusters, it may struggle with irregularly shaped clusters and is sensitive to initial centroid placement.

- **Hierarchical Clustering**

Hierarchical clustering creates a tree-like structure of clusters, where the leaf nodes represent individual data points and internal nodes represent clusters. While providing a visual hierarchy, it can be computationally expensive for large datasets, and the choice of linkage method influences the results.

- **DBSCAN (Density-Based Spatial Clustering of Applications with Noise)**

DBSCAN identifies clusters based on dense regions, connecting data points within a specified density threshold. It's effective in detecting clusters of varying shapes and sizes but may struggle with varying densities and requires careful parameter tuning.

- **OPTICS (Ordering Points to Identify the Clustering Structure)**

OPTICS identifies clusters based on density but provides more flexibility in defining cluster shapes and sizes. It creates a reachability plot that helps visualize the clustering structure. However, like DBSCAN, it requires careful parameter selection.

- **Agglomerative Clustering**

Agglomerative clustering is a hierarchical method that starts with individual data points and iteratively merges clusters. It builds a tree structure of clusters, but it can be computationally intensive, especially for large datasets.

- **Fuzzy C-Means Clustering**

Fuzzy C-Means allows data points to belong to multiple clusters with varying degrees of membership. It assigns membership values to each data point, indicating the degree of association with each cluster. It is suitable for scenarios where data points may have ambiguous cluster assignments.

3.2 DISADVANTAGES

- Manual Reporting: Relies on eyewitnesses, causing reporting delays.
- Slow Response: Dependence on manual reporting leads to delayed emergency responses.
- Limited Accuracy: Lacks advanced tech, impacting precise accident detection.
- Incomplete Information: Eyewitnesses may lack comprehensive accident details.
- Technology Gap: Relies on outdated tech, missing advanced

solutions.

- Relies on predefined rules, lacking real-time adaptability.
- Limited predictive capabilities for optimal ambulance dispatch.
- Over-reliance on historical accident data, ignoring emerging trends.
- Manual route clearance causing delays in emergency responses.
- Static clustering algorithms may not adapt to dynamic changes.
- Communication gaps with hospitals and traffic departments.
- Limited GIS integration affects precision in ambulance positioning.
- Single-algorithm dependence restricts adaptability.
- Lack of real-time alerts to hospitals and traffic departments delays response.
- Collecting and transmitting large amounts of sensitive data from vehicles raises concerns about privacy and data security.
- The accuracy and reliability of deep learning algorithms heavily depend on the quality and quantity of training data.
- In real-world scenarios, factors such as adverse weather conditions, road infrastructure, and unpredictable human behavior can pose challenges to the accurate detection of safety risks.
- Despite the potential benefits, some drivers may resist or distrust the use of monitoring and intervention systems that track their behavior.

CHAPTER 4

PROPOSED SYSTEM

The proposed system, "LifeSaver," presents a cutting-edge approach to transforming emergency response strategies through the integration of advanced technologies. This system comprises several key modules aimed at optimizing ambulance positioning, enhancing real-time communication, and predicting ambulance dispatch for prompt and effective emergency responses.

- **Real-Time Accident Detection**

Implementing real-time accident detection involves deploying advanced sensors to promptly identify and respond to road incidents. This approach enhances the system's capability to provide immediate assistance and emergency responses. The specific components include:

Accelerometers

Utilize accelerometers to measure changes in the vehicle's acceleration. Sudden and abnormal changes may indicate a potential accident, triggering the system for further analysis.

Collision Sensors

Integrate collision sensors to detect the impact force or sudden deceleration associated with a collision. This helps in identifying accidents and assessing their severity.

Over-speed Sensors

Incorporate over-speed sensors to monitor the vehicle's speed in real-time. Detecting instances of excessive speed can be crucial in understanding the dynamics leading up to an accident.

Immediate Alert System

Implement an immediate alert system that is activated upon sensor detection of potential accidents. This system sends alerts to emergency services, family members, and relevant authorities in real-time.

Integration with IoT

Connect the sensors to an Internet of Things (IoT) platform for seamless data transmission and analysis. This integration enables quick and efficient communication of accident data to the central system.

- **VaDE-Based Clustering**

The VaDE-Based Clustering Module forms the foundation of the system, utilizing Variational Deep Embedding (VaDE) for unsupervised generative clustering. This sophisticated module integrates deep neural networks and Gaussian Mixture Models to accurately identify accident-prone clusters, providing a robust basis for optimizing ambulance positioning. In the encoding stage, pre-processed data undergoes transformation into latent representations using deep neural networks. VaDE's encoder network maps input data, such as accident features and locations, into a latent space—a condensed representation of hidden patterns within the dataset. Following this, the Gaussian Mixture Model (GMM) is employed to select clusters representing distinct accident scenarios, such as high-speed collisions or urban congestion. VaDE generates a latent embedding in a lower-dimensional space, encapsulating essential features of the chosen cluster. This serves as a compressed representation of underlying characteristics, including time of occurrence, severity, and geographic location. Variational inference is then applied to iteratively refine GMM parameters and optimize latent representations, balancing accuracy and computational efficiency. The latent embedding undergoes decoding through a Deep Neural Network (DNN), transforming it back into an observable format. This reconstructed information guides decisions on strategic ambulance positioning based on historical accident patterns. The algorithm assigns each historical accident to a specific cluster based on probabilistic clustering from the GMM, contributing to informed decision-making in ambulance deployment strategies. This cohesive process ensures the accurate identification of accident-prone areas and informs optimal ambulance placement for efficient emergency response.

- **Dynamic Ambulance Deployment**

The Dynamic Ambulance Deployment Module is designed to strategically deploy ambulances based on real-time demand and predictive insights. Targeting a critical five-minute drive time, this module adapts to changing conditions, continuously optimizing ambulance positions for dynamic emergency response. It aims to ensure timely assistance in critical situations.

- **Ambulance Positioning Simulator**

Integrating Geographic Information System (GIS), the Ambulance Positioning Simulator provides enhanced visualization. This module offers a real-time display of optimized ambulance positions on digital maps, facilitating dynamic route planning considering live traffic conditions. The simulator enhances situational awareness for prompt and effective emergency responses.

- **Ambulance Prediction System**

The Ambulance Prediction System utilizes a pre-trained Ambulance Deployment Model based on VaDE. This dynamic system predicts the optimal ambulance for a given incident, considering details such as accident severity and geographic coordinates.

- **Real-time Alert System Module**

Incorporating a robust Real-time Alert System, the proposed system enables instantaneous communication during emergencies. This module promptly notifies hospitals and traffic departments, facilitating swift route clearance for ambulances. The synchronized response contributes to minimizing delays and optimizing overall emergency response efficiency.

Real-Time Alert System Components

- **Traffic Department Alert:**
  - Notifies traffic departments promptly, providing real-time incident information for immediate route adjustments.
- **Hospital Notification:**
  - Alerts medical facilities about incoming emergencies, enabling timely preparations for patient care.
- **Intelligent Routing Suggestions:**

• Integrates with navigation and traffic management systems, suggesting efficient ambulance travel routes based on real-time conditions.

ADVANTAGES

- Immediate detection enables faster emergency assistance.
- Quick response minimizes mortality rates in accidents.
- Actively contributes to overall road safety.

- Advanced sensors provide immediate alerts for preventive measures.
- Optimized positioning ensures judicious use of emergency resources.
- Real-time detection enhances the overall reliability of the intelligent transportation system.
- Rapid ambulance deployment for quicker emergency response.
- Increased chances of saving lives through strategic ambulance placement.
- Optimal resource allocation for efficient emergency services.
- Instant communication with hospitals and traffic departments for route clearance.
- Data-driven insights for informed decision-making in emergency planning.
- Strategic positioning of ambulances in high-demand areas.
- Adaptive system for effective response to changing incident patterns.
- Geographic information systems for accurate accident-prone area visualization.
- Predictive modelling to forecast ambulance requirements based on accidents.
- Collaborative coordination among emergency stakeholders for synchronized response.
- Real-time Monitoring: Constantly monitors vehicle data, enabling immediate detection of anomalies or dangerous driving behaviors.
- Predictive Analytics: Deep learning algorithms can analyze patterns in driving behavior to predict potential accidents before they occur.
- Emergency Response: Instantly notifies emergency services in the event of a crash, providing accurate location and critical data to responders for faster assistance.
- Behavioral Insights: Provides insights into driver behavior, allowing for targeted interventions such as driver training or alerts for risky behavior.

BLOCK DIAGRAM

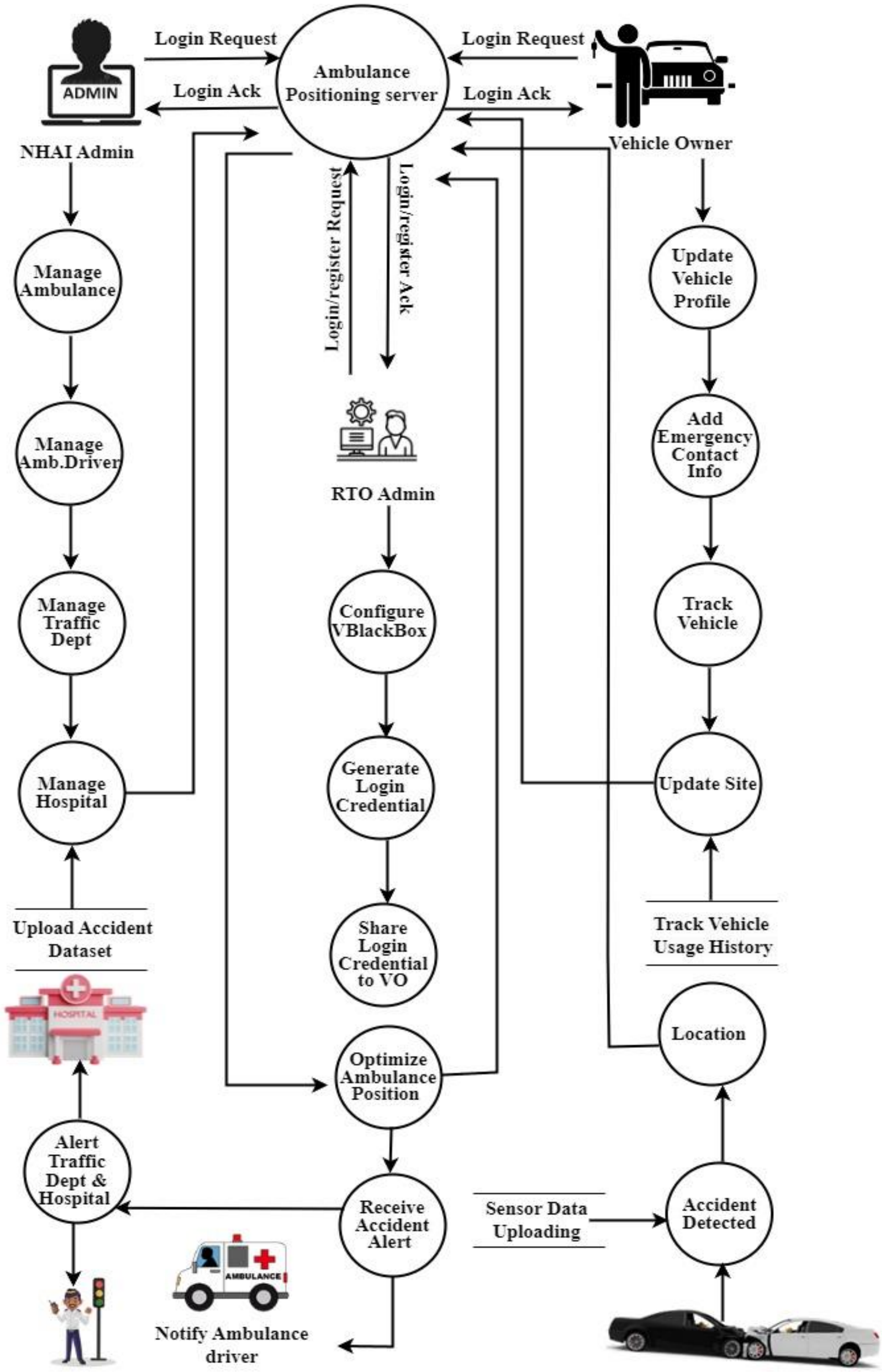


Figure 4.1 proposed system block diagram

CHAPTER 5

SYSTEM SPECIFICATION

5.1 HARDWARE REQUIREMENTS

- **Processor:** Intel Core i7 or equivalent AMD processor
- **RAM:** 16 GB or higher
- **Storage:** SSD (Solid State Drive) for faster data access
- **Accident Detection Sensors**
    - Accelerometers
    - Collision Sensors
    - Over-speed Sensors
- **GPS Module**
    - Integrated with ESP8266 for accurate location tracking.
- **IoT Connectivity**
    - Communication modules (Wi-Fi, cellular) for real-time data transmission
- **Central Processing Unit (CPU)**
    - Atmega 8 microcontroller for specific processing tasks
    - High-performance processor for overall system management
- **Power Supply**
    - Reliable power source with backup options to ensure continuous operation.
- **Vehicle Interface**
    - Integration with the vehicle's internal systems for enhanced data collection

5.2 SOFTWARE REQUIREMENTS

- **Development Environment**

Python: Programming language for backend server logic.

Integrated Development Environment (IDE): PyCharm

- **Database Management**

MySQL: RDBMS for storing and managing accident data, ambulance details, and system logs.

WampServer: Windows-based web development environment that includes Apache, MySQL, and PHP.

- **Web Application Framework**

Flask: Web application framework for developing the NHAI Ambulance Control Centres Web App.

- **Machine Learning Framework**

PyTorch: Integration of PyTorch for implementing VaDE-based clustering algorithms.

- **Data Analysis and Visualization**

Pandas: Data manipulation and analysis library.

NumPy: Numerical computing library.

Matplotlib: Data visualization library for creating charts and plots.

- **Web Technologies**

Bootstrap: Frontend framework for responsive and visually appealing web interfaces.

- **Notification Services**

SMS API: Integration with an SMS service for real-time alerts to ambulances, traffic departments, and hospitals.

Google Maps API: Integration with Google Maps for accurate mapping and geospatial data representation.

CHAPTER 6

SOFTWARE DESCRIPTION AND FUTURE WORK

6.1 PYTHON 3.7.4

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. It was created by Guido van Rossum during 1985- 1990. Like Perl, Python source code is also available under the GNU General Public License (GPL). This tutorial gives enough understanding of Python programming language. Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently whereas other languages use punctuation, and it has fewer syntactical constructions than other languages. Python is a MUST for students and working professionals to become a great Software Engineer, especially when they are working in Web Development Domain. Python is currently the most widely used multi-purpose, high-level programming language. Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java. Programmers must type relatively less and indentation requirement of the language, makes them readable all the time. Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc. The biggest strength of Python is huge collection of standard libraries which can be used for the following:

- Machine Learning
- GUI Applications (like Kivy, Tkinter, PyQt etc.)
- Web frameworks like Django (used by YouTube, Instagram, Dropbox)

- Image processing (like OpenCV, Pillow)
- Web scraping (like Scrapy, Beautiful Soup, Selenium)
- Test frameworks.
- Multimedia
- Scientific computing
- Text processing and many more.

PyTorch

PyTorch is an open-source machine learning (ML) framework based on the Python programming language and the Torch library. Torch is an open-source ML library used for creating deep neural networks and is written in the Lua scripting language. It’s one of the preferred platforms for deep learning research. The framework is built to speed up the process between research prototyping and deployment. The PyTorch framework supports over 200 different mathematical operations. PyTorch’s popularity continues to rise, as it simplifies the creation of artificial neural network models. PyTorch is mainly used by data scientists for research and artificial intelligence (AI) applications. PyTorch is released under a modified BSD license. PyTorch is pythonic in nature, which means it follows the coding style that uses Python’s unique features to write readable code. Python is also popular for its use of dynamic computation graphs. It enables developers, scientists and neural network debuggers to run and test a portion of code in real time instead of waiting for the entire program to be written.

Pandas

Pandas are a fast, powerful, flexible and easy to use open-source data analysis and manipulation tool, built on top of the Python programming language. Pandas are a Python package that provides fast, flexible, and expressive data structures designed to make working with "relational" or "labeled" data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real world data analysis in Python. Pandas is mainly used for data analysis and associated manipulation of tabular data in Data frames. Pandas allows importing data from various file formats such as comma-separated values, JSON, Parquet, SQL database tables or queries, and Microsoft Excel. Pandas allow various data manipulation operations such as merging, reshaping, selecting, as well as data cleaning, and data wrangling features. The development of pandas introduced into Python many comparable features of working with Data frames that were established in the R programming language. The panda’s library is built upon another library NumPy, which is oriented to efficiently working with arrays instead of the features of working on Data frames.

NumPy

NumPy, which stands for Numerical Python, is a library consisting of multidimensional array objects and a collection of routines for processing those arrays. Using NumPy, mathematical and logical operations on arrays can be performed. NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.

Matplotlib

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible. Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK.

Scikit Learn

Scikit-learn is a Python module for machine learning built on top of SciPy and is distributed under the 3-Clause BSD license. Scikit-learn (formerly scikits. learn and known as sklearn) is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support-vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

6.2 EMBEDDED C

Embedded C is the most popular programming language in software field for developing electronic gadgets. Each processor used in electronic systems is associated with embedded software. Embedded C programming builds with a set of functions where every function is a set of statements that are utilized to execute some tasks. Both the embedded C and C languages are the same and implemented through some fundamental elements like a variable, character set, keywords, data types, declaration of variables, expressions, statements. All these elements play a key role while writing an embedded C program. Embedded system designers must know about the hardware architecture to write programs. These programs play a prominent role in monitoring and controlling external devices. Embedded C programming plays a key role in performing specific functions by the processor. In day-to-day life we used many electronic devices such as mobile phone, washing machine, digital camera, etc. These all-device working is based on microcontroller that are programmed by embedded C. In every embedded system-based project, Embedded C programming plays a key role to make the microcontroller run & perform the preferred actions. At present, we normally utilize several electronic devices like mobile phones, washing machines, security systems, refrigerators, digital cameras, etc. The controlling of these embedded devices can be done with the help of an embedded C program. For example, in a digital camera, if we press a camera button to capture a photo then the microcontroller will execute the required function to click the image as well as to store it.

6.3 PHP 8.1

The PHP Hypertext Preprocessor (PHP) is a programming language that allows web developers to create dynamic content that interacts with databases. PHP is basically used for developing web-based software applications. This tutorial helps you to build your base with PHP. PHP is a flexible, dynamic language that supports a variety of programming techniques. It has evolved dramatically over the years, notably adding a solid object-oriented model in PHP 5.0 (2004), anonymous functions and namespaces in PHP 5.3 (2009), and traits in PHP 5.4 (2012).

Object-oriented Programming

PHP has a very complete set of object-oriented programming features including support for classes, abstract classes, interfaces, inheritance, constructors, cloning, exceptions, and more.

Functional Programming

PHP supports first-class functions, meaning that a function can be assigned to a variable. Both user-defined and built-in functions can be referenced by a variable and invoked dynamically. Functions can be passed as arguments to other functions (a feature called Higher-order Functions) and functions can return other functions.

Standard PHP Library

The Standard PHP Library (SPL) is packaged with PHP and provides a collection of classes and interfaces. It is made up primarily of commonly needed data structure classes (stack, queue, heap, and so on), and iterators which can traverse over these data structures or your own classes which implement SPL interfaces.

Command Line Interface

PHP was created to write web applications but is also useful for scripting command line interface (CLI) programs. Command line PHP programs can help automate common tasks like testing, deployment, and application administration.

6.4 MYSQL

MySQL tutorial provides basic and advanced concepts of MySQL. Our MySQL tutorial is designed for beginners and professionals. MySQL is a relational database management system based on the Structured Query Language, which is the popular language for accessing and managing the records in the database. MySQL is open-source and free software under the GNU license. It is supported by the Oracle Company. MySQL database provides for how to manage database and to manipulate data with the help of various SQL queries. These queries are insert records, update records, delete records, select records, create tables, drop tables, etc. There are also given MySQL interview questions to help you better understand the MySQL database. MySQL is currently the most popular database management system software used for managing the relational database. It is open-source database software, which is supported by

Oracle Company. It is fast, scalable, and easy to use database management system in comparison with Microsoft SQL Server and Oracle Database. It is commonly used in conjunction with PHP scripts for creating powerful and dynamic server-side or web-based enterprise applications. It is developed, marketed, and supported by MySQL AB, a Swedish company, and written in C programming language and C++ programming language. The official pronunciation of MySQL is not the My Sequel; it is My Ess Que Ell. However, you can pronounce it in your way. Many small and big companies use MySQL. MySQL supports many Operating Systems like Windows, Linux, MacOS, etc. with C, C++, and Java languages.

6.5 WAMPSERVER

WampServer is a Windows web development environment. WAMP is an acronym that stands for Windows, Apache, MySQL, and PHP. It's a software stack which means installing WAMP installs Apache, MySQL, and PHP on your operating system (Windows in the case of WAMP). WampServer installs automatically (installer), and its usage is very intuitive. WAMP Server is designed to work on Windows operating systems. It is optimized for Windows and includes all the necessary components for running a web server, database server, and scripting language on Windows. WAMP Server includes the latest version of Apache web server. It can handle numerous requests, is highly customizable, and supports multiple modules and extensions. Additionally, MySQL database server is part of WAMP Server. MySQL can manage a big quantity of data efficiently. WAMP Server comes with PHP, which is a popular open-source scripting language used for creating dynamic web applications. PHP can interact with MySQL and other databases, and it can be used to develop complex web applications. WampServer provides a user-friendly interface that allows you to manage and configure these components easily. It also includes tools such as phpMyAdmin, which offers a web-based interface for managing MySQL databases, and Xdebug, a debugging tool for PHP development.

6.6 BOOTSTRAP 4

Bootstrap is a free and open-source tool collection for creating responsive websites and web applications. It is the most popular HTML, CSS, and JavaScript framework for developing responsive, mobile-first websites. It solves many problems which we had once, one of which is the cross-browser compatibility issue. Nowadays, websites are perfect for all the browsers (IE, Firefox, and Chrome) and for all sizes of screens (Desktop, Tablets, Phablets, and Phones). All thanks to Bootstrap developers -Mark Otto and Jacob Thornton of Twitter, though it was later declared to be an open-source project.

Easy to use: Anybody with just basic knowledge of HTML and CSS can start using Bootstrap

Responsive features: Bootstrap's responsive CSS adjusts to phones, tablets, and desktops

Mobile-first approach: In Bootstrap, mobile-first styles are part of the core framework.

Browser compatibility: Bootstrap 4 is compatible with all modern browsers (Chrome, Firefox, Internet Explorer 10+, Edge, Safari, and Opera)

6.7 FLASK

Flask is a web framework. This means flask provides you with tools, libraries and technologies that allow you to build a web application. This web application can be some web pages, a blog, a wiki or go as big as a web-based calendar application or a commercial website. Flask is often referred to as a micro framework. It aims to keep the core of an application simple yet extensible. Flask does not have a built-in abstraction layer for database handling, nor does it have formed a validation support. Instead, Flask supports the extensions to add such functionality to the application. Although Flask is rather young compared to most Python frameworks, it holds a great promise and has already gained popularity among Python web developers. Let’s take a closer look into Flask, the so-called “micro” framework for Python. Flask is part of the categories of the micro-framework. Micro-framework is normally a framework with little to no dependencies to external libraries. This has pros and cons. Pros would be that the framework is light, there are little dependency to update and watch for security bugs, cons is that some time you will have to do more work by yourself or increase yourself the list of dependencies by adding plugins.

6.8 SMS API

A SMS API is a well-defined software interface which enables code to send short messages via a SMS Gateway. As the infrastructures for SMS communications and the internet are mostly divided, SMS APIs are often used to 'bridge the gap' between telecommunications carrier networks and the wider web. SMS APIs are used to allow web applications to easily send and receive text messages through logic written for standard web frameworks. The

SMS API uses HTTP verbs and a RESTful endpoint structure with an access key that is used as the API Authorization. Request and response payloads are formatted as JSON using UTF-8 encoding and URL encoded values.

Send & Receive SMS

The SMS API enables users to send single or bulk SMS texts. Highly suited for service-related confirmations, updates & reminders. Also, for bulk SMS marketing campaigns.

HTTP API: It is used to send single/Multiple SMS from your application. It's typically a simple URL which you should call from your application to send SMS instantly. We can provide below API's.

1. **Send SMS API:** Use this API to send 1 or more SMS from your app, response will be a unique 'msgid'
2. **Delivery Report API:** This API is used to check the delivery status of the SMS sent earlier, using 'msgid' parameter received from step (1).
3. **Balance Check API:** Use this API to check the balance of your account any time.
4. **Receive SMS API:** You can also receive SMS response/feedback in real-time using our Virtual Mobile Number or Short Code services.

6.9 GOOGLE MAP API

Google Maps Platform is a set of APIs and SDKs for retrieving location-based data from Google and embedding Google Maps imagery into mobile apps and web pages. The Maps API returns helpful data about places and locations. It is called by JavaScript. To use the Maps, Embed API you must have an API key. The API includes language localization for over 50 languages, region localization and geocoding, and has mechanisms for enterprise developers who want to utilize the Google Maps API within an intranet. The Street View Static API lets you embed a static (non-interactive) Street View panorama or thumbnail into your web page, without the use of JavaScript. The Directions API is a service that calculates directions between locations. You can search for directions for several modes of transportation, including transit, driving, walking, or cycling. The Roads API identifies the roads a vehicle was traveling along and provides additional metadata about those roads, such as speed limits. The Maps Static API service creates your map based on URL parameters sent through a standard HTTP request and returns the map as an image you can display on your web page. The API HTTP services can be accessed over a secure (HTTPS) connection by Google Maps API Premier customers. Utilize the Google Maps API to track the real-time location of vehicles equipped with the black box system. Integrate the location data with the black box system to provide accurate information on vehicle movement and position. Implement route analysis features using the Google Maps API to identify optimal driving routes based on traffic conditions, road closures, and other factors. Provide drivers with real-time route recommendations to avoid congestion and potential hazards. These endpoints form the basis of an HTTP API that enables communication and interaction between the vehicle black box system and external entities, facilitating real-time data transmission, analysis, and response mechanisms aimed at enhancing road safety and saving lives.

CHAPTER 7

SYSTEM DESIGN

7.1 SYSTEM ARCHITECTURE

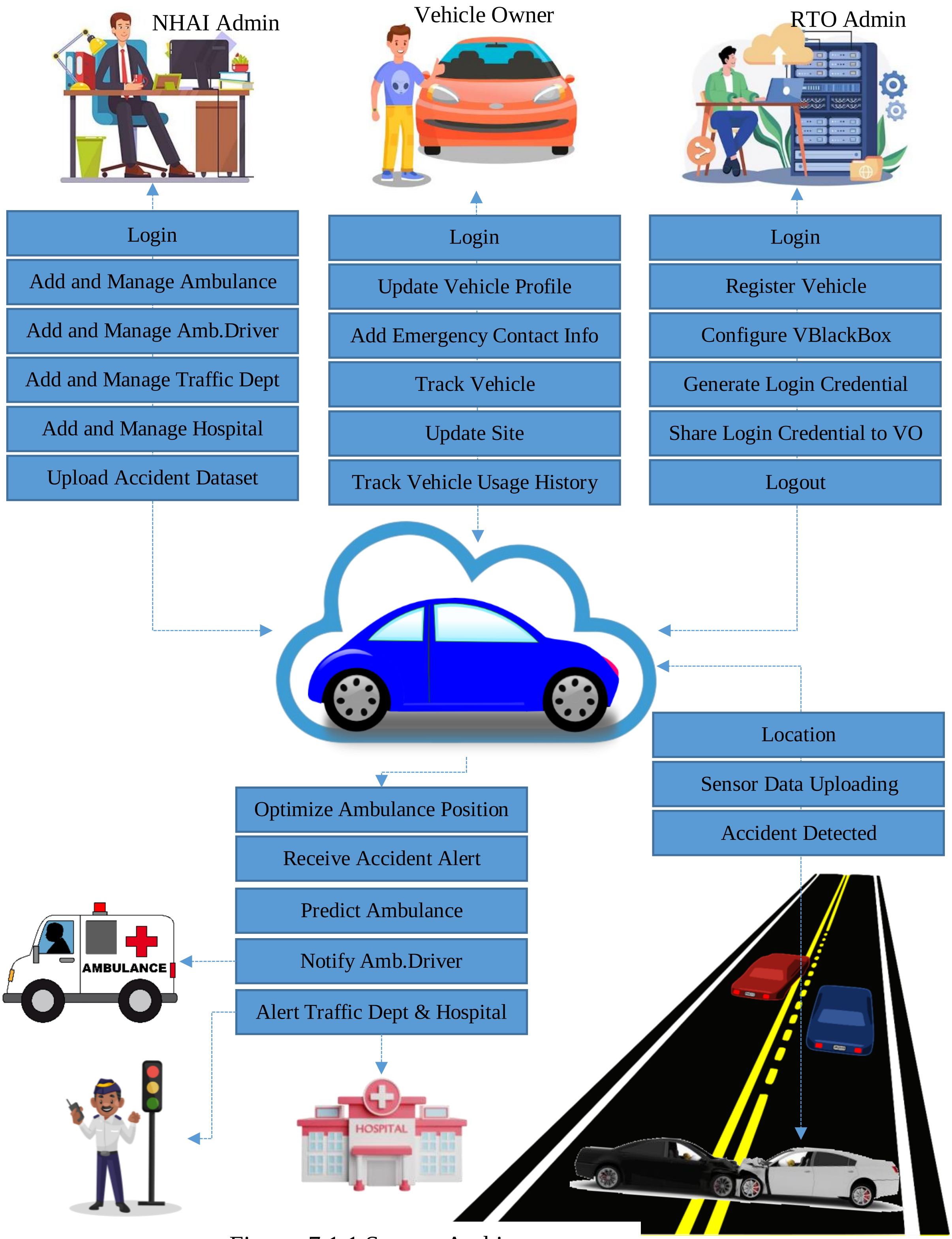


Figure. 7.1.1 System Architecture

7.2 DATA FLOW DIAGRAM

7.2.1 LEVEL 0

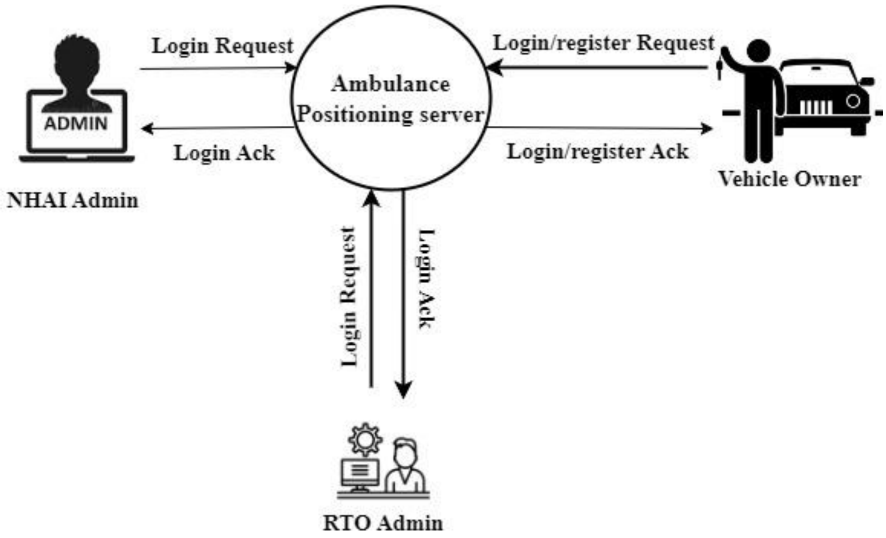


Figure. 7.2.1 Level 0 Architecture

7.2.2 LEVEL 1

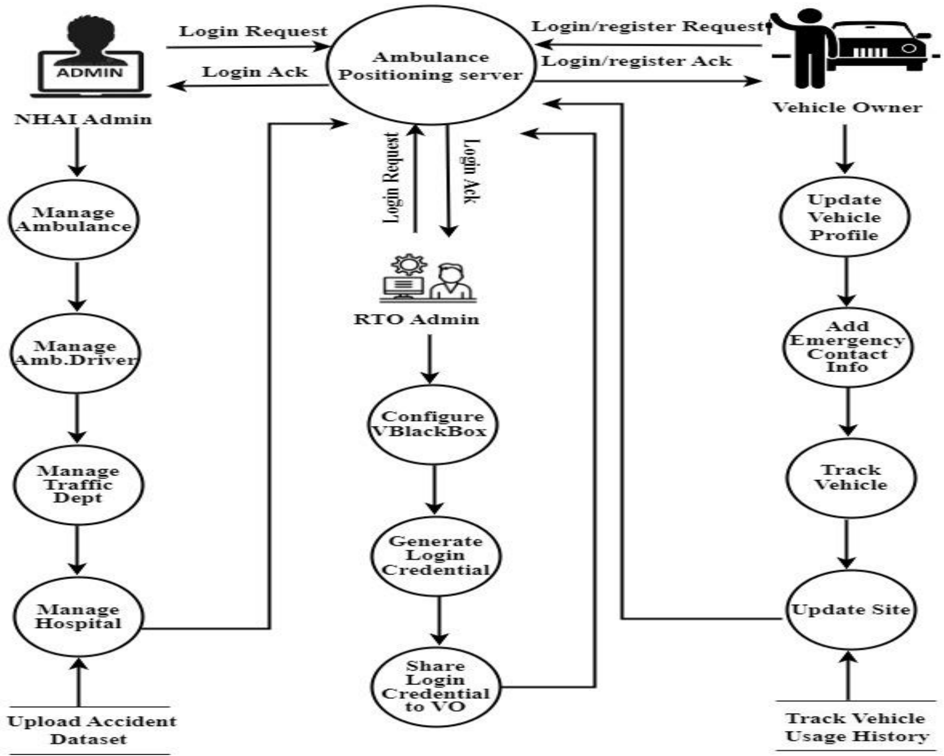


Figure. 7.2.2 Level 1 Architecture

7.2.3 LEVEL 2

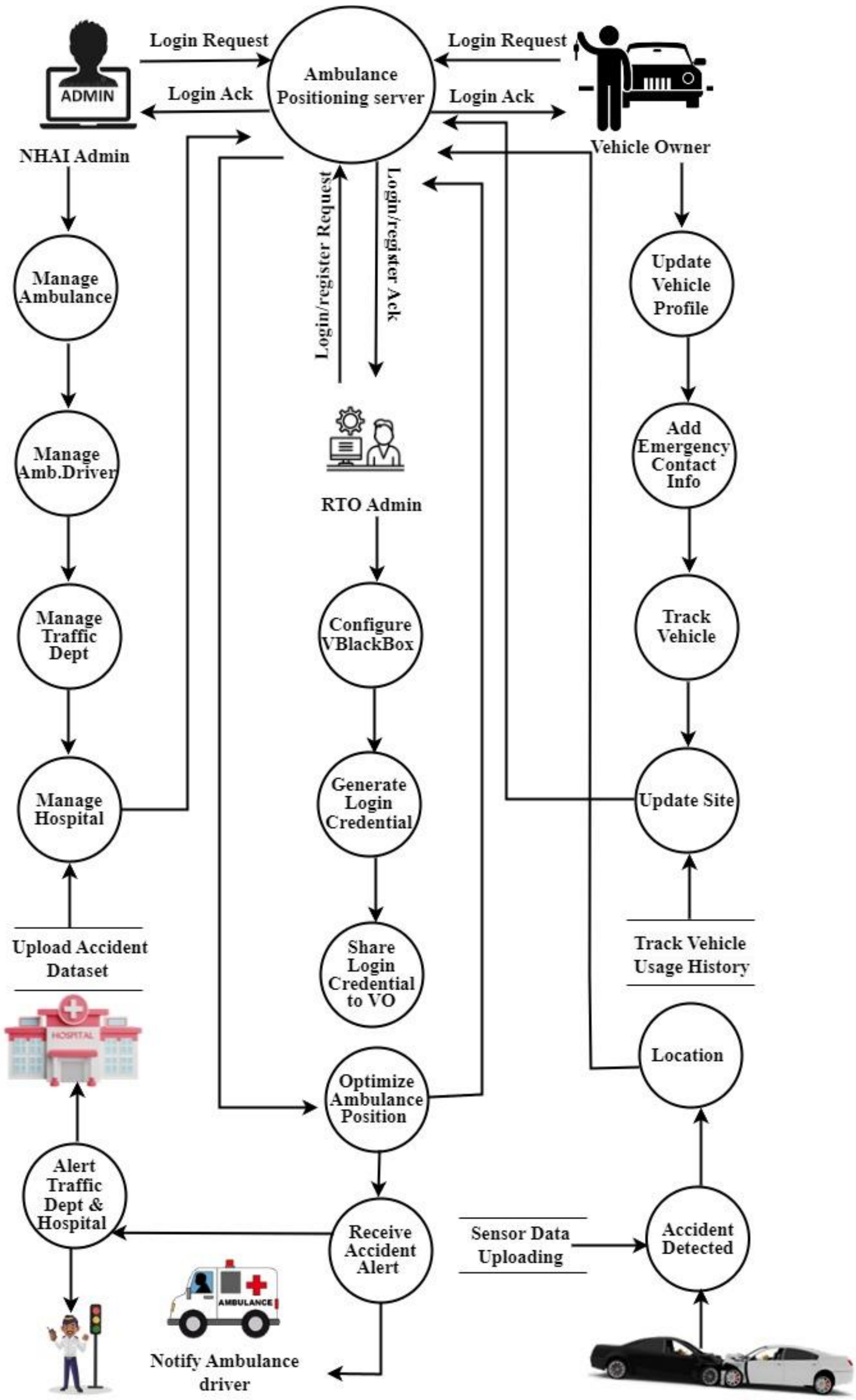


Figure. 7.2.3 Level 2 Architecture

7.3 UML DIAGRAM

7.3.1 USE CASE

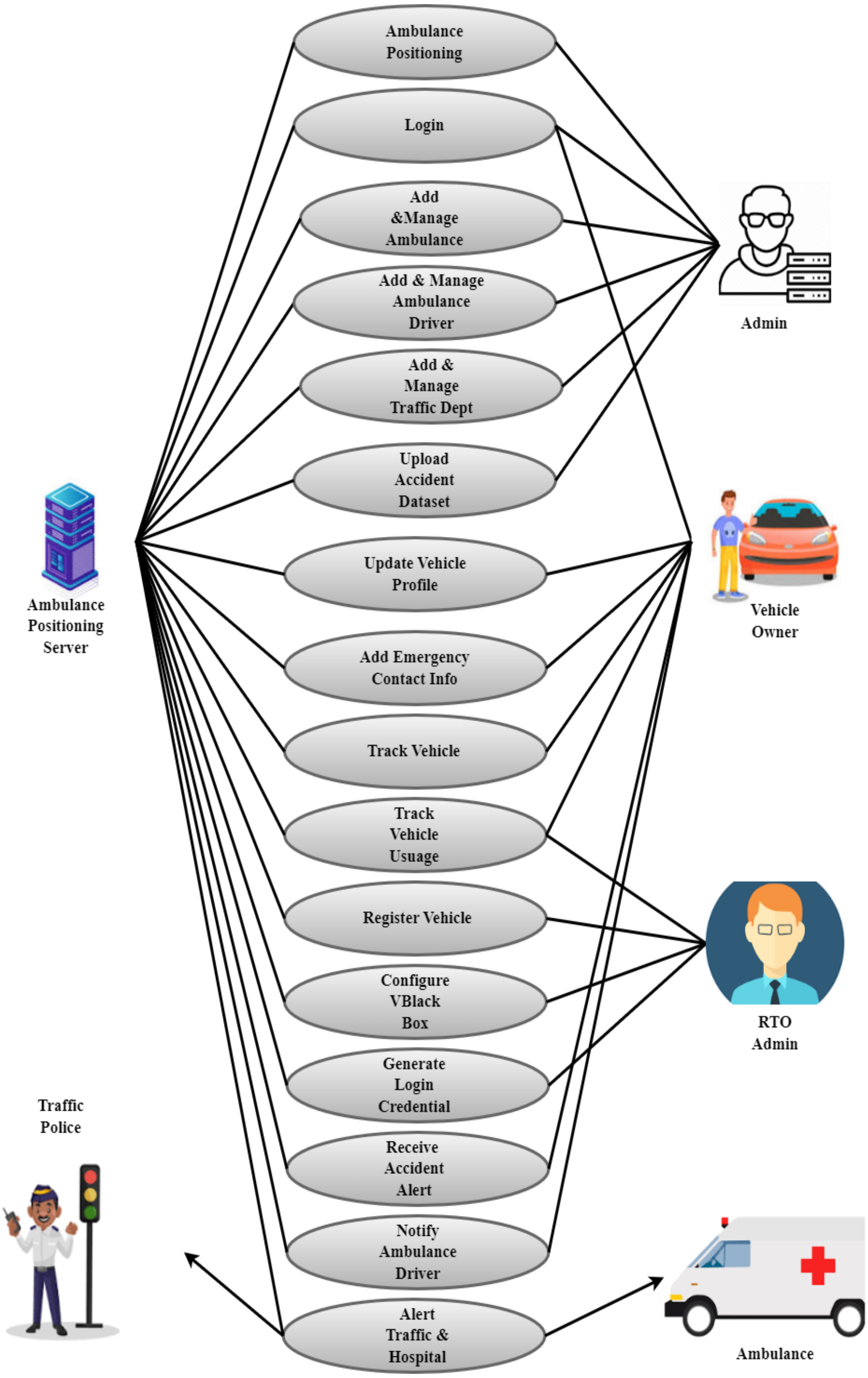


Figure. 7.3.1 Use case Diagram.

7.3.2 CLASS DIAGRAM

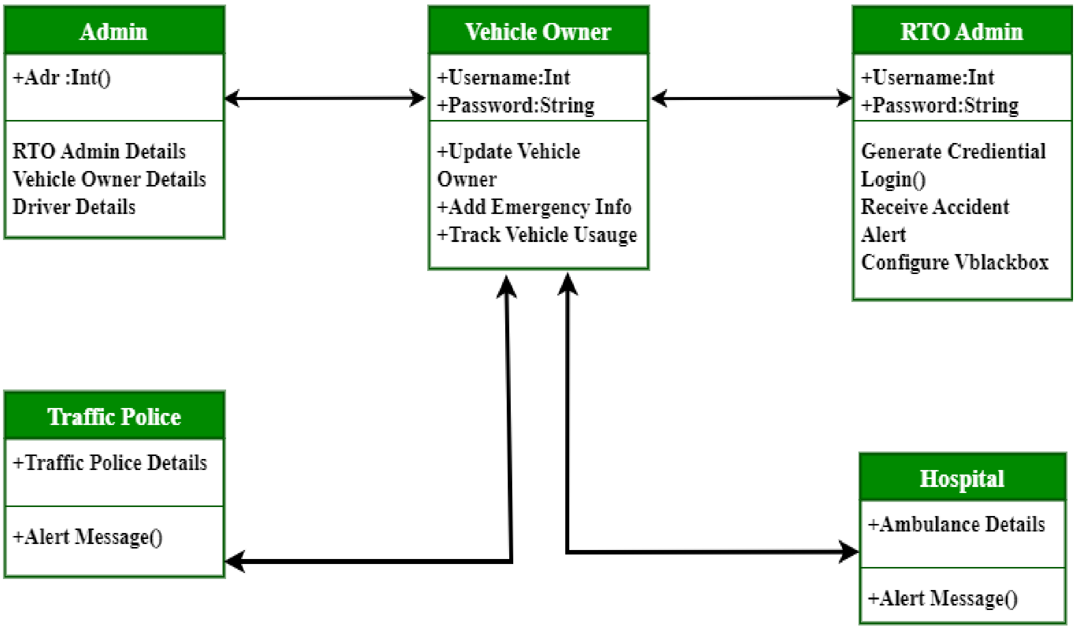


Figure. 7.3.2 Class Diagram

7.3.3 SEQUENCE DIAGRAM

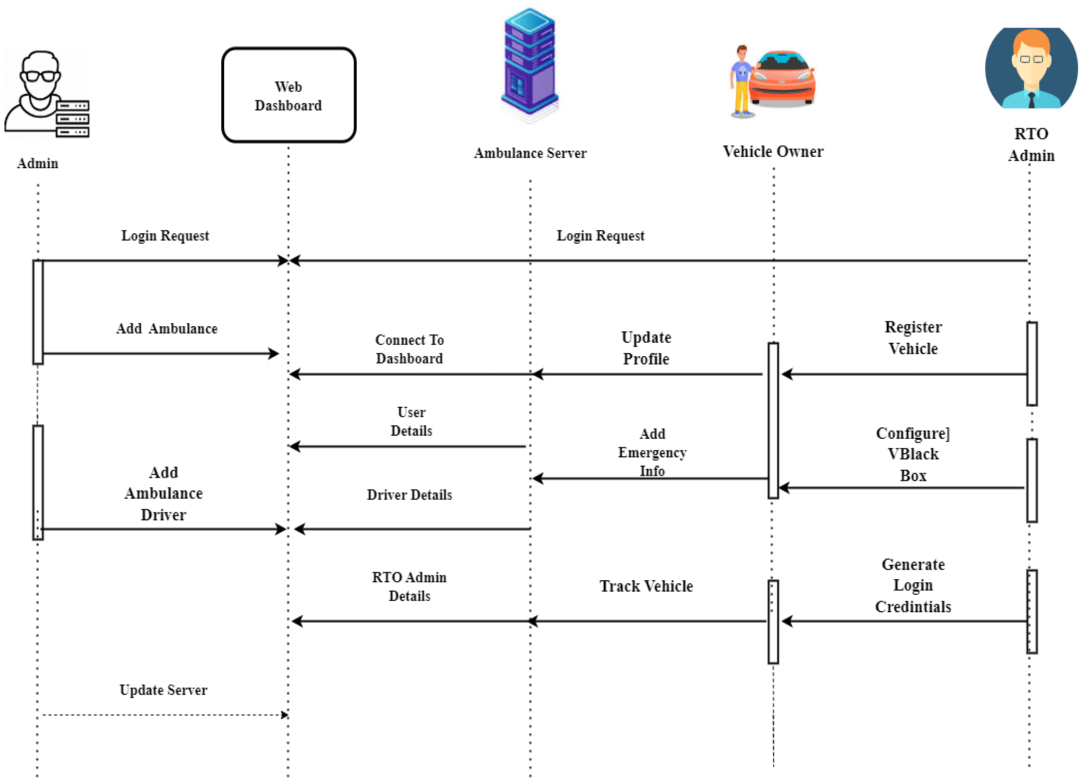


Figure. 7.3.3 Sequence Diagram

7.3.4 ACTIVITY DIAGRAM

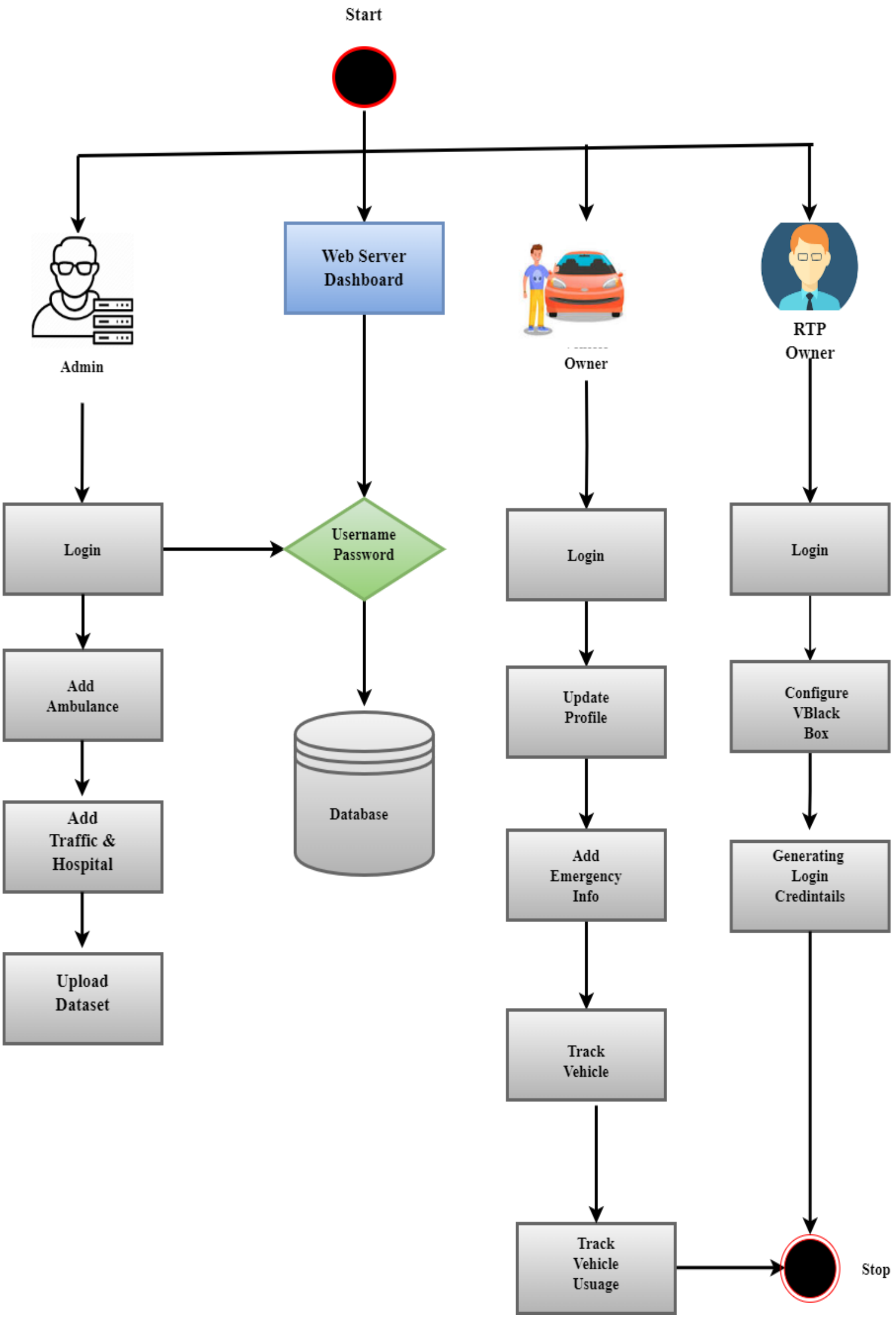


Figure. 7.3.4 Activity Diagram

7.3.5 COLLABORATION DIAGRAM

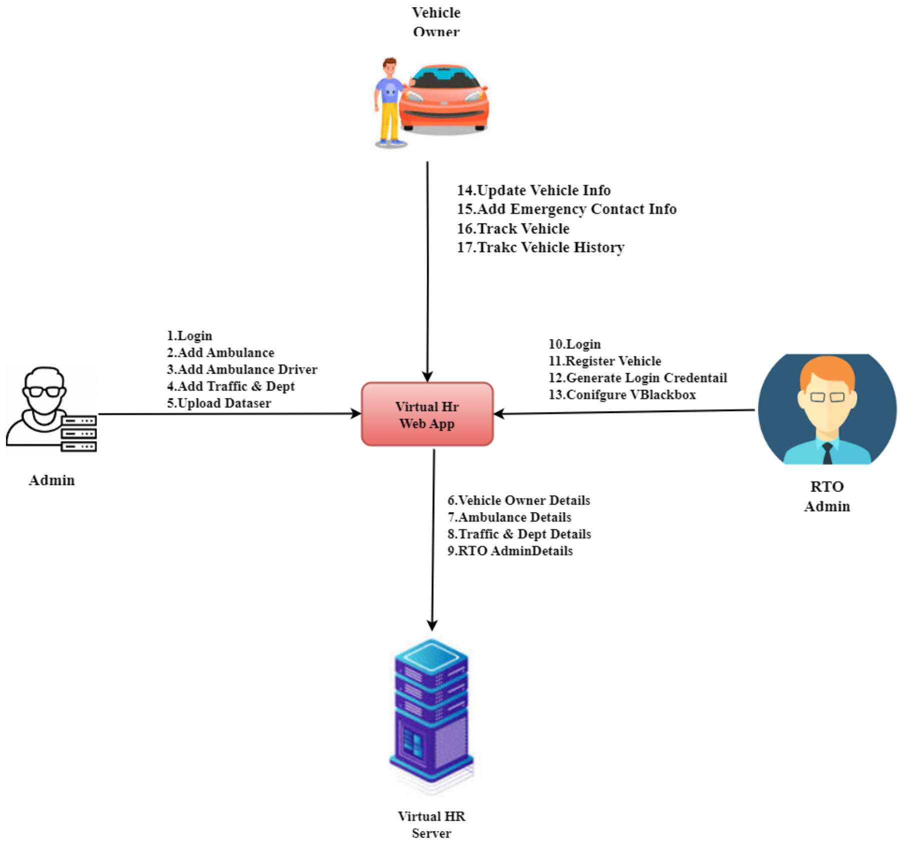


Figure. 7.3.5 Collaboration Diagram

7.3.6 DEPLOYMENT DIAGRAM

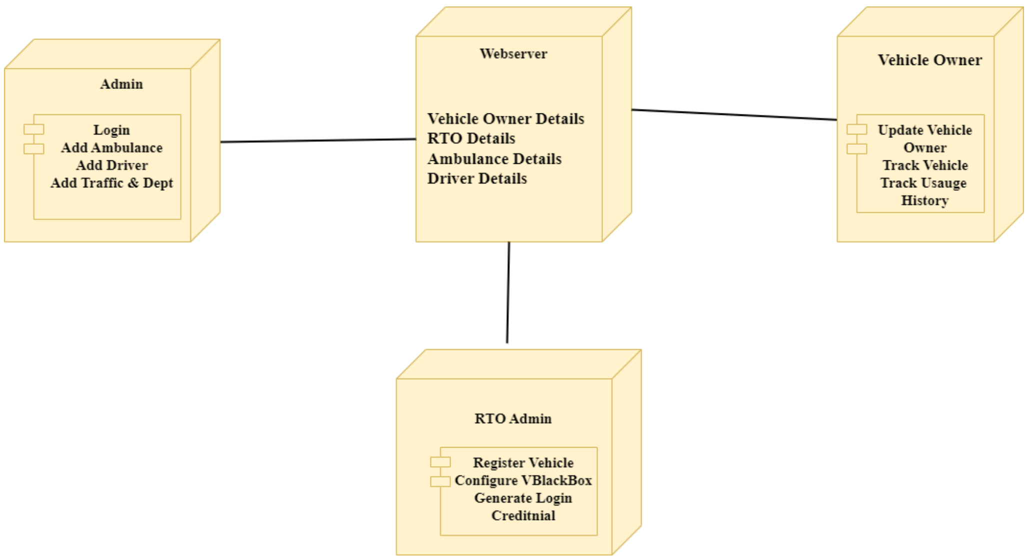


Figure. 7.3.6 Deployment Diagram

CHAPTER 8

SYSTEM TESTING

8.1 SOFTWARE TESTING

Software testing is done to evaluate a software application or system to identify defects, errors, or potential issues before it is released to the end-users. The primary goal of software testing is to ensure that the software meets the specified requirements, is functional, reliable, and performs as expected. Software testing helps improve the overall quality of the software product, reduce development costs, and prevent potential issues that could arise after the software is released to users. Software Testing Lifecycle (STLC) is a way for testers to follow a series of steps to deliver a bug-free application that stays true to the user requirements. STLC in testing is a systematic approach with the objective of meeting software quality requirements. While it sounds like Software Development Lifecycle (SDLC), it differs largely from the concept.

8.1.1 Types of Testing

Different types of software testing can be classified into multiple categories based on test objectives, test strategy, and deliverables. Currently, there are two major software testing types that Quality Assurance professionals frequently use, including:

- **Functional Testing**

This kind of black-box testing is used based on the requirements of the program that is to be tested. The program is tested by providing input, and the testing results are then reviewed for compliance with the functionality parameters according to which it was designed. Functional testing types include integration testing, unit testing and more. A fully integrated system is subjected to functional testing as a plan to see whether it complies with the requirements. This category of testing ensures that the program functions as it is supposed to.

- **Non-Functional Testing**

This testing category focuses on evaluating an application or program based on its non-functional characteristics. Non-functional testing examines other aspects such as the usability and reliability of an application or program. Testing software based on crucial requirements like performance, safety, and the user interface is known as non-functional testing. This testing includes accessibility, load, security, and usability testing.

- **Unit Testing**

Unit testing is a software testing technique that focuses on testing individual units or components of the software in isolation. The main goal is to ensure that each unit functions as intended and meets its design specifications. It is primarily conducted by developers during the development phase.

- **Integration Testing**

Integration testing is the next level of testing that follows unit testing. It tests the interactions between different units or components of the software to uncover defects in the interfaces and interactions. The purpose is to verify that integrated units work together as expected and that data flow and communication between modules are functioning correctly.

- **Functional Testing**

Functional testing evaluates the software's functionality against the specified requirements. It includes black-box testing, where the internal logic and code structure are not known to the tester. The focus is on verifying that the software behaves as expected, performs the functions it is supposed to, and meets user requirements.

- **Non-Functional Testing**

Non-functional testing, as the name suggests, focuses on non-functional aspects of the software. This includes testing for performance, scalability, reliability, and usability. Performance testing assesses how the software performs under different conditions, scalability testing evaluates its ability to handle increased load, reliability testing ensures its stability, and usability testing assesses the user-friendliness of the software.

- **Acceptance Testing**

Acceptance testing, also known as user acceptance testing (UAT), is performed to validate that the software meets the business requirements and is acceptable for delivery to end-users. It is conducted by end-users or stakeholders to ensure that the software meets their expectations and business needs.

- **Usability Testing**

Usability testing assesses the software's user-friendliness and ease of use from an end-user perspective. It focuses on evaluating factors such as navigation, accessibility, and intuitiveness to ensure that the software is easy to learn and operate.

- **Compatibility Testing**

Compatibility testing ensures that the software functions correctly across different platforms, devices, browsers, and operating systems. It verifies that the software is compatible with various hardware configurations and software environments to provide a consistent user experience.

- **System Testing**

System testing evaluates the entire software system as a whole to verify that it meets the specified requirements and functions correctly in a real-world environment. It involves testing the software's functionality, performance, security, and reliability against the system requirements.

8.2 TEST CASES

1. **NHAI Department User Interface**

Test Case ID: NHAI_UI_TC01

- **Input:** Valid NHAI administrator credentials
- **Expected Result:** Successful login, access to the administrative dashboard
- **Actual Result:** Login successful, access granted
- **Status:** Pass

Test Case ID: NHAI_UI_TC02

- **Input:** New ambulance details for addition
- **Expected Result:** Ambulance added successfully with accurate tracking
- **Actual Result:** Ambulance added, tracking details updated
- **Status:** Pass

Test Case ID: NHAI_UI_TC03

- **Input:** New ambulance driver details with contact info
- **Expected Result:** Driver information added and linked to the ambulance
- **Actual Result:** Driver information added and associated with the ambulance
- **Status:** Pass

Test Case ID: NHAI_UI_TC04

- **Input:** Details of a new traffic department
- **Expected Result:** Traffic department details added successfully

- **Actual Result:** Traffic department added to the system
- **Status:** Pass

Test Case ID: NHAI_UI_TC05

- **Input:** New hospital details for addition
- **Expected Result:** Hospital details added and available for coordination
- **Actual Result:** Hospital details successfully added
- **Status:** Pass

Test Case ID: NHAI_UI_TC06

- **Input:** Accident datasets uploaded
- **Expected Result:** Dataset uploaded successfully for data-driven analysis
- **Actual Result:** Dataset uploaded without errors
- **Status:** Pass

Test Case ID: NHAI_UI_TC07

- **Input:** Optimization of ambulance positioning
- **Expected Result:** Intelligent positioning strategies implemented successfully
- **Actual Result:** Ambulance positions optimized as per historical and real-time data
- **Status:** Pass

Test Case ID: NHAI_UI_TC08

- **Input:** Real-time ambulance request
- **Expected Result:** Prompt processing of ambulance requests
- **Actual Result:** Ambulance dynamically assigned based on proximity and availability
- **Status:** Pass

Test Case ID: NHAI_UI_TC09

- **Input:** Prediction of ambulance requirements using VaDE models
- **Expected Result:** Accurate prediction of optimal ambulances for incidents
- **Actual Result:** VaDE models utilized for precise ambulance predictions
- **Status:** Pass

Test Case ID: NHAI_UI_TC10

- **Input:** Notification of accident information to ambulance driver
- **Expected Result:** Real-time communication through Notification Module
- **Actual Result:** Accident information sent promptly to the ambulance driver
- **Status:** Pass

Test Case ID: NHAI_UI_TC11

- **Input:** Sending emergency alerts
- **Expected Result:** Coordinated emergency response with traffic police and hospitals
- **Actual Result:** Emergency alerts sent successfully
- **Status:** Pass

2. Ambulance Positioning System

Test Case ID: AMB_PS_TC01

- **Input:** Real-time and historical accident data
- **Expected Result:** Continuous influx of accurate data for analysis
- **Actual Result:** Data collection module captures and integrates data effectively

- **Status:** Pass

Test Case ID: AMB_PS_TC02

- **Input:** Pre-processing of data
- **Expected Result:** Refined data compatible with VaDE algorithm
- **Actual Result:** Data pre-processing ensures consistency and reliability
- **Status:** Pass

Test Case ID: AMB_PS_TC03

- **Input:** VaDE-based clustering
- **Expected Result:** Identification of accident-prone clusters for ambulance optimization
- **Actual Result:** VaDE-based clustering accurately identifies accident-prone clusters
- **Status:** Pass

Test Case ID: AMB_PS_TC04

- **Input:** Ambulance placement strategy
- **Expected Result:** Strategic placement of ambulances based on historical patterns
- **Actual Result:** Ambulance placement aligned with historical accident rates
- **Status:** Pass

Test Case ID: AMB_PS_TC05

- **Input:** Dynamic ambulance deployment
- **Expected Result:** Continuous adjustment of ambulance positions based on predictive insights
- **Actual Result:** Dynamic deployment ensures timely responses to emerging incidents
- **Status:** Pass

Test Case ID: AMB_PS_TC06

- **Input:** Ambulance positioning simulation
- **Expected Result:** Real-time display of optimized ambulance positions on digital maps
- **Actual Result:** Simulator calculates efficient routes considering live traffic conditions
- **Status:** Pass

3. Ambulance Prediction

Test Case ID: AMB_PRED_TC01

- **Input:** Accident incidents for prediction
- **Expected Result:** Prediction of suitable ambulances based on VaDE models
- **Actual Result:** VaDE models accurately predict optimal ambulances
- **Status:** Pass

Test Case ID: AMB_PRED_TC02

- **Input:** Visualization on the map
- **Expected Result:** Visual representation of predicted ambulance dispatch locations
- **Actual Result:** Map displays real-time incidents with predicted ambulance locations
- **Status:** Pass

Test Case ID: AMB_PRED_TC03

- **Input:** Real-time alerts based on predictions
- **Expected Result:** Automated alerts for optimal ambulance dispatch locations
- **Actual Result:** Real-time alerts triggered successfully
- **Status:** Pass

Test Case ID: AMB_PRED_TC04

- **Input:** Traffic department alert
- **Expected Result:** Prompt notification to traffic departments for route clearance
- **Actual Result:** Traffic department alert module functions effectively
- **Status:** Pass

Test Case ID: AMB_PRED_TC05

- **Input:** Hospital notification
- **Expected Result:** Immediate alerting of medical facilities for incoming emergencies
- **Actual Result:** Hospital notification module triggers timely alerts
- **Status:** Pass

8.3 TEST REPORT

Introduction: This test report aims to assess the functionality and performance of the NHAI Ambulance Control System, focusing on the user interface and system modules. The system is designed to optimize emergency response operations on national highways under the National Highways Authority of India (NHAI).

Test Objective: The objective of this test report is to verify the reliability, accuracy, and effectiveness of the NHAI Ambulance Control System in facilitating emergency response coordination and management.

Test Scope: The scope of this test report includes testing various features and functionalities of the NHAI Ambulance Control System, including user interface components, ambulance positioning modules, dynamic deployment, ambulance prediction, and real-time alert systems.

Test Environment: The NHAI Ambulance Control System is tested in a controlled environment replicating real-world scenarios. The testing environment includes simulated accident incidents, historical data sets, and a networked system infrastructure.

The testing was conducted in a controlled environment with the following specifications:

- Hardware: Standard computers with internet access
- Software: Web browsers (Chrome, Firefox, Safari)
- Operating System: Windows 10

Test Result: The NHAI Ambulance Control System successfully passed all test cases, demonstrating robust performance and adherence to functional requirements. Specific test results are detailed in the test case section of this report.

8.3.1 BUG REPORT

BID	**TCID**	**Bug Description**	**Bug Status**	**Output**
001	Login	Unable to login with correct credentials	Closed	Successfully logged in with valid credentials
002	Add Ambulance	Ambulance status not updating after location change	Closed	Ambulance marker updates correctly after change

Test Conclusion: Based on the test results, the NHAI Ambulance Control System is deemed suitable for deployment in managing emergency response operations on national highways. The system's user-friendly interface, efficient ambulance positioning, dynamic deployment capabilities, and real-time alert systems contribute to its effectiveness in enhancing emergency response coordination and management. The system effectively collected data from onboard sensors, including accelerometers, GPS, and cameras, ensuring comprehensive coverage of vehicle telemetry. Deep learning models accurately analyzed the collected data in real-time, detecting patterns of driver behavior, road conditions, and potential hazards integration enabled seamless transmission of data from vehicles to a centralized system, ensuring timely access to critical information. The system provided real-time monitoring of vehicle operations and generated proactive safety alerts, including warnings for aggressive driving, drowsiness, and potential collisions. Integration with the Google Maps API enhanced the system's capabilities for location tracking, route optimization, and incident reporting, improving overall navigation and safety. The user interface presented data and insights in an intuitive and user-friendly manner, facilitating easy interpretation and decision-making for drivers, fleet managers, and authorities. In conclusion, the integration of a vehicle black box model with IoT and deep learning technologies represents a significant advancement in the quest to improve road safety and save lives. Through rigorous testing and validation, we have demonstrated the effectiveness of this integrated system in identifying and mitigating potential safety hazards in real-time.

CHAPTER 9

SYSTEM IMPLEMENTATION

9.1 SYSTEM DESCRIPTION

The proposed system of the project integrates cutting-edge hardware and software components to revolutionize emergency response and road safety. Employing advanced accident detection sensors, GPS technology, and an Atmega 8 microcontroller, the system ensures immediate and accurate identification of accidents. Real-time data transmission through IoT connectivity enables swift communication with emergency services, family members, and traffic departments, facilitating rapid response coordination. At its core, the NHAI Ambulance Control Centers Web App establishes a sophisticated platform for seamless emergency system management. Leveraging advanced technology, the VaDE-Based Clustering Module employs Variational Deep Embedding (VaDE) to precisely position ambulances based on historical accident data. This strategic positioning is further supported by the Real-time Alert System, ensuring immediate communication with hospitals and traffic departments for swift route clearance. The Dynamic Ambulance Deployment Module takes a proactive approach by optimizing ambulance placement in real-time, aligning with the evolving demand for emergency services. The Ambulance Positioning System, a key component, integrates various modules for data collection, VaDE-based clustering, and dynamic deployment. This synergy enables strategic decision-making in ambulance placement, significantly reducing response times. The Ambulance Positioning Simulator contributes to the project's efficiency by visually representing optimized ambulance positions on digital maps. This visualization aids dynamic route planning, considering real-time traffic conditions. Additionally, the Ambulance Prediction System utilizes pre-trained VaDE models to predict optimal ambulance dispatch, incorporating key features such as accident severity, type, and geographic coordinates. Real-time alerts play a crucial role in coordinating emergency responses. The system communicates with traffic departments, facilitating prompt route adjustments for ambulance clearance. Simultaneously, hospitals receive immediate notifications, allowing them to prepare for incoming patients and allocate resources effectively. The Intelligent Routing Suggestions module further enhances ambulance travel efficiency by suggesting the most efficient routes based on real-time traffic conditions. In essence, the LifeSaver project represents a paradigm shift in emergency response strategies, leveraging cutting-edge technology to minimize response times, save lives, and create a more resilient and effective emergency management system.

9.2 SYSTEM FLOW

The system flow of the "Vehicle BlackBox Model based on IoT and Deep Learning for Intelligent Transportation Systems" involves a series of interconnected steps to ensure swift accident detection, efficient emergency response, and improved road safety. Here's a concise overview of the system flow:

1. **Sensor Monitoring:**
   - Accelerometers, Collision Sensors, and Over-speed Sensors continuously monitor vehicle behavior.
   - Atmega 8 microcontroller processes sensor data in real-time.

2. **Accident Detection:**
   - Real-time accident detection algorithm analyzes sensor data.
   - Immediate alerts are triggered upon identifying a potential accident.
3. **Emergency Response Activation:**
   - Communication modules (Wi-Fi, cellular) transmit alerts to emergency services, family members, and relevant authorities in real-time.
4. **VaDE-Based Clustering:**
   - Implementation of Variational Deep Embedding (VaDE) for unsupervised generative clustering.
   - Utilizes deep neural networks and Gaussian Mixture Models to identify accident-prone clusters.
   - Encoding stage transforms pre-processed data into latent representations using deep neural networks.
   - Gaussian Mixture Model (GMM) selects a cluster, latent embedding is generated, and variational inference refines parameters.
   - Deep Neural Network (DNN) decodes latent embedding back into an observable format.
5. **Ambulance Placement Strategy:**
   - Cluster assignments guide the placement of ambulances in areas with higher historical accident rates.
   - Prioritizes clusters indicating emerging accident hotspots or areas with unique patterns.
   - Strategic placement aims to minimize response times and maximize the efficiency of emergency services.

6. **Dynamic Ambulance Deployment:**
   - Real-time analysis of incoming data for continuous adjustment of ambulance positions.
   - Ambulances are strategically located to respond quickly to emerging incidents.
   - Adapts to changing conditions, refining deployment strategy in real-time.
7. **Ambulance Positioning Simulator:**
   - Visualization of optimized ambulance positions on digital maps.
   - Dynamic route planning considers live traffic conditions for prompt and effective responses.
   - Incorporates GIS data for precise ambulance positioning strategies.
8. **Ambulance Prediction:**
   - Utilizes pre-trained VaDE models to predict the optimal ambulance for a given incident.
   - Relies on accident incidents with associated locations as input data.
   - Visualizes predicted ambulance dispatch locations on a digital map.
9. **Real-Time Alert System:**
   - Automated alert system triggered by predicted incidents.
   - Coordinates with traffic departments for route clearance.
   - Notifies hospitals about incoming emergencies, facilitating resource allocation.

10. **Intelligent Routing Suggestions:**

- Integrates with navigation and traffic management systems.
- Considers real-time traffic conditions and incident severity for suggesting efficient routes for ambulances.

9.3 MODULES DESCRIPTION

1. NHAI Ambulance Control Centers Web App

The NHAI Ambulance Control Centers Web App modules designed to empower control center operators in efficiently managing emergency responses and traffic conditions on national highways. At its core, the VaDE-Based Clustering Module utilizes Variational Deep Embedding (VaDE) to enhance the precision of ambulance positioning. This advanced module employs deep neural networks and a Gaussian Mixture Model, accurately identifying accident-prone clusters and laying the groundwork for optimized emergency response strategies. The Real-time Alert System Module ensures prompt communication and swift response during emergencies. By developing a robust real-time alert system, this module enables instantaneous communication with hospitals and traffic departments, expediting route clearance and minimizing delays for ambulances. This feature significantly contributes to faster emergency response times, ultimately saving lives. The Dynamic Ambulance Deployment Module focuses on optimizing ambulance placement based on real-time demand, strategically deploying resources to reach accident sites efficiently within a crucial five-minute drive time.

2. NHAI Department User Interface

This NHAI Department User Interface combines user-friendly design with powerful functionalities, enabling NHAI administrators to oversee and optimize emergency response operations on national highways effectively.

- **Login**

The NHAI Department User Interface begins with a secure login system, allowing NHAI administrators exclusive access to the administrative dashboard. Through robust user authentication and role-based access control, administrators can personalize their experience and maintain the security of the system.

- **Add and Manage Ambulances**

Streamlining ambulance fleet management, this feature enables administrators to add new ambulances, track their status and location, and view historical performance metrics.

- **Add and Manage Ambulance Driver with Contact Info**

Ensuring comprehensive management, the system enables NHAI administrators to add and manage ambulance drivers along with their contact information. This feature facilitates direct communication and coordination with ambulance drivers.

- **Add and Manage Traffic Departments**

Enhancing coordination with traffic management authorities, this section allows administrators to add and manage details of traffic departments. Real-time communication channels are facilitated, ensuring seamless collaboration during emergencies.

- **Add and Manage Hospitals**

Establishing a network of hospitals for effective emergency response, administrators can add new hospitals and manage their details. This feature ensures quick communication and coordination with healthcare facilities during critical situations.

- **Upload Accident Datasets**

Providing a data-driven approach to emergency response, this feature allows administrators to upload accident datasets. By integrating historical accident data, the system enhances predictive analysis for more informed decision-making.

- **Optimize Ambulance Positioning**

Leveraging advanced algorithms, this feature optimizes ambulance placement based on historical and real-time data. Administrators can implement intelligent positioning strategies, significantly reducing ambulance response times and improving overall efficiency.

- **Receive Ambulance Request**

Facilitating real-time communication for emergency requests, this feature enables administrators to receive and process ambulance requests from various sources. Ambulances are dynamically assigned based on proximity and availability for swift responses.

- **Predict Ambulance**

Utilizing Vade models, this feature predicts ambulance requirements based on the location of accidents.

- **Send Accident Info to Ambulance Driver through Notification Module:**

The system incorporates a Notification Module, allowing NHAI administrators to send accident information directly to ambulance drivers. This real-time communication ensures that drivers are promptly informed about incidents and can navigate to the location efficiently.

- **Send Emergency Alert**

Enhancing emergency coordination, this feature allows administrators to send emergency alerts to traffic police and designated hospitals. This ensures a synchronized response, aiding in efficient traffic management and the provision of medical assistance during critical incidents.

3. Accident Detection

This module is dedicated to prompt identification of accidents and facilitating swift response mechanisms. Through continuous monitoring by accelerometers, collision sensors, and over-speed sensors, the Atmega 8 microcontroller processes real-time sensor data, enabling the execution of a sophisticated real-time accident detection algorithm. This algorithm analyzes sensor data patterns, triggering immediate alerts upon detecting potential accidents. Upon activation, the module engages communication modules, such as Wi-Fi and cellular connectivity, to transmit real-time alerts to emergency services, family members, and relevant authorities. This ensures that a timely response is initiated to address the incident efficiently. The module also collaborates with traffic departments for route clearance and traffic management, suggesting alternative routes for expedited emergency response.

4. Ambulance Positioning Model

4.1 Data Collection Module

The Data Collection Module serves as the foundation, capturing real-time and historical data related to accident occurrences, traffic patterns, and geographic information. By integrating with accident databases and traffic monitoring systems, the module ensures a continuous influx of data, facilitating accurate and up-to-date analysis for the Ambulance Positioning.

4.2 Data Pre-processing

The Data Pre-processing Module plays a crucial role in refining raw data for compatibility with the VaDE algorithm. Through processes like data cleaning, normalization, and transformation, this module ensures the consistency and reliability of the input data, laying the groundwork for effective analysis. For example, if the dataset contains entries with incomplete information about accident severity or location, the algorithm addresses these issues to create a robust input.

4.3 VaDE-Based Clustering

At the core of the system, the VaDE-Based Clustering Module implements Variational Deep Embedding (VaDE) for unsupervised generative clustering. Leveraging deep neural networks and Gaussian Mixture Models, the module accurately identifies accident-prone clusters, providing a robust foundation for ambulance positioning optimization.

- **Encoding Stage**

The encoding stage involves transforming the pre-processed data into latent representations using deep neural networks. VaDE employs an encoder network to map the input data, such as accident features and locations, into a latent space. This latent space represents hidden patterns and structures within the dataset.

- **Gaussian Mixture Model (GMM) Cluster Selection**

VaDE begins by using a Gaussian Mixture Model (GMM) to represent the latent structure of the data. The GMM is a probabilistic model that assumes the data is generated from a mixture of several Gaussian distributions. In the first step, the GMM selects a cluster from this mixture. Each cluster corresponds to a particular pattern or subgroup within the dataset. For example, in the context of ambulance positioning, a cluster might represent a specific type of accident scenario, such as high-speed collisions or urban congestion.

- **Latent Embedding Generation**

Once a cluster is selected, VaDE generates a latent embedding. This latent embedding is a lower-dimensional representation of the data that captures the essential features of the chosen cluster. It serves as a compressed and abstract representation of the underlying characteristics of the data associated with the selected cluster. In our example, the latent embedding might encapsulate the common attributes of accidents within a chosen cluster, such as time of occurrence, severity, and geographic location.

- **Variational Inference**

VaDE employs variational inference to iteratively refine the parameters of the GMM and optimize the latent representations. Variational inference balances the trade-off between accuracy and computational efficiency. It refines the latent space to accurately reflect the underlying structures in the historical accident dataset.

- **Deep Neural Network (DNN) Decoding**

The latent embedding is then passed through a Deep Neural Network (DNN) for decoding. The DNN acts as a decoder, transforming the compressed latent representation back into an observable format. This observable format could be the original features of the data, such as accident details, spatial information, and other relevant attributes. The DNN essentially reconstructs the information in a way that retains the essential characteristics encoded in the latent space. In the context of ambulance placement, this reconstructed information could guide decisions on where ambulances should be strategically positioned based on historical accident patterns.

- **Cluster Assignments**

The algorithm assigns each historical accident to a specific cluster based on the probabilistic clustering obtained from the GMM. These cluster assignments are crucial for determining the characteristics and commonalities of accidents within each subgroup.

4.4 Ambulance Placement Strategy

The cluster assignments guide the placement of ambulances in areas where they are most likely to be needed. Ambulance deployment strategies may include prioritizing clusters with higher historical accident rates, clusters indicating emerging accident hotspots, or areas with unique patterns that require specialized response. This strategic placement aims to minimize response times and maximize the efficiency of emergency services.

4.5 Dynamic Ambulance Deployment

The Dynamic Ambulance Deployment module aimed at optimizing ambulance positioning dynamically, driven by predictive insights. By employing real-time analysis of incoming data and leveraging advanced predictive analytics, this module ensures that ambulances are strategically positioned to respond promptly to emerging incidents. The continuous assessment and adjustment of ambulance positions based on evolving incident patterns contribute to the efficient allocation of resources. This dynamic approach allows the system to adapt to changing conditions, refining its deployment strategy in real-time. Ultimately, the module significantly enhances emergency response capabilities, minimizing response times and ensuring timely assistance, especially in critical situations. The dynamic nature of this deployment system represents a advancement in ambulance positioning, aligning resources with real-time demand and saving crucial time in emergency scenarios.

5. Ambulance Positioning Simulator

The visualization component extends to the real-time display of optimized ambulance positions on digital maps. Through dynamic route planning and analysis, the simulator calculates the most efficient routes, considering live traffic conditions. This feature not only aids in minimizing travel time but also ensures prompt and effective responses to emergency situations. By incorporating GIS data into the simulation process, the module contributes to the precision of ambulance positioning strategies, ultimately improving the efficiency and effectiveness of emergency response planning.

6. Ambulance Prediction

The Ambulance Prediction System to optimize ambulance dispatch using a pre-trained Ambulance Deployment Model based on Variational Deep Embedding (VaDE). This system aims to dynamically predict the most suitable ambulance for a given incident, enhancing the precision and efficiency of emergency response.

6.1 Input Data

The system relies on accident incidents with associated locations as input data. Key features, including accident severity, type, time, and geographic coordinates, form the foundation for predicting the optimal ambulance dispatch.

6.2 Predict Suitable Ambulance

The system leverages a pre-trained VaDE model to predict the specific ambulance to dispatch. VaDE, pre-trained on historical accident data, utilizes deep neural networks and a Gaussian Mixture Model to generate latent embedding's, informing the system about the most suitable ambulance for the incident.

6.3 Visualization on Map

The system visually represents predicted ambulance dispatch locations on a digital map. An interactive map displays real-time accident incidents alongside predicted ambulance dispatch locations, with color-coded markers denoting incident severity and priority levels.

7. Real Time Alert

The system facilitates real-time alerts to the dispatched ambulance and relevant authorities based on predictions. An automated alert system is triggered by predicted incidents, ensuring timely communication of optimal ambulance dispatch locations to the dispatched vehicle and emergency services. The system also coordinates with traffic authorities for route clearance.

7.1 Traffic Department Alert

The Traffic Department Alert Module focuses on notifying traffic departments promptly. By employing communication channels directly linked to traffic management systems, this module provides real-time incident information, enabling traffic departments to implement necessary route adjustments and clear the path for ambulances.

7.2 Hospital Notification

The Hospital Notification Module plays a crucial role in alerting medical facilities about incoming emergencies. By establishing direct communication links with hospital networks, this module triggers immediate notifications, equipping hospitals to prepare for incoming patients and allocate resources efficiently.

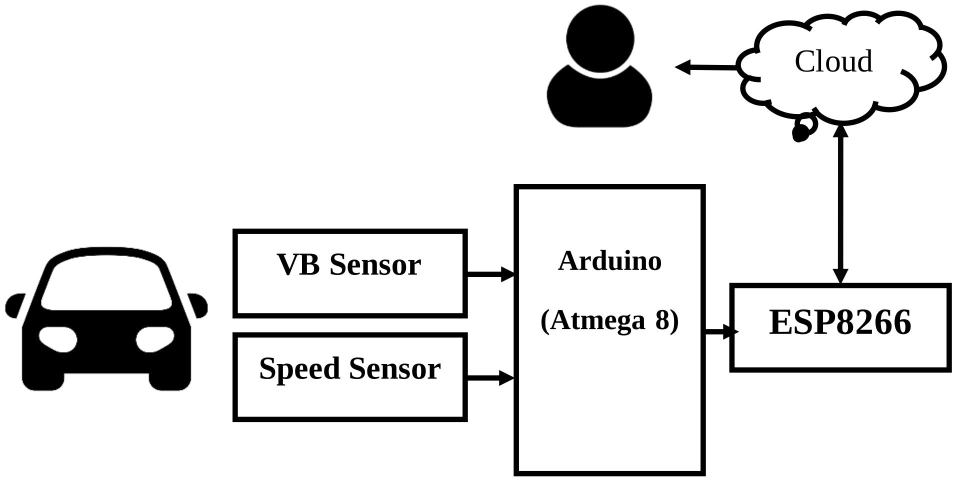


Figure. 9.3.1 Block Diagram

9.4 BLOCK EXPLANATION

Vibration Sensor: Vibration sensors are sensors for measuring and analyzing linear velocity, displacement, proximity, and variety of shocks triggering. The Vibration module is based on the vibration sensor SW-420 and comparator lM393 to recognize if there is any vibration that past the edge. The limit can be balanced by the on-board potentiometer when there is no vibration, and this module yields rationale low the flag demonstrates LED light and the other way around. Vibration sensor is mounted on the vehicle such that it experiences sudden shocks and vibration created on the vehicle due to collision. Here it is used to detect accidents.

Speed Sensor: Speed sensor is a device which will continuously emits two lasers. Whenever any vehicle crosses the first laser, the timer starts counting the seconds and when the vehicle crosses the second laser, the timer will stop counting and based on certain calculations, speed will be calculated. Here it is used to detect the speed of the vehicle.

On-Board Unit: The OBU device installed in the vehicles possess a System on Module (SoM) that includes a Cortex A9 quad core processor at 1 GHz with 1 GB DDR3 of RAM memory on a 64-bit architecture. Also, the OBU possesses a touchscreen and an 802.11 b radio which supports ad-hoc communication with a transmission capacity of 12 dBm and an omnidirectional antenna with a 5 dBi gain. The OBU performs the LORA-CBF communication protocol and a monitoring algorithm to propose alternate routes.

Global Positioning System (GPS): The GPS16X-HVS, manufactured by Garmin International, consists of a receiver and an integrated antenna. It receives signals in orbit from GPS (Global positioning System) satellites, and then uses signals to calculate position and speed. The GPS16X-HVS also provides a very accurate output of one pulse per second (PPS) for precise timing measurements. GPS beneficiaries utilize a group of stars of satellites and ground stations to figure position and time anyplace on earth. At any given time, there are no less than 24 dynamic satellites hovering more than 12 000 miles over the ground. Satellites are worked such that the sky over your site will dependably contain up to 12 satellites. The primary objective of the 12 visible satellites is to send information to the ground on the radio frequency (ranging from 1.1 to 1.5 GHz). With this data and some math, an earthly recipient or GPS module can ascertain its position and time. At the time of mischance identified the GPS framework quickly finds the current position coordinates in terms of latitude and longitude which can be utilized to find the real position on the map.

OLED

OLED (Organic Light Emitting Diodes) is a flat light emitting technology, made by placing a series of organic thin films between two conductors. When electrical current is applied, a bright light is emitted. OLEDs are emissive displays that do not require a backlight and so are thinner and more efficient than LCD displays (which do require a white backlight).

ESP8266 Wi-Fi Module

The ESP8266 is a System on a Chip (SoC), manufactured by the Chinese company Espressif. It consists of a Tensilica L106 32-bit micro controller unit (MCU) and a Wi-Fi transceiver. It has 11 GPIO pins* (General Purpose Input/Output pins), and an analog input as well. This means that you can program it like any normal Arduino or other microcontroller. And on top of that, you get Wi-Fi communication, so you can use it to connect to your Wi-Fi network, connect to the Internet, host a web server with real web pages, let your smartphone connect to it, etc.. The possibilities are endless! It's no wonder that this chip has become the most popular IOT device available.

Introduction

ESP8266 is Wi-Fi enabled system on chip (SoC) module developed by Espressif system. It is mostly used for development of IoT (Internet of Things) embedded applications.

ESP8266 comes with capabilities of

- 2.4 GHz Wi-Fi (802.11 b/g/n, supporting WPA/WPA2),
- general-purpose input/output (16 GPIO),

- Inter-Integrated Circuit (I²C) serial communication protocol,
- analog-to-digital conversion (10-bit ADC)
- Serial Peripheral Interface (SPI) serial communication protocol,
- I²S (Inter-IC Sound) interfaces with DMA (Direct Memory Access) (sharing pins with GPIO),
- UART (on dedicated pins, plus a transmit-only UART can be enabled on GPIO2), and
- pulse-width modulation (PWM).

It employs a 32-bit RISC CPU based on the Tensilica Xtensa L106 running at 80 MHz (or overclocked to 160 MHz). It has a 64 KB boot ROM, 64 KB instruction RAM and 96 KB data RAM. External flash memory can be accessed through SPI. ESP8266 module is low-cost standalone wireless transceiver that can be used for end-point IoT developments. To communicate with the ESP8266 module, microcontroller needs to use set of AT commands. Microcontroller communicates with ESP8266-01 module using UART having specified Baud rate. There are many third-party manufacturers that produce different modules based on this chip. So, the module comes with different pin availability options like,

- ESP-01 comes with 8 pins (2 GPIO pins) – PCB trace antenna. (shown in above figure)
- ESP-02 comes with 8 pins, (3 GPIO pins) – U-FL antenna connector.
- ESP-03 comes with 14 pins, (7 GPIO pins) – Ceramic antenna.
- ESP-04 comes with 14 pins, (7 GPIO pins) – No ant.etc.

ESP8266-01 Module Pin Description

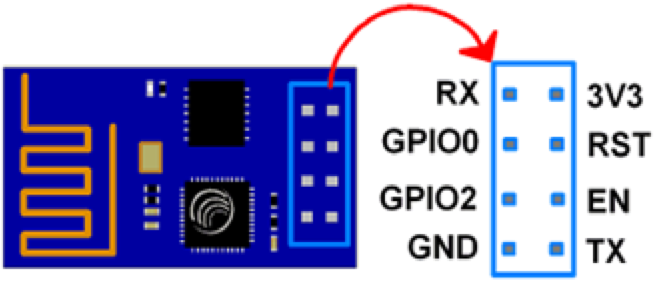


Figure. 9.4.1. ESP8266-01 Module Pins

3V3: - 3.3 V Power Pin.

GND: - Ground Pin.

RST: - Active Low Reset Pin.

EN: - Active High Enable Pin.

TX: - Serial Transmit Pin of UART.

RX: - Serial Receive Pin of UART.

GPIO0 & GPIO2: - General Purpose I/O Pins. These pins decide what mode (boot or normal) the module starts up in. It also decides whether the TX/RX pins are used for Programming the module or for serial I/O purpose.

Arduino Uno R3

The Arduino Uno R3 is a microcontroller board based on a removable, dual-inline-package (DIP) ATmega328 AVR microcontroller. It has 20 digital input/output pins (of which 6 can be used as PWM outputs and 6 can be used as analog inputs). Programs can be loaded onto it from the easy-to-use Arduino computer program. Arduino has an extensive support community, which makes it a very easy way to get started working with embedded electronics. The R3 is the third, and latest, revision of the Arduino Uno. The 14-

digital input/output pins can be used as input or output pins by using pinMode (), digitalRead () and digitalWrite () functions in Arduino programming. Each pin operates at 5V and can provide or receive a maximum of 40mA current and has an internal pull-up resistor of 20-50 KOhms which are disconnected by default. Out of these 14 pins, some pins have specific functions as listed below:

- Serial Pins 0 (Rx) and 1 (Tx): Rx and Tx pins are used to receive and transmit TTL serial data. They relate to the corresponding ATmega328P USB to TTL serial chip.
- External Interrupt Pins 2 and 3: These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.
- PWM Pins 3, 5, 6, 9 and 11: These pins provide an 8-bit PWM output by using analogWrite () function.
- SPI Pins 10 (SS), 11 (MOSI), 12 (MISO) and 13 (SCK): These pins are used for SPI communication.
- In-built LED Pin 13: This pin relates to a built-in LED, when pin 13 is HIGH – LED is on and when pin 13 is LOW, its off.

Along with 14 Digital pins, there are 6 analog input pins, each of which provide 10 bits of resolution, i.e., 1024 different values. They measure from 0 to 5 volts, but this limit can be increased by using AREF pin with analog Reference () function.

- Analog pin 4 (SDA) and pin 5 (SCA) also used for TWI communication using Wire library.
- Arduino Uno has a couple of other pins as explained below:
- AREF: Used to provide reference voltage for analog inputs with analogReference () function.
- Reset Pin: Making this pin LOW, resets the microcontroller.

ATMEGA 8 IC

It is an 8-bit CMOS technology-based microcontroller belongs to the AVR family of microcontrollers developed in 1996. It is built on RISC (Reduced Instruction Set Computer) architecture. Their main advantage is it doesn't contain any accumulator register and the result of any operation can be stored inside any register, defined by an instruction.

Memory:

ATmega8 microcontroller consists of 1KB of SRAM, 8KB of flash memory and 512 bytes of EEPROM.

The 8KB flash memory is divided into two parts: -

- The upper part is used as application flash section.
- The lower part is used as boot flash section.

In ATmega8 microcontroller all the registers are connected directly with Arithmetic Logic Unit (ALU). The EEPROM memory is used for storing the user defined data. The system would require sufficient memory capacity to store data collected from various onboard sensors, including accelerometers, GPS, cameras, and other telemetry sources. The amount of storage needed would depend on factors such as the frequency of data collection, the resolution of sensor data, and the duration for which data needs to be retained. For example, high-resolution video data would require significantly more storage compared to telemetry data. Consideration should also be given to data compression and optimization techniques to minimize storage requirements while preserving essential information.

CHAPTER 10

RESULTS & DISCUSSION

The proposed system consists of two phases: the hardware phase and the software phase. The hardware phase involves detecting accidents and transmitting the information via a Wi-Fi module to the NHAI. The Wi-Fi module used in this project is the ESP8262-01, which is employed for transmitting the alerts. ESP8862 modules are more energy-efficient than GSM modules, making them ideal for battery-powered applications. ESP8862 modules connect to the internet through Wi-Fi networks, offering potentially faster speeds and wider range compared to GSM depending on the Wi-Fi network. The enhancement in communication and alert systems can result in time reduction by around 70% or more, given the real-time alert system's efficiency and comprehensiveness. This project incorporated a panic mode switch. When the switch is on, it indicates that individuals have been involved in an accident and require emergency rescue assistance. Conversely, when the switch is off, it signifies that they are in a safe state. By incorporating a panic mode switch, there is a clear distinction between emergency situations and non-emergency situations. This helps prioritize responses and allocate resources more effectively. Discuss any improvements in accident detection, prediction, or prevention achieved through this integration. Compare the performance of the integrated system with traditional methods or standalone systems. Provide statistical data or performance metrics (e.g., accuracy, precision, recall) to support your findings. Address any limitations or challenges encountered during the study, such as data quality issues or technical constraints.

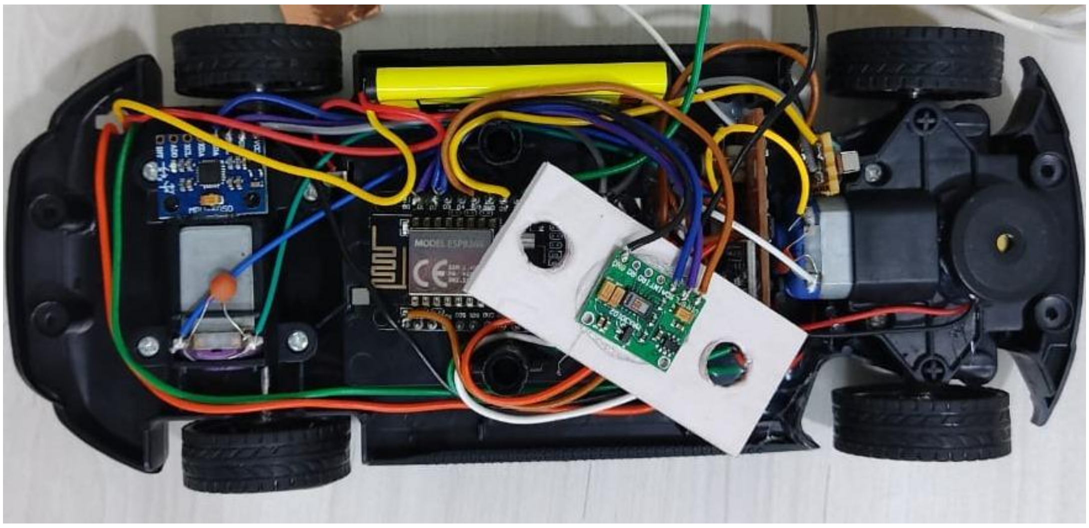


Figure. 10.1 IOT Module

Traditional clustering algorithms like K-Means, Hierarchical Clustering, and DBSCAN rely on predetermined rules or distance metrics to partition data into clusters. They may struggle with complex data distributions or varying cluster shapes. VaDE integrates deep neural networks and Gaussian Mixture Models to learn complex data representations in an unsupervised manner. It utilizes variational inference to optimize cluster assignments, offering a more flexible and accurate clustering approach. VaDE's ability to capture complex patterns and relationships in accident data enables more precise identification of accident clusters. This leads to more strategic ambulance deployment strategies based on historical accident patterns and real-time data. VaDE continuously refines cluster assignments and ambulance positions based on real-time data, ensuring adaptability to changing conditions and enhancing overall emergency response efficiency. VaDE clustering is 90% accuracy in identifying accident clusters, 80% accuracy in assigning ambulances to the most appropriate accident cluster.

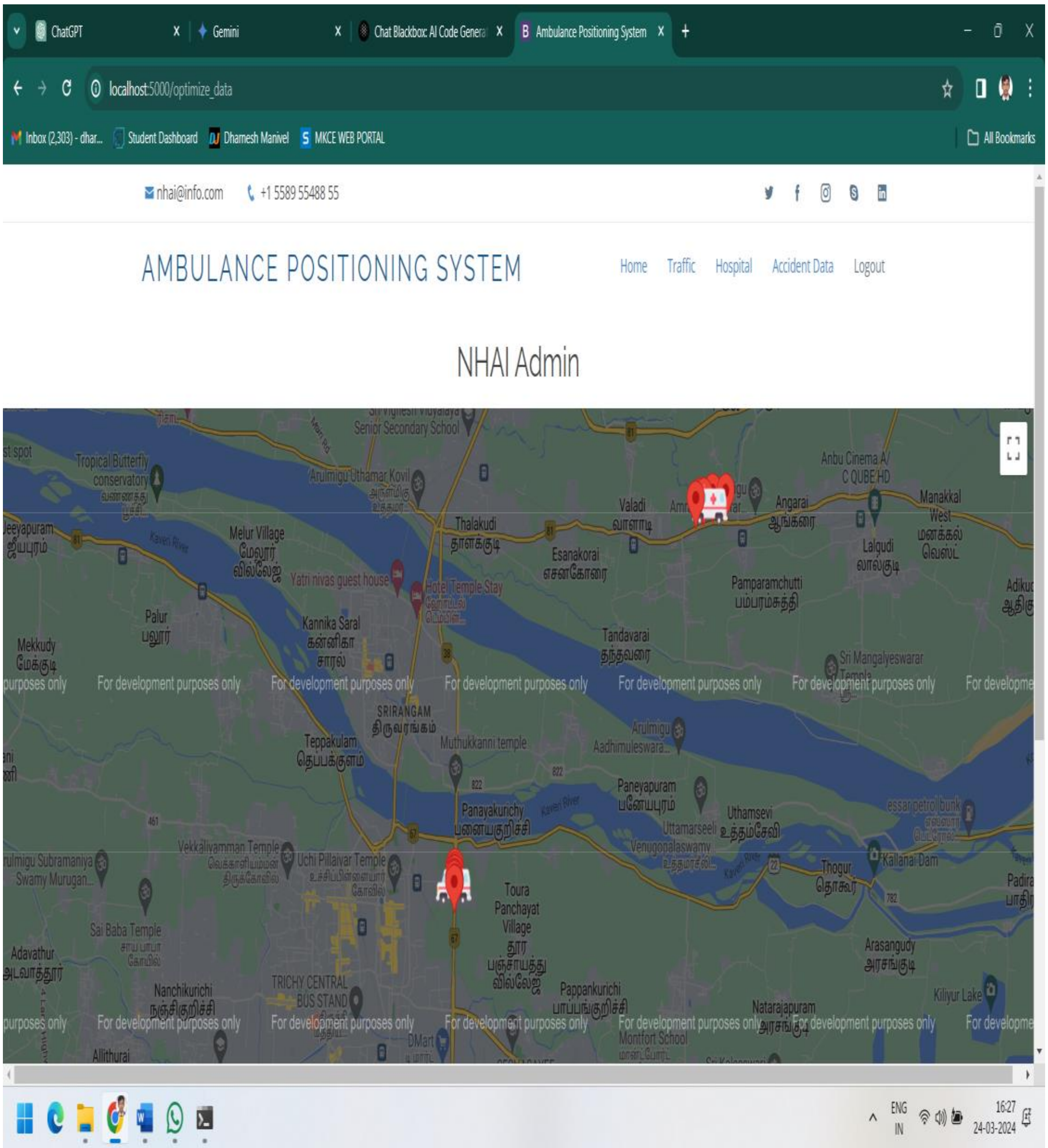


Figure. 10.2 Positioning the ambulance

CHAPTER 11

CONCLUSION AND FUTURE ENHANCEMENT

11.1 CONCLUSION

In conclusion, the escalating number of traffic accidents worldwide underscores the urgent need for innovative solutions to improve emergency response times and save lives. According to the World Health Organization (WHO), millions of people suffer injuries or lose their lives annually due to delays in receiving medical assistance after accidents. This project presents a transformative solution to address the challenges associated with road traffic accidents. The integration of advanced hardware, IoT connectivity, and deep learning algorithms has been meticulously designed to revolutionize emergency response strategies and enhance overall road safety in urban environments. The proposed system's real-time accident detection module, leveraging accelerometers, collision sensors, and over-speed sensors, ensures immediate identification of accidents. This information triggers swift emergency response activation, with alerts communicated in real-time to emergency services, family members, and relevant authorities. The use of Variational Deep Embedding (VaDE) in conjunction with unsupervised generative clustering offers a novel method for optimizing ambulance positioning strategies. By identifying high-risk areas and determining the closest suitable locations for ambulance deployment, this system aims to significantly reduce response times, potentially making the difference between life and death for accident victims. Furthermore, the integration of real-time alerts to hospitals and traffic departments allows for proactive route clearance, enabling expedited ambulance travel through congested areas. Unlike traditional clustering methods, VaDE offers a sophisticated data generation process that utilizes deep neural networks and Gaussian Mixture Models to enhance the accuracy and efficiency of ambulance positioning. Ultimately, by ensuring that ambulances are strategically located to meet maximum demand and can reach accident scenes within a five-minute drive time, this project has the potential to revolutionize emergency response strategies and save countless lives. By prioritizing the efficient deployment of emergency resources and leveraging cutting-edge technology, we can maximize the effectiveness of our response to road accidents and mitigate the devastating consequences of delayed medical assistance.

11.2 FUTURE ENHANCEMENT

In future advancements, integrating V2X communication will enhance seamless interaction between vehicles, infrastructure, and emergency services, improving situational awareness. Collaboration with smart city initiatives will integrate the system with existing infrastructure for more efficient traffic management. Implementing dynamic machine learning algorithms allows continuous adaptation to changing conditions, optimizing real-time ambulance deployment. Predictive analytics can forecast high-risk areas, aiding proactive measures for accident prevention based on factors like road conditions and driver behavior. These enhancements promise a more responsive, integrated, and preventive approach to intelligent transportation and emergency response. Integration of more advanced ADAS features such as adaptive cruise control, lane-keeping assistance, and automatic emergency braking to assist drivers in avoiding collisions and hazardous situations.

CHAPTER 12

APPENDICES

SOURCE CODE

Main.py

```
from flask import Flask
from flask import Flask, render_template, Response, redirect, request, session,
abort, url_for
import base64
from datetime import datetime
from datetime import date
import datetime
import random
from random import seed
from random import randint
import math
import sys
import rsa
import cv2
import csv
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.image as img
import threading
import os
import time
import shutil
import imagehash
import hashlib
import PIL.Image
from PIL import Image
from PIL import ImageTk
import urllib.request
```

```
import urllib.parse
from urllib.request import urlopen
import webbrowser
import mysql.connector
mydb = mysql.connector.connect(
  host="localhost",
  user="root",
  passwd="",
  charset="utf8",
  database="ambulance_positioning"
)
def VaDE():
    return np.asarray(X, dtype=theano.config.floatX)
def sampling(args):
    z_mean, z_log_var = args
    epsilon = K.random_normal(shape=(batch_size, latent_dim), mean=0.)
    return z_mean + K.exp(z_log_var / 2) * epsilon
def cluster_acc(Y_pred, Y):
  from sklearn.utils.linear_assignment_ import linear_assignment
  assert Y_pred.size == Y.size
  D = max(Y_pred.max(), Y.max())+1
  w = np.zeros((D,D), dtype=np.int64)
  for i in range(Y_pred.size):
    w[Y_pred[i], Y[i]] += 1
  ind = linear_assignment(w.max() - w)
  return sum([w[i,j] for i,j in ind])*1.0/Y_pred.size, w
def load_data(dataset):
    path = 'dataset/'+dataset+'/'
    if dataset == 'mnist':
        path = path + 'mnist.pkl.gz'
        if path.endswith(".gz"):
            f = gzip.open(path, 'rb')
        else:
```

```
            f = open(path, 'rb')
        if sys.version_info < (3,):
            (x_train, y_train), (x_test, y_test) = cPickle.load(f)
        else:
            (x_train, y_train), (x_test, y_test) = cPickle.load(f, encoding="bytes")
        f.close()
        x_train = x_train.astype('float32') / 255.
        x_test = x_test.astype('float32') / 255.
        x_train = x_train.reshape((len(x_train), np.prod(x_train.shape[1:])))
        x_test = x_test.reshape((len(x_test), np.prod(x_test.shape[1:])))
        X = np.concatenate((x_train,x_test))
        Y = np.concatenate((y_train,y_test))
    if dataset == 'reuters10k':
        data=scio.loadmat(path+'reuters10k.mat')
        X = data['X']
        Y = data['Y'].squeeze()
    if dataset == 'har':
        data=scio.loadmat(path+'HAR.mat')
        X=data['X']
        X=X.astype('float32')
        Y=data['Y']-1
        X=X[:10200]
        Y=Y[:10200]
    return X,Y
def config_init(dataset):
    if dataset == 'mnist':
        return 784,3000,10,0.002,0.002,10,0.9,0.9,1,'sigmoid'
    if dataset == 'reuters10k':
        return 2000,15,4,0.002,0.002,5,0.5,0.5,1,'linear'
    if dataset == 'har':
        return 561,120,6,0.002,0.00002,10,0.9,0.9,5,'linear'
def gmmpara_init():
    theta_init=np.ones(n_centroid)/n_centroid
```

```
    u_init=np.zeros((latent_dim,n_centroid))
    lambda_init=np.ones((latent_dim,n_centroid))

theta_p=theano.shared(np.asarray(theta_init,dtype=theano.config.floatX),name
="pi")
    u_p=theano.shared(np.asarray(u_init,dtype=theano.config.floatX),name="u")
lambda_p=theano.shared(np.asarray(lambda_init,dtype=theano.config.floatX),n
ame="lambda")
    return theta_p,u_p,lambda_p
def get_gamma(tempz):
    temp_Z=T.transpose(K.repeat(tempz,n_centroid),[0,2,1])
    temp_u_tensor3=T.repeat(u_p.dimshuffle('x',0,1),batch_size,axis=0)

temp_lambda_tensor3=T.repeat(lambda_p.dimshuffle('x',0,1),batch_size,axis=0
)

temp_theta_tensor3=theta_p.dimshuffle('x','x',0)*T.ones((batch_size,latent_dim
,n_centroid))
    temp_p_c_z=K.exp(K.sum((K.log(temp_theta_tensor3)-
0.5*K.log(2*math.pi*temp_lambda_tensor3)-\
                   K.square(temp_Z-
temp_u_tensor3)/(2*temp_lambda_tensor3)),axis=1))+1e-10
    return temp_p_c_z/K.sum(temp_p_c_z,axis=-1,keepdims=True)
def vae_loss(x, x_decoded_mean):
    Z=T.transpose(K.repeat(z,n_centroid),[0,2,1])
    z_mean_t=T.transpose(K.repeat(z_mean,n_centroid),[0,2,1])
    z_log_var_t=T.transpose(K.repeat(z_log_var,n_centroid),[0,2,1])
    u_tensor3=T.repeat(u_p.dimshuffle('x',0,1),batch_size,axis=0)
    lambda_tensor3=T.repeat(lambda_p.dimshuffle('x',0,1),batch_size,axis=0)

theta_tensor3=theta_p.dimshuffle('x','x',0)*T.ones((batch_size,latent_dim,n_cen
troid))
```

```
    p_c_z=K.exp(K.sum((K.log(theta_tensor3)-
0.5*K.log(2*math.pi*lambda_tensor3)-\
                      K.square(Z-u_tensor3)/(2*lambda_tensor3)),axis=1))+1e-10
    gamma=p_c_z/K.sum(p_c_z,axis=-1,keepdims=True)
    gamma_t=K.repeat(gamma,latent_dim)
    if datatype == 'sigmoid':
        loss=alpha*original_dim * objectives.binary_crossentropy(x, x_decoded_mean)\
+K.sum(0.5*gamma_t*(latent_dim*K.log(math.pi*2)+K.log(lambda_tensor3)+
K.exp(z_log_var_t)/lambda_tensor3+K.square(z_mean_t-
u_tensor3)/lambda_tensor3),axis=(1,2))\
        -0.5*K.sum(z_log_var+1,axis=-1)\
        -
K.sum(K.log(K.repeat_elements(theta_p.dimshuffle('x',0),batch_size,0))*gamm
a,axis=-1)\
        +K.sum(K.log(gamma)*gamma,axis=-1)
    else:
        loss=alpha*original_dim * objectives.mean_squared_error(x, x_decoded_mean)\
+K.sum(0.5*gamma_t*(latent_dim*K.log(math.pi*2)+K.log(lambda_tensor3)+
K.exp(z_log_var_t)/lambda_tensor3+K.square(z_mean_t-
u_tensor3)/lambda_tensor3),axis=(1,2))\
        -0.5*K.sum(z_log_var+1,axis=-1)\
        -
K.sum(K.log(K.repeat_elements(theta_p.dimshuffle('x',0),batch_size,0))*gamm
a,axis=-1)\
        +K.sum(K.log(gamma)*gamma,axis=-1)
    return loss
def load_pretrain_weights(vade,dataset):
    ae = model_from_json(open('pretrain_weights/ae_'+dataset+'.json').read())
    ae.load_weights('pretrain_weights/ae_'+dataset+'_weights.h5')
    vade.layers[1].set_weights(ae.layers[0].get_weights())
    vade.layers[2].set_weights(ae.layers[1].get_weights())
```

```
    vade.layers[3].set_weights(ae.layers[2].get_weights())
    vade.layers[4].set_weights(ae.layers[3].get_weights())
    vade.layers[-1].set_weights(ae.layers[-1].get_weights())
    vade.layers[-2].set_weights(ae.layers[-2].get_weights())
    vade.layers[-3].set_weights(ae.layers[-3].get_weights())
    vade.layers[-4].set_weights(ae.layers[-4].get_weights())
    sample = sample_output.predict(X,batch_size=batch_size)
    if dataset == 'mnist':
        g = mixture.GMM(n_components=n_centroid,covariance_type='diag')
        g.fit(sample)
        u_p.set_value(floatX(g.means_.T))
        lambda_p.set_value((floatX(g.covars_.T)))
    if dataset == 'reuters10k':
        k = KMeans(n_clusters=n_centroid)
        k.fit(sample)
        u_p.set_value(floatX(k.cluster_centers_.T))
    if dataset == 'har':
        g =
mixture.GMM(n_components=n_centroid,covariance_type='diag',random_state
=3)
        g.fit(sample)
        u_p.set_value(floatX(g.means_.T))
        lambda_p.set_value((floatX(g.covars_.T)))
    print ('pretrain weights loaded!')
    return vade
def lr_decay():
    if dataset == 'mnist':

adam_nn.lr.set_value(floatX(max(adam_nn.lr.get_value()*decay_nn,0.0002)))

adam_gmm.lr.set_value(floatX(max(adam_gmm.lr.get_value()*decay_gmm,0.
0002)))
    else:
```

```
        adam_nn.lr.set_value(floatX(adam_nn.lr.get_value()*decay_nn))
        adam_gmm.lr.set_value(floatX(adam_gmm.lr.get_value()*decay_gmm))
    print ('lr_nn:%f'%adam_nn.lr.get_value())
    print ('lr_gmm:%f'%adam_gmm.lr.get_value())
def epochBegin(epoch):
    if epoch % decay_n == 0 and epoch!=0:
        lr_decay()
    '''
    sample = sample_output.predict(X,batch_size=batch_size)
    g = mixture.GMM(n_components=n_centroid,covariance_type='diag')
    g.fit(sample)
    p=g.predict(sample)
    acc_g=cluster_acc(p,Y)
    if epoch <1 and ispretrain == False:
        u_p.set_value(floatX(g.means_.T))
        print ('no pretrain,random init!')
    '''
    gamma = gamma_output.predict(X,batch_size=batch_size)
    acc=cluster_acc(np.argmax(gamma,axis=1),Y)
    global accuracy
    accuracy+=[acc[0]]
    if epoch>0 :
        #print ('acc_gmm_on_z:%0.8f'%acc_g[0])
        print ('acc_p_c_z:%0.8f'%acc[0])
    if epoch==1 and dataset == 'har' and acc[0]<0.77:
        print ('=========== HAR dataset:bad init!Please run again! ============')
        sys.exit(0)
def on_epoch_begin():
    dataset = 'mnist'
    db = sys.argv[1]
    if db in ['mnist','reuters10k','har']:
        dataset = db
```

```
    print ('training on: ' + dataset)
    ispretrain = True
    batch_size = 100
    latent_dim = 10
    intermediate_dim = [500,500,2000]
    theano.config.floatX='float32'
    accuracy=[]
    X,Y = load_data(dataset)

original_dim,epoch,n_centroid,lr_nn,lr_gmm,decay_n,decay_nn,decay_gmm,al
pha,datatype = config_init(dataset)
    theta_p,u_p,lambda_p = gmmpara_init()
    x = Input(batch_shape=(batch_size, original_dim))
    h = Dense(intermediate_dim[0], activation='relu')(x)
    h = Dense(intermediate_dim[1], activation='relu')(h)
    h = Dense(intermediate_dim[2], activation='relu')(h)
    z_mean = Dense(latent_dim)(h)
    z_log_var = Dense(latent_dim)(h)
    z = Lambda(sampling, output_shape=(latent_dim,))([z_mean, z_log_var])
    h_decoded = Dense(intermediate_dim[-1], activation='relu')(z)
    h_decoded = Dense(intermediate_dim[-2], activation='relu')(h_decoded)
    h_decoded = Dense(intermediate_dim[-3], activation='relu')(h_decoded)
    x_decoded_mean = Dense(original_dim, activation=datatype)(h_decoded)
    Gamma = Lambda(get_gamma, output_shape=(n_centroid,))(z)
    sample_output = Model(x, z_mean)
    gamma_output = Model(x,Gamma)
    vade = Model(x, x_decoded_mean)
    if ispretrain == True:
        vade = load_pretrain_weights(vade,dataset)
    adam_nn= Adam(lr=lr_nn,epsilon=1e-4)
    adam_gmm= Adam(lr=lr_gmm,epsilon=1e-4)
```

```
    vade.compile(optimizer=adam_nn,
loss=vae_loss,add_trainable_weights=[theta_p,u_p,lambda_p],add_optimizer=a
dam_gmm)
    epoch_begin=EpochBegin()
    vade.fit(X, X,
            shuffle=True,
            nb_epoch=epoch,
            batch_size=batch_size,
            callbacks=[epoch_begin])
def get_gamma(tempz):
    temp_Z=T.transpose(K.repeat(tempz,n_centroid),[0,2,1])
    temp_u_tensor3=T.repeat(u_p.dimshuffle('x',0,1),batch_size,axis=0)

temp_lambda_tensor3=T.repeat(lambda_p.dimshuffle('x',0,1),batch_size,axis=0
)

temp_theta_tensor3=theta_p.dimshuffle('x','x',0)*T.ones((batch_size,latent_dim
,n_centroid))
    temp_p_c_z=K.exp(K.sum((K.log(temp_theta_tensor3)-
0.5*K.log(2*math.pi*temp_lambda_tensor3)-\
                       K.square(temp_Z-
temp_u_tensor3)/(2*temp_lambda_tensor3)),axis=1))
    return temp_p_c_z/K.sum(temp_p_c_z,axis=-1,keepdims=True)
    ispretrain = True
    batch_size = 100
    latent_dim = 10
    intermediate_dim = [500,500,2000]
    theano.config.floatX='float32'
    X,Y = load_data()
    original_dim = 2000
    n_centroid = 4
    theta_p, u_p, lambda_p = gmm_para_init()
    x = Input(batch_shape=(batch_size, original_dim))
```

```
    h = Dense(intermediate_dim[0])(x)
    h=Activation('relu')(h)
    h=Dropout(0.2)(h)
    h = Dense(intermediate_dim[1])(h)
    h=Activation('relu')(h)
    h=Dropout(0.2)(h)
    h = Dense(intermediate_dim[2])(h)
    h=Activation('relu')(h)
    h=Dropout(0.2)(h)
    z_mean = Dense(latent_dim)(h)
    z_log_var = Dense(latent_dim)(h)
    z = Lambda(sampling, output_shape=(latent_dim,))([z_mean, z_log_var])
    z = Lambda(sampling, output_shape=(latent_dim,))([z_mean, z_log_var])
    h_decoded = Dense(intermediate_dim[-1])(z)
    h_decoded = Dense(intermediate_dim[-2])(h_decoded)
    h_decoded = Dense(intermediate_dim[-3])(h_decoded)
    x_decoded_mean = Dense(original_dim)(h_decoded)
    p_c_z = Lambda(get_gamma, output_shape=(n_centroid,))(z_mean)
    sample_output = Model(x, z_mean)
    p_c_z_output = Model(x, p_c_z)
    vade = Model(x, x_decoded_mean)
    vade.load_weights('trained_model_weights/reuters_all_weights_nn.h5')
    accuracy =

cluster_acc(np.argmax(p_c_z_output.predict(X,batch_size=batch_size),axis=1),
Y)
```

Arduino Code

```
#include <WiFiClient.h>
#include <ESP8266WebServer.h>
#include <ESP8266HTTPClient.h>
#include <Adafruit_SSD1306.h>
#include <Adafruit_GFX.h>
```

```
#include <Wire.h>
#include <Adafruit_MPU6050.h>
#include "MAX30105.h"
#include "heartRate.h"
#include <WifiLocation.h>
MAX30105 particleSensor;
Adafruit_SSD1306 display = Adafruit_SSD1306();
Adafruit_MPU6050 mpu;
Adafruit_Sensor *mpu_temp, *mpu_accel, *mpu_gyro;
const byte RATE_SIZE = 4; //Increase this for more averaging. 4 is good.
byte rates[RATE_SIZE]; //Array of heart rates
byte rateSpot = 0;
long lastBeat = 0; //Time at which the last beat occurred
float beatsPerMinute;
int beatAvg;
int buttonPin = D6;
int buttonState;
const char* googleApiKey = "AIzaSyBnKyOpsNq-vWYtrwayN3BkF3b4k3O9A_A";
const char *ssid = "xxxxx";
const char *password = "xxxx";
const char *host = "https://iotcloud.co.in";
String cdata;
int i, l, r, k, h, b, sp, cv, fv, lat, lon;
float t, a, x, y, z;
void setup() {
  Serial.begin(115200);
  WiFi.mode(WIFI_OFF);
  delay(1000);
  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);
  Serial.println("");
  Serial.print("Connecting");
```

```
  while (WiFi.status() != WL_CONNECTED)
  {
    delay(500);
    Serial.print("*");
  }
  Serial.println("");
  Serial.print("Connected to ");
  Serial.println(ssid);
  Serial.print("IP address: ");
  Serial.println(WiFi.localIP());
  while (!Serial)
    delay(10); // will pause Zero, Leonardo, etc until serial console opens
  Serial.println("Adafruit MPU6050 test!");
  if (!mpu.begin()) {
    Serial.println("Failed to find MPU6050 chip");
    while (1) {
      delay(10);
    }
  }
  Serial.println("MPU6050 Found!");
  mpu_accel = mpu.getAccelerometerSensor();
  mpu_accel->printSensorDetails();
  if (!particleSensor.begin(Wire, I2C_SPEED_FAST)) //Use default I2C port,
400kHz speed
  {
    Serial.println("MAX30105 was not found. Please check wiring/power. ");
    while (1);
  }
  Serial.println("Place your index finger on the sensor with steady pressure.");
  particleSensor.setup(); //Configure sensor with default settings
  particleSensor.setPulseAmplitudeRed(0x0A); //Turn Red LED to low to
indicate sensor is running
  particleSensor.setPulseAmplitudeGreen(0); //Turn off Green LED
```

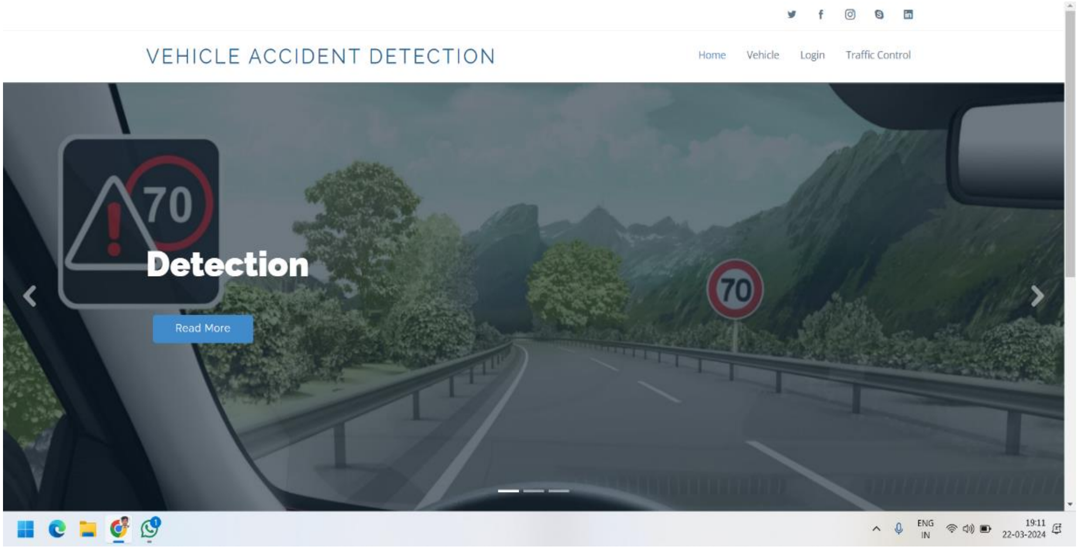
```
  particleSensor.enableDIETEMPRDY();
}
 loadAlgomodeParameters();
  int result = MAX32664.hubBegin();
  if (result == CMD_SUCCESS){
    Serial.println("Sensorhub begin!");
  }else{
    //stay here.
    while(1){
      Serial.println("Could not communicate with the sensor! please make proper
connections");
      delay(5000);
    }
  }
  bool ret = MAX32664.startBPTcalibration();
  while(!ret){
    delay(10000);
    Serial.println("failed calib, please retsart");
    //ret = MAX32664.startBPTcalibration();
  }
  delay(1000);
  //Serial.println("start in estimation mode");
  ret = MAX32664.configAlgoInEstimationMode();
  while(!ret){
    //Serial.println("failed est mode");
    ret = MAX32664.configAlgoInEstimationMode();
    delay(10000);
  }
  //MAX32664.enableInterruptPin();
  Serial.println("Getting the device ready..");
  delay(1000);
location_t loc = location.getGeoFromWiFi();
   /* Serial.println("Location request data");
```

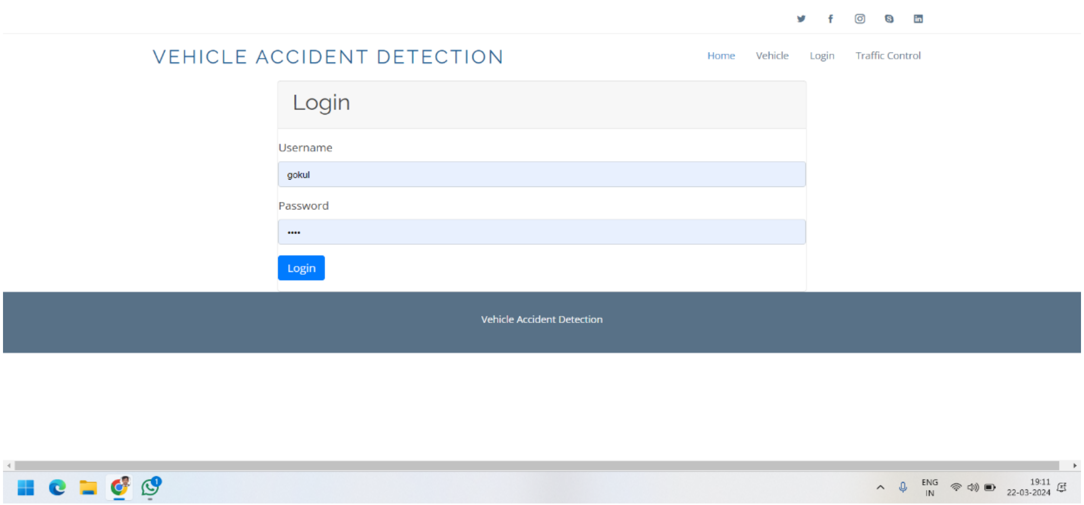
```
    Serial.println(location.getSurroundingWiFiJson());
    Serial.println("Latitude: " + String(loc.lat, 6));
    Serial.println("Longitude: " + String(loc.lon, 6));
    Serial.println("Accuracy: " + String(loc.accuracy));
 */
}
  pinMode(D5, INPUT);
  pinMode(D6, OUTPUT);
}
void mfioInterruptHndlr(){
  //Serial.println("i");
}
void enableInterruptPin(){
  //pinMode(mfioPin, INPUT_PULLUP);
  attachInterrupt(digitalPinToInterrupt(MAX32664.mfioPin),
mfioInterruptHndlr, FALLING);
}
void loadAlgomodeParameters(){
  algomodeInitialiser algoParameters;
  algoParameters.calibValSys[0] = 120;
  algoParameters.calibValSys[1] = 122;
  algoParameters.calibValSys[2] = 125;
  algoParameters.calibValDia[0] = 80;
  algoParameters.calibValDia[1] = 81;
  algoParameters.calibValDia[2] = 82;
  algoParameters.spo2CalibCoefA = 1.5958422;
  algoParameters.spo2CalibCoefB = -34.659664;
  algoParameters.spo2CalibCoefC = 112.68987;
  MAX32664.loadAlgorithmParameters(&algoParameters);
}
void loop() {
  digitalWrite(D6, HIGH);
  long irValue = particleSensor.getIR();
```

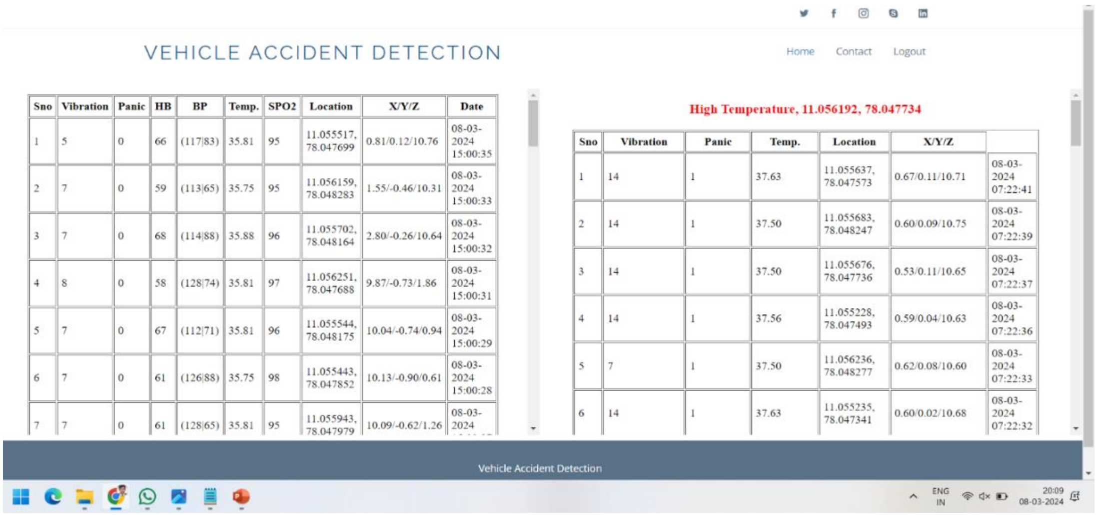
```
  float temperature = particleSensor.readTemperature();
  Serial.print("temperatureC=");
  Serial.println(temperature, 4);
  sensors_event_t accel;
  mpu_accel->getEvent(&accel);
uint8_t num_samples = MAX32664.readSamples();
  if(num_samples){
int l = MAX32664.max32664Output.sys;
int r = MAX32664.max32664Output.dia;
int h = MAX32664.max32664Output.hr;
int sp = MAX32664.max32664Output.spo2;
  }
  x = (accel.acceleration.x);
  y = (accel.acceleration.y);
  z = (accel.acceleration.z);
  i  = analogRead(A0);
  b = digitalRead(D5);
 lat = String(loc.lat, 6);
 lon = String(loc.lon, 6)
  if (irValue > 49000 && irValue < 65000)
  {
    cdata =  String(i) + String("/") + String(b) + String("/")  + String(h) + String("/") + String("(") + String(l) + String("|") + String(r) + String(")") + String("/") +  String(temperature) + String("/") + String(sp) + String("/") + String(lat) + String("/") + String(lon) + String("/") + String(x) + String("/") + String(y) + String("/") + String(z);
    String plaintext = cdata;
    String postData;
    postData = "status=Your ID/" + cdata;
    http.begin("http://iotcloud.co.in/iot_va/new.php");
    http.addHeader("Content-Type", "application/x-www-form-urlencoded");
    http.POST(postData);
    http.getString();
```

```
  }
if (irValue < 49000 || irValue > 65000 )
  {
    cdata =  String(i) + String("/") + String(b) + String("/") +  String(temperature)
+ String("/") + String(lat) + String("/")  + String(lon) + String("/") + String(x) +
String("/") + String(y) + String("/") + String(z);
    String plaintext = cdata;
    String postData;
    postData = "status=Your ID/" + cdata;
    http.begin("http://iotcloud.co.in/iot_va/new1.php");
    http.addHeader("Content-Type", "application/x-www-form-urlencoded");
    http.POST(postData);
    http.getString();
    Serial.println(cdata);
    Serial.println(k);
    Serial.println(httpCode);
    Serial.println(payload);
    http.end();
  }
  digitalWrite(D6, LOW);
  delay(1000);
}
```

SCREENSHOTS

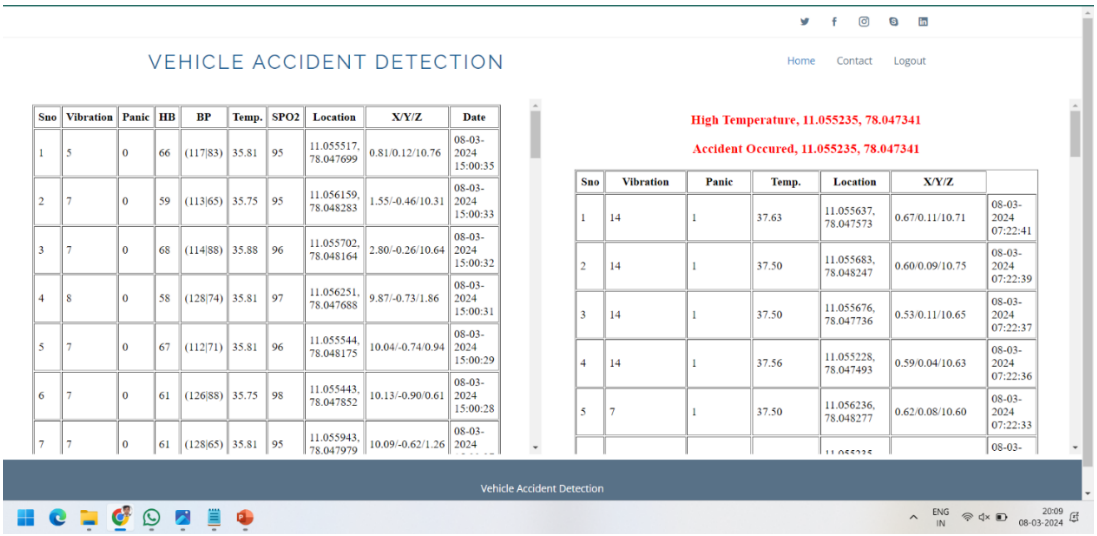






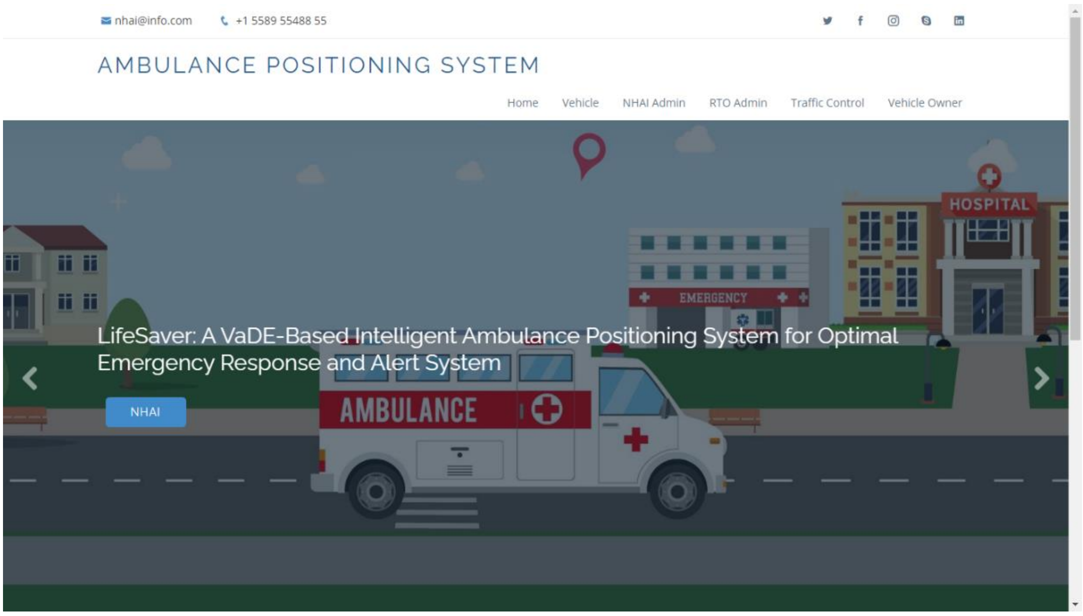
Sno	Vibration	Panic	HB	BP	Temp.	SPO2	Location	X/Y/Z	Date
1	5	0	66	(117|83)	35.81	95	11.055517, 78.047699	0.81/0.12/10.76	08-03-2024 15:00:35
2	7	0	59	(113|65)	35.75	95	11.056159, 78.048283	1.55/-0.46/10.31	08-03-2024 15:00:33
3	7	0	68	(114|88)	35.88	96	11.055702, 78.048164	2.80/-0.26/10.64	08-03-2024 15:00:32
4	8	0	58	(128|74)	35.81	97	11.056251, 78.047688	9.87/-0.73/1.86	08-03-2024 15:00:31
5	7	0	67	(112|71)	35.81	96	11.055544, 78.048175	10.04/-0.74/0.94	08-03-2024 15:00:29
6	7	0	61	(126|88)	35.75	98	11.055443, 78.047852	10.13/-0.90/0.61	08-03-2024 15:00:28
7	7	0	61	(128|65)	35.81	95	11.055943, 78.047979	10.09/-0.62/1.26	08-03-2024

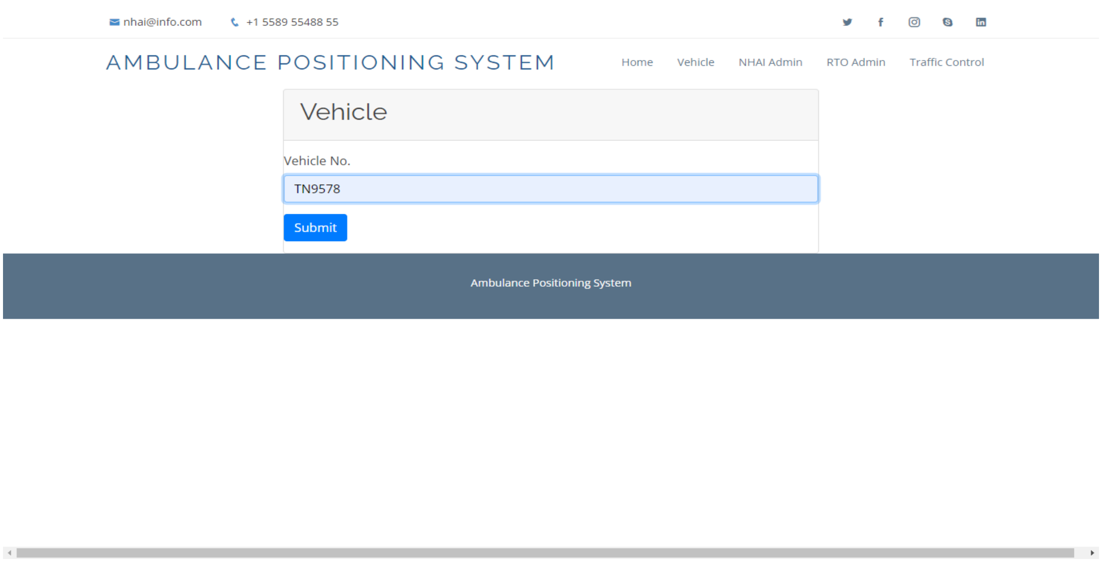
Sno	Vibration	Panic	Temp.	Location	X/Y/Z	
1	14	1	37.63	11.055637, 78.047573	0.67/0.11/10.71	08-03-2024 07:22:41
2	14	1	37.50	11.055683, 78.048247	0.60/0.09/10.75	08-03-2024 07:22:39
3	14	1	37.50	11.055676, 78.047736	0.53/0.11/10.65	08-03-2024 07:22:37
4	14	1	37.56	11.055228, 78.047493	0.59/0.04/10.63	08-03-2024 07:22:36
5	7	1	37.50	11.056236, 78.048277	0.62/0.08/10.60	08-03-2024 07:22:33
6	14	1	37.63	11.055235, 78.047341	0.60/0.02/10.68	08-03-2024 07:22:32

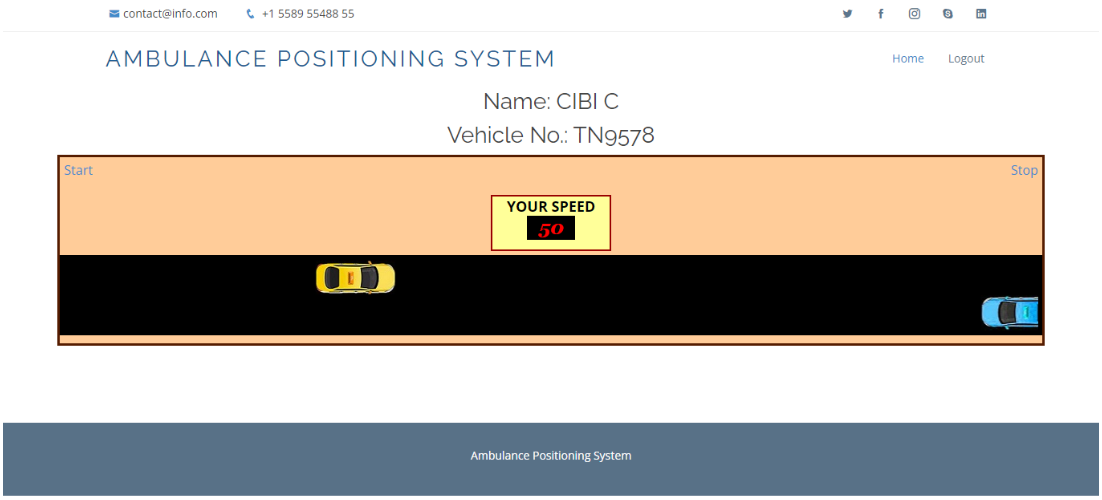


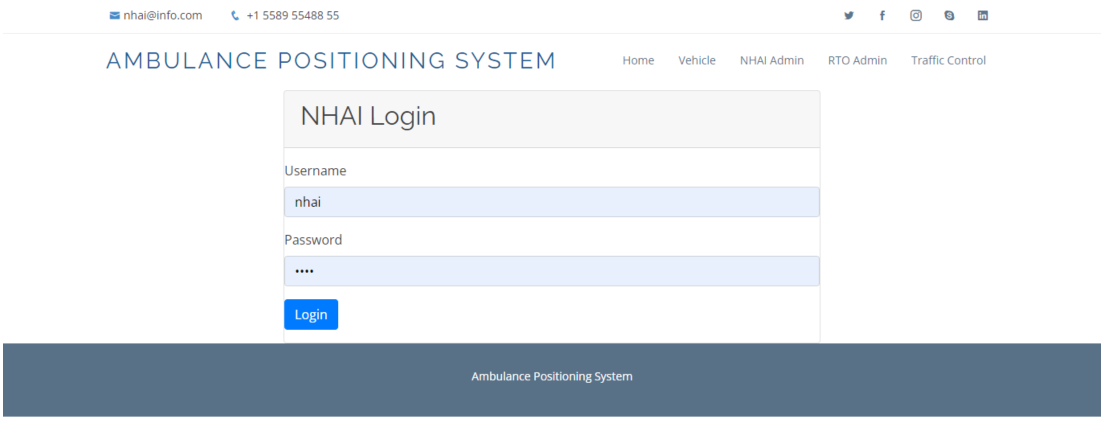
Sno	Vibration	Panic	HB	BP	Temp.	SPO2	Location	X/Y/Z	Date
1	5	0	66	(117|83)	35.81	95	11.055517, 78.047699	0.81/0.12/10.76	08-03-2024 15:00:35
2	7	0	59	(113|65)	35.75	95	11.056159, 78.048283	1.55/-0.46/10.31	08-03-2024 15:00:33
3	7	0	68	(114|88)	35.88	96	11.055702, 78.048164	2.80/-0.26/10.64	08-03-2024 15:00:32
4	8	0	58	(128|74)	35.81	97	11.056251, 78.047688	9.87/-0.73/1.86	08-03-2024 15:00:31
5	7	0	67	(112|71)	35.81	96	11.055544, 78.048175	10.04/-0.74/0.94	08-03-2024 15:00:29
6	7	0	61	(126|88)	35.75	98	11.055443, 78.047852	10.13/-0.90/0.61	08-03-2024 15:00:28
7	7	0	61	(128|65)	35.81	95	11.055943, 78.047979	10.09/-0.62/1.26	08-03-2024

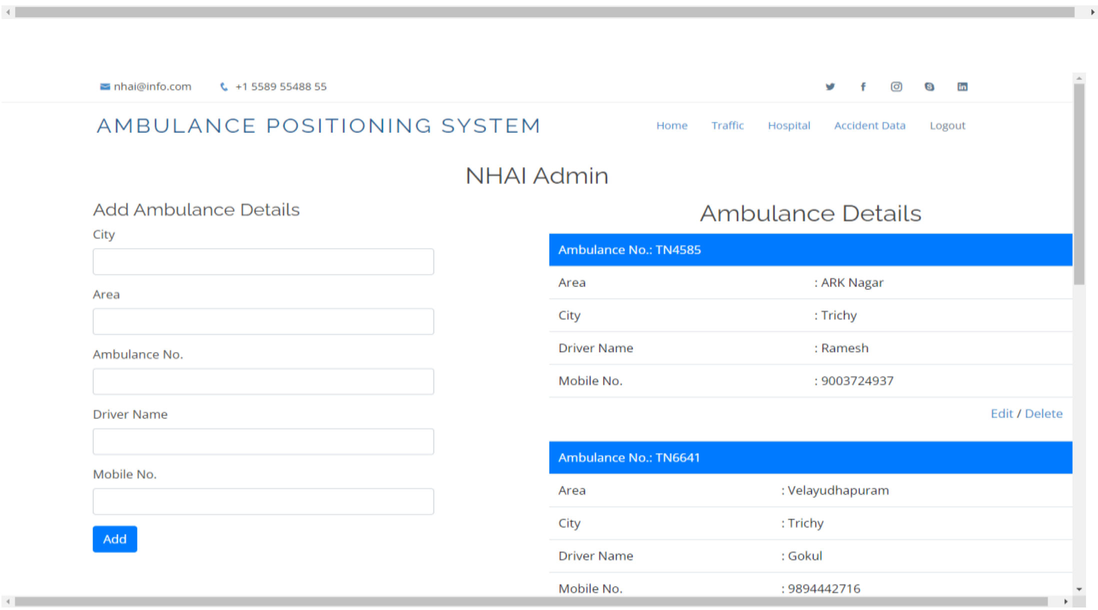
Sno	Vibration	Panic	Temp.	Location	X/Y/Z	
1	14	1	37.63	11.055637, 78.047573	0.67/0.11/10.71	08-03-2024 07:22:41
2	14	1	37.50	11.055683, 78.048247	0.60/0.09/10.75	08-03-2024 07:22:39
3	14	1	37.50	11.055676, 78.047736	0.53/0.11/10.65	08-03-2024 07:22:37
4	14	1	37.56	11.055228, 78.047493	0.59/0.04/10.63	08-03-2024 07:22:36
5	7	1	37.50	11.056236, 78.048277	0.62/0.08/10.60	08-03-2024 07:22:33

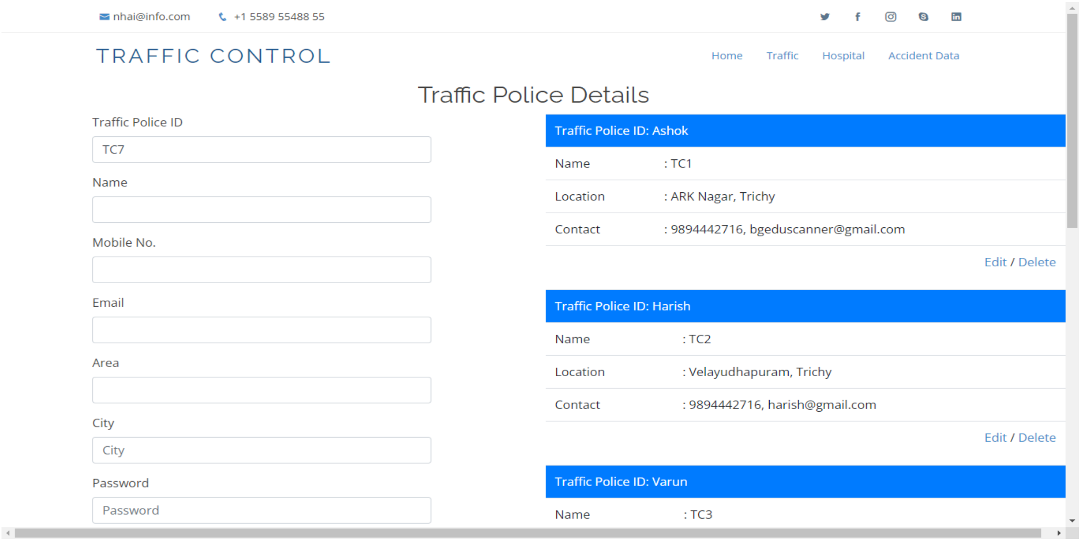


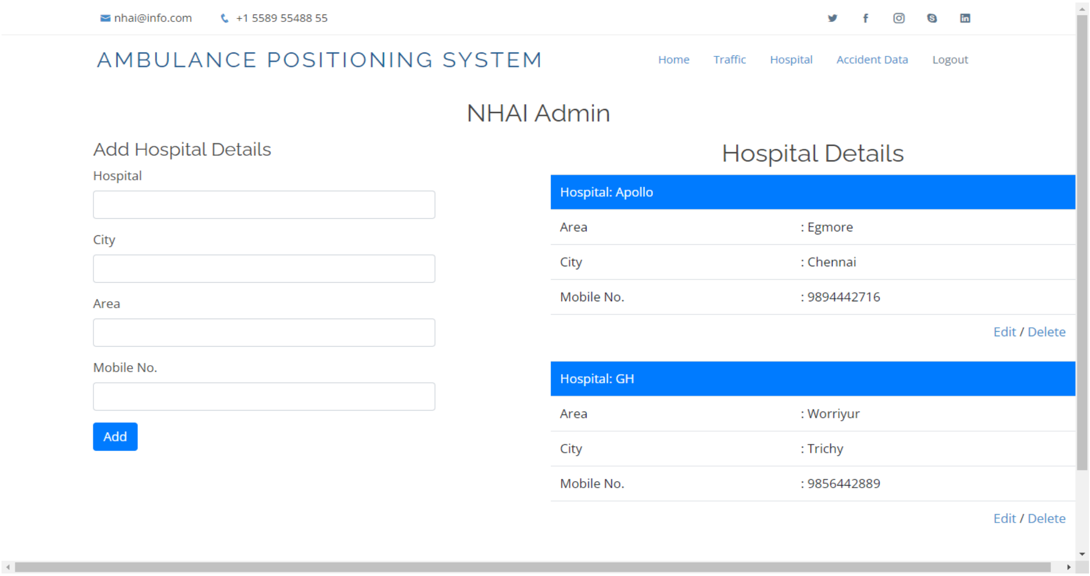


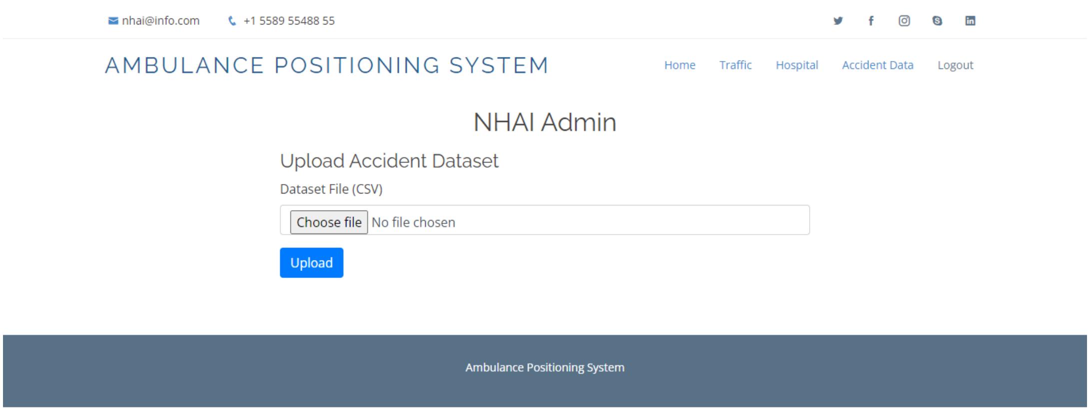

contact@info.com
+1 5589 55488 55
AMBULANCE POSITIONING SYSTEM
Home
Logout
Name: CIBI C
Vehicle No.: TN9578
Start
Stop
YOUR SPEED
50
Ambulance Positioning System

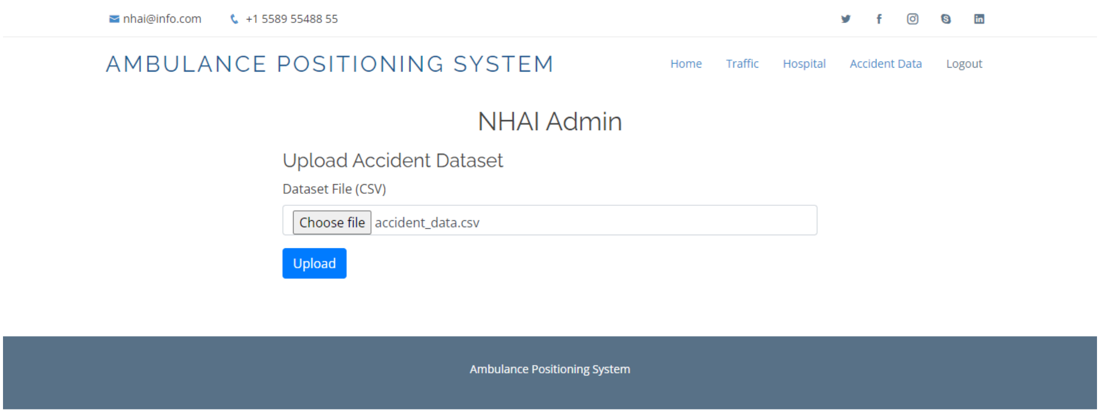

nhai@info.com
+1 5589 55488 55
AMBULANCE POSITIONING SYSTEM
Home
Vehicle
NHAI Admin
RTO Admin
Traffic Control
NHAI Login
Username
nhai
Password
Login
Ambulance Positioning System

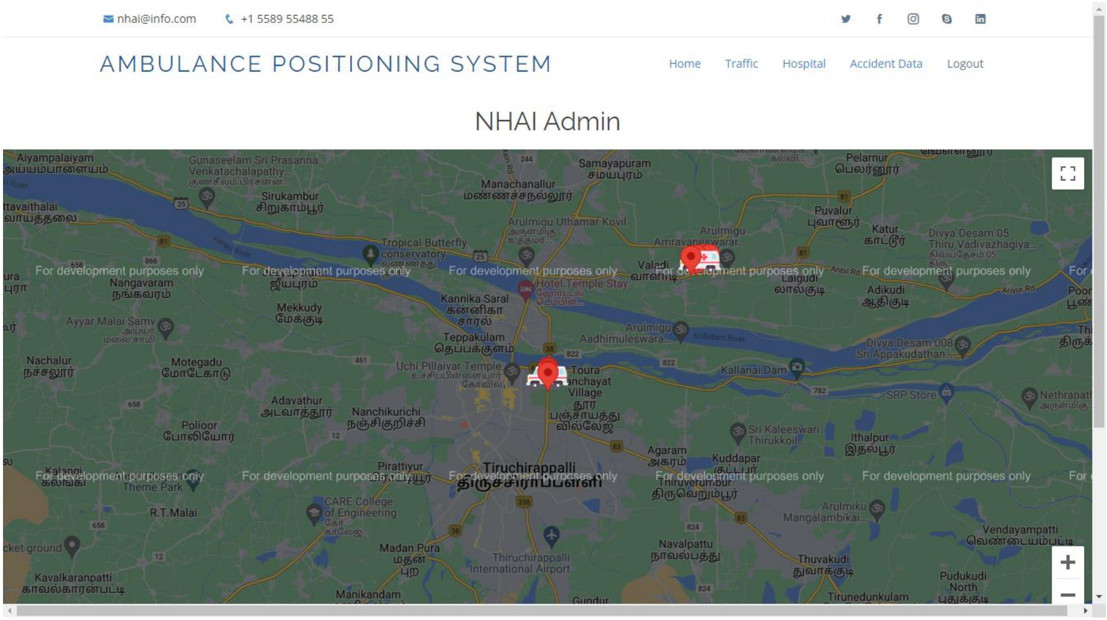

nhai@info.com
+1 5589 55488 55
AMBULANCE POSITIONING SYSTEM
Home
Traffic
Hospital
Accident Data
Logout
NHAI Admin
Add Ambulance Details
City
Area
Ambulance No.
Driver Name
Mobile No.
Add
Ambulance Details
Ambulance No.: TN4585
Area : ARK Nagar
City : Trichy
Driver Name : Ramesh
Mobile No. : 9003724937
Edit / Delete
Ambulance No.: TN6641
Area : Velayudhapuram
City : Trichy
Driver Name : Gokul
Mobile No. : 9894442716

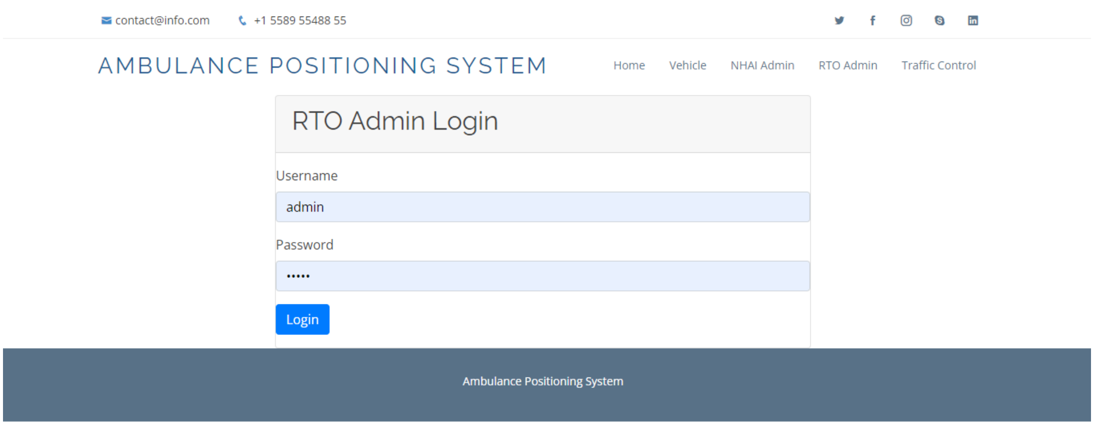

nhai@info.com
+1 5589 55488 55
TRAFFIC CONTROL
Home
Traffic
Hospital
Accident Data
Traffic Police Details
Traffic Police ID
TC7
Name
Mobile No.
Email
Area
City
City
Password
Password
Traffic Police ID: Ashok
Name : TC1
Location : ARK Nagar, Trichy
Contact : 9894442716, bgeduscanner@gmail.com
Edit / Delete
Traffic Police ID: Harish
Name : TC2
Location : Velayudhapuram, Trichy
Contact : 9894442716, harish@gmail.com
Edit / Delete
Traffic Police ID: Varun
Name : TC3

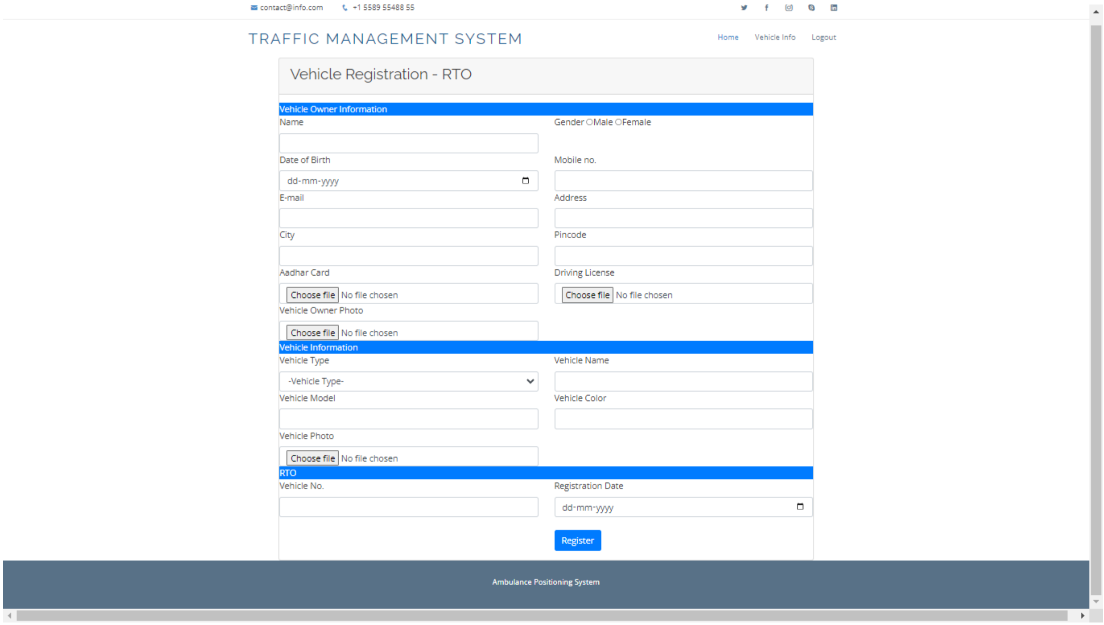

nhai@info.com
+1 5589 55488 55
AMBULANCE POSITIONING SYSTEM
Home
Traffic
Hospital
Accident Data
Logout
NHAI Admin
Add Hospital Details
Hospital
City
Area
Mobile No.
Add
Hospital Details
Hospital: Apollo
Area : Egmore
City : Chennai
Mobile No. : 9894442716
Edit / Delete
Hospital: GH
Area : Worriyur
City : Trichy
Mobile No. : 9856442889
Edit / Delete

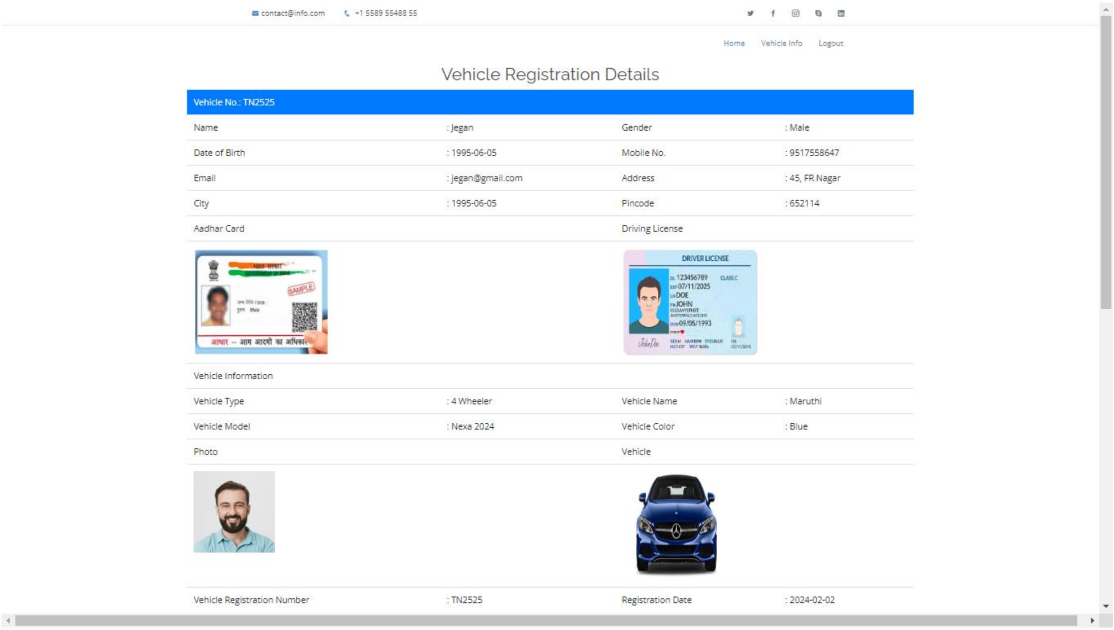

nhai@info.com
+1 5589 55488 55
AMBULANCE POSITIONING SYSTEM
Home
Traffic
Hospital
Accident Data
Logout
NHAI Admin
Upload Accident Dataset
Dataset File (CSV)
Choose file No file chosen
Upload
Ambulance Positioning System

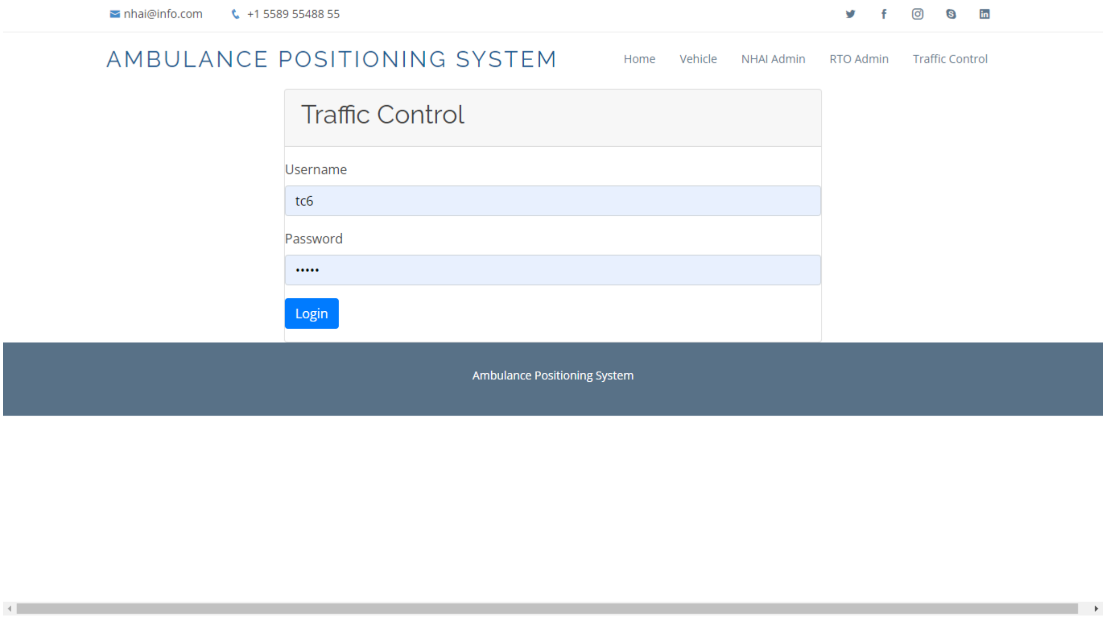

nhai@info.com
+1 5589 55488 55
AMBULANCE POSITIONING SYSTEM
Home
Traffic
Hospital
Accident Data
Logout
NHAI Admin
Upload Accident Dataset
Dataset File (CSV)
Choose file
accident_data.csv
Upload
Ambulance Positioning System


nhai@info.com
+1 5589 55488 55
AMBULANCE POSITIONING SYSTEM
Home
Traffic
Hospital
Accident Data
Logout
NHAI Admin
For development purposes only
Tiruchirappalli
Samayapuram
Manachanallur
Sirukambur
Nangavaram
Mekkudy
Kannika Saral
Teppakulam
Adavathur
Nanchikurichi
Nachalur
Motegadu
Polioor
R.T Malai
Kavalkaranpatti
Madan Pura
Thiruchirappalli International Airport
Navalpattu
Thuvakudi
Kuddapar
Agaram
Ithalpur
Puvalur
Katur
Adikudi
Lalgudi
Pelarnur
Vendayampatti
Pudukodi North
Tirunedunkulam
Manikandam

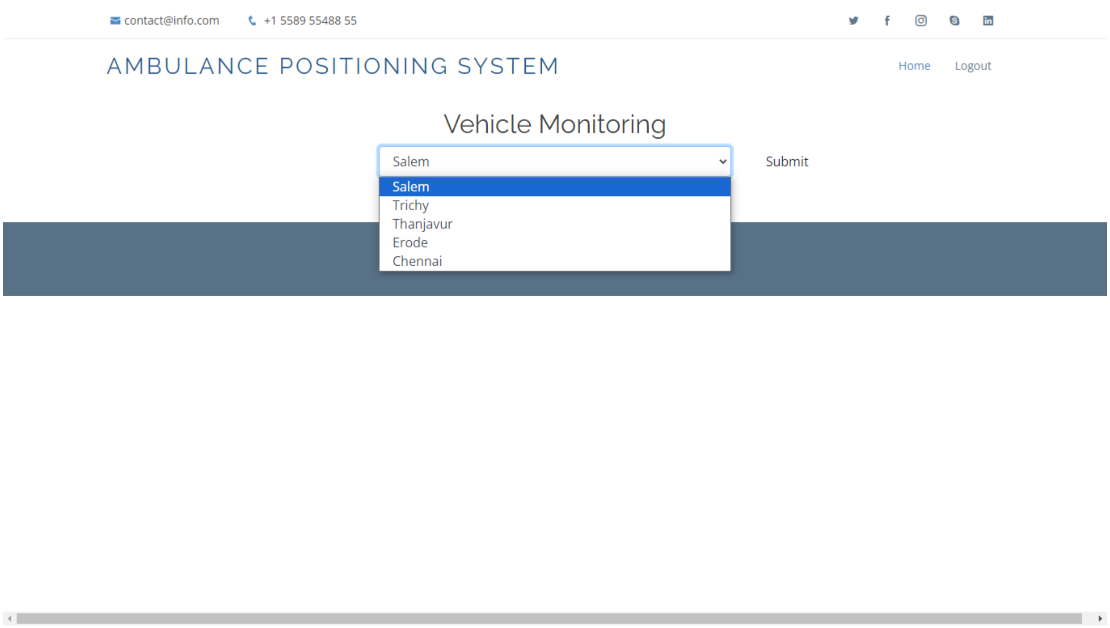

contact@info.com
+1 5589 55488 55
AMBULANCE POSITIONING SYSTEM
Home
Vehicle
NHAI Admin
RTO Admin
Traffic Control
RTO Admin Login
Username
admin
Password
•••••
Login
Ambulance Positioning System


contact@info.com
+1 5589 55488 55
TRAFFIC MANAGEMENT SYSTEM
Home
Vehicle Info
Logout
Vehicle Registration - RTO
Vehicle Owner Information
Name
Gender Male Female
Date of Birth
dd-mm-yyyy
Mobile no.
E-mail
Address
City
Pincode
Aadhar Card
Choose file No file chosen
Driving License
Choose file No file chosen
Vehicle Owner Photo
Choose file No file chosen
Vehicle Information
Vehicle Type
-Vehicle Type-
Vehicle Name
Vehicle Model
Vehicle Color
Vehicle Photo
Choose file No file chosen
RTO
Vehicle No.
Registration Date
dd-mm-yyyy
Register
Ambulance Positioning System


contact@info.com
+1 5589 55488 55
Home
Vehicle Info
Logout
Vehicle Registration Details
Vehicle No.: TN2525
Name
: Jegan
Gender
: Male
Date of Birth
: 1995-06-05
Mobile No.
: 9517558647
Email
: jegan@gmail.com
Address
: 45, FR Nagar
City
: 1995-06-05
Pincode
: 652114
Aadhar Card
Driving License
SAMPLE
DRIVER LICENSE
Vehicle Information
Vehicle Type
: 4 Wheeler
Vehicle Name
: Maruthi
Vehicle Model
: Nexa 2024
Vehicle Color
: Blue
Photo
Vehicle
Vehicle Registration Number
: TN2525
Registration Date
: 2024-02-02


nhai@info.com
+1 5589 55488 55
AMBULANCE POSITIONING SYSTEM
Home
Vehicle
NHAI Admin
RTO Admin
Traffic Control
Traffic Control
Username
tc6
Password
Login
Ambulance Positioning System


contact@info.com
+1 5589 55488 55
AMBULANCE POSITIONING SYSTEM
Home
Logout
Vehicle Monitoring
Salem
Submit
Ambulance Positioning System


contact@info.com
+1 5589 55488 55
AMBULANCE POSITIONING SYSTEM
Home
Logout
Vehicle Monitoring
Salem
Submit
Salem
Trichy
Thanjavur
Erode
Chennai

contact@info.com +1 5589 55488 55

AMBULANCE POSITIONING SYSTEM

Home Logout

Vehicle Monitoring

Salem Submit

Sno	Vehicle No.	Vehicle Speed (Above 70)	Accident Status	Date / Time
1	TN2121	110	Accident Occured	2024-03-08 04:58:27
2	TN2121	120	Accident Occured	2024-03-08 05:00:17
3	TN2121	100	Accident Occured	2024-03-08 05:00:19
4	TN2525	120	No Accident	2024-03-22 11:38:25
5	TN2525	120	No Accident	2024-03-22 11:39:01
6	TN2525	110	Accident Occured	2024-03-22 11:39:12
7	TN9578	110	No Accident	2024-03-22 12:01:23
8	TN2525	120	No Accident	2024-03-22 18:52:10

Ambulance Positioning System

nhai@info.com +1 5589 55488 55

AMBULANCE POSITIONING SYSTEM

Home Vehicle NHAI Admin RTO Admin Traffic Control

Vehicle Owner Login

Username

cib002

Password

•••••

Login

Incorrect username/password!!!

Ambulance Positioning System

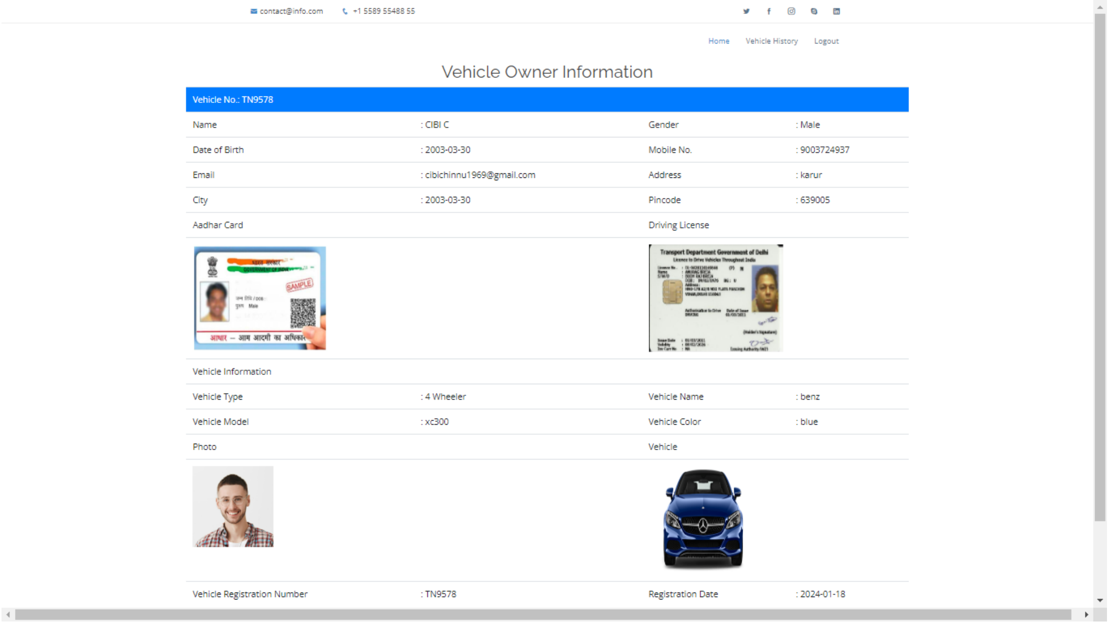
contact@info.com +1 5589 55488 55

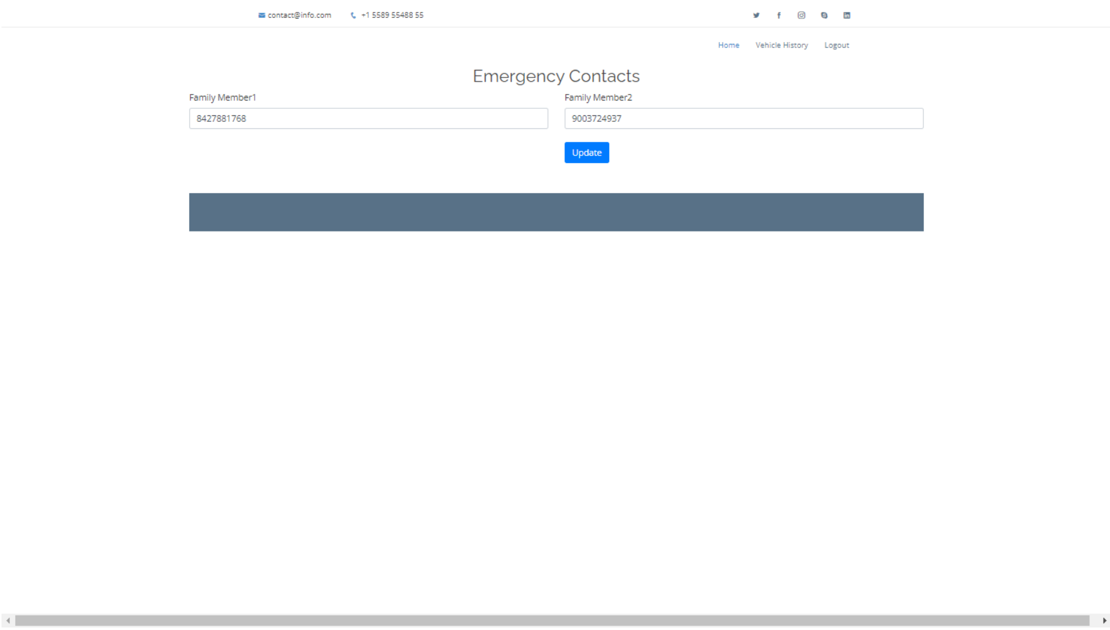
Home Vehicle History Logout

Vehicle History

Vehicle No.: TN9578

Sno	Vehicle Speed (Above 70)	Accident Status	Route	Date / Time
1	70	Accident Occured	Erode	2024-03-22 12:00:14
2	110	No Accident	Thanjavur	2024-03-22 12:00:40
3	110	No Accident	Trichy	2024-03-22 12:01:23
4	120	Accident Occured	Chennai	2024-03-22 12:01:45


contact@info.com
+1 5589 55488 55
Home
Vehicle History
Logout
Vehicle Owner Information
Vehicle No.: TN9578
Name
: CIBI C
Gender
: Male
Date of Birth
: 2003-03-30
Mobile No.
: 9003724937
Email
: cibichinnu1969@gmail.com
Address
: karur
City
: 2003-03-30
Pincode
: 639005
Aadhar Card
Driving License
Vehicle Information
Vehicle Type
: 4 Wheeler
Vehicle Name
: benz
Vehicle Model
: xc300
Vehicle Color
: blue
Photo
Vehicle
Vehicle Registration Number
: TN9578
Registration Date
: 2024-01-18


contact@info.com
+1 5589 55488 55
Home
Vehicle History
Logout
Emergency Contacts
Family Member1
8427881768
Family Member2
9003724937
Update

REFERENCES

1. T. D. Hanawy Hussein, M. Frikha and J. Rahebi, "Harris Hawks Optimization for Ambulance Vehicle Routing in Smart Cities", Eastern-European Journal of Enterprise Technologies, vol. 122, no. 3, 2023.
2. Chowdhury, S. Kaisar, M. E. Khoda, R. Naha, M. A. Khoshkholghi and M. Aiash, "IoT-based emergency vehicle services in intelligent transportation system", Sensors, vol. 23, no. 11, pp. 5324, 2023.
3. U. Mittal and P. Chawla, "Acoustic Based Emergency Vehicle Detection Using Ensemble of deep Learning Models", Procedia Computer Science, vol. 218, pp. 227-234, 2023.
4. S. J. Arul, M. B. S, L. S, G. Kaliyaperumal Sufiyan and J. K. K. A, "Modelling and Simulation of Smart Traffic Light System for Emergency Vehicle using Image Processing Techniques", 2023 3rd International Conference on Innovative Practices in Technology and Management (ICIPTM), pp. 1-4, 2023.
5. M. A. Abdeen, M. H. Ahmed, H. Seliem, T. R. Sheltami, T. M. Alghamdi and M. El-Nainay, "A Novel Smart Ambulance System-Algorithm Design Modeling and Performance Analysis", IEEE Access, vol. 10, pp. 42656-42672, 2022.
6. T. Darwassh, Hanawy Hussein, M. Frikha, S. Ahmed and J. Rahebi, "BA-CNN: Bat algorithm-based convolutional neural network algorithm for ambulance vehicle routing in smart cities", Mobile Information Systems, vol. 2022, 2022.
7. S. Mahalakshmi, T. Ragunthar, N. Veena, S. Sumukha and P. R. Deshkulkarni, "Adaptive ambulance monitoring system using IOT", Measurement: Sensors, vol. 24, pp. 100555, 2022.
8. Amrouche, Y. Bentrcia, A. Abed and N. Hezil, "Vehicle Detection and Tracking in Real-time using YOLOv4-tiny", 2022 7th International Conference on Image and Signal Processing and their Applications (ISPA), pp. 1-5, 2022.

9. R. Wu, Q. Zhou, H. Yu and Y. Wang, "Ambulance dispatching algorithm in shenzhen based on deep reinforcement learning", Computer Engineering, 2022.
10. G. Liu, J. Qu, X. Li and J. Wu, "Dynamic ambulance redeployment based on deep reinforcement learning", Journal of Management Sciences in China, vol. 23, no. 2, pp. 39-53, 2021.
11. X. Zhang, Q. Yan, J. Wu, B. Zhang, L. Zhou and Y. Wang, "Problems and countermeasures in the development of domestic pre-hospital emergency system", Medical Information, vol. 33, no. 23, pp. 12-14, 2020.
12. Z. Wang, T. Xia, R. Jiang, X. Liu, K.-S. Kim, X. Song, et al., "Forecasting ambulance demand with profiled human mobility via heterogeneous multi-graph neural networks", 2021 IEEE 37th International Conference on Data Engineering (ICDE), pp. 1751-1762, 2021.
13. A. Baghel, A. Srivastava, A. Tyagi, S. Goel and P. Nagrath, "Analysis of Ex-YOLO Algorithm with Other Real-Time Algorithms for Emergency Vehicle Detection", pp. 607-618, 2020.
14. Karkar, "Smart ambulance system for highlighting emergency-routes", 2019 Third World Conference on Smart Trends in Systems Security and Sustainablity (WorldS4), pp. 255-259, 2019.ss
15. Dumka and A. Sah, "Smart ambulance system using concept of big data and internet of things", Healthcare data analytics and management: Elsevier, pp. 155-176, 2019.
16. Suthaputchakun and Y. Cao, "Ambulance-to-traffic light controller communications for rescue mission enhancement: A Thailand use case", IEEE Communications Magazine, vol. 57, no. 12, pp. 91-97, 2019.
17. H. Mehta, P. Kanani and P. Lande, "Google maps", International Journal of Computer Applications, vol. 178, no. 8, pp. 41-46, 2019.

18. Mohamed N. Ashmawy, Ahmad M. Khairy, Mohamed W. Hamdy, Anas El-Shazly, Karim El-Rashidy, Mohamed Salah, et al., "SmartAmb: An Integrated Platform for Ambulance Routing and Patient Monitoring", Published in 2019 31st International Conference on Microelectronics (ICM) by IEEE.
19. C. Senthamilarasi, J. Jansi Rani, B. Vidhya and H. Aritha, "A Smart Patient Health Monitoring System Using IoT", International Journal of Pure and Applied Mathematics, vol. 119, no. 16, 2018.
20. A. B. Faiz, A. Imteaj and M. Chowdhury, "Smart vehicle accident detection and alarming system using a smartphone", 2015 International Conference on Computer and Information Engineering (ICCIE), pp. 66-69, 2015.

Office of the Controller General of Patents, Designs & Trade Marks
Department for Promotion of Industry and Internal Trade
Ministry of Commerce & Industry,
Government of India

INTELLECTUAL PROPERTY INDIA
PATENTS | DESIGNS | TRADE MARKS
GEOGRAPHICAL INDICATIONS

Application Details	
APPLICATION NUMBER	202441018306
APPLICATION TYPE	ORDINARY APPLICATION
DATE OF FILING	13/03/2024
APPLICANT NAME	1 . Ms. K. Anitha 2 . Mrs. P. Sanmugavalli 3 . M. Dharnesh 4 . S. Gokul 5 . C. Cibi 6 . K. Dharani Kumar
TITLE OF INVENTION	VEHICLE BLACKBOX MODEL INTEGRATED WITH IOT AND DEEP LEARNING TO SAVE LIVES
FIELD OF INVENTION	ELECTRONICS
E-MAIL (As Per Record)	senanipindia@gmail.com
ADDITIONAL-EMAIL (As Per Record)	
E-MAIL (UPDATED Online)	
PRIORITY DATE	
REQUEST FOR EXAMINATION DATE	--
PUBLICATION DATE (U/S 11A)	29/03/2024

Application Status	
APPLICATION STATUS	Awaiting Request for Examination

View Documents

Filed → Published → RQ Filed → Under Examination → Disposed

In case of any discrepancy in status, kindly contact ipo-helpdesk@nic.in

(12) PATENT APPLICATION PUBLICATION | (21) Application No.202441018306 A

(19) INDIA

(22) Date of filing of Application :13/03/2024 | (43) Publication Date : 29/03/2024

(54) Title of the invention : VEHICLE BLACKBOX MODEL INTEGRATED WITH IOT AND DEEP LEARNING TO SAVE LIVES

(51) International classification	:G08B0025010000, G06N0003080000, A61G0003000000, A61G0001020000, G08G0001000000	(71)Name of Applicant : 1)Ms. K. Anitha Address of Applicant :Assistant Professor, Department of Information Technology, M.Kumarasamy College of Engineering, Karur-639113 ---------- ----------- 2)Mrs. P. Sanmugavalli 3)M. Dharnesh 4)S. Gokul 5)C. Cibi 6)K. Dharani Kumar Name of Applicant : NA Address of Applicant : NA (72)Name of Inventor : 1)Ms. K. Anitha Address of Applicant :Assistant Professor, Department of Information Technology, M.Kumarasamy College of Engineering, Karur-639113 ---------- ---------- 2)Mrs. P. Sanmugavalli Address of Applicant :Assistant Professor, Department of Electronics and Communication Engineering, M.Kumarasamy College of Engineering, Karur-639113 ---------- ---------- 3)M. Dharnesh Address of Applicant :UG Student, Department of Information Technology, M.Kumarasamy College of Engineering, Karur-639113 ---------- ---------- 4)S. Gokul Address of Applicant :UG Student, Department of Information Technology, M.Kumarasamy College of Engineering, Karur-639113 ---------- ---------- 5)C. Cibi Address of Applicant :UG Student, Department of Electronics and Communication Engineering, M.Kumarasamy College of Engineering, Karur-639113 ---------- ----------- 6)K. Dharani Kumar Address of Applicant :UG Student, Department of Electronics and Communication Engineering, M.Kumarasamy College of Engineering, Karur-639113 ---------- -----------
(86) International Application No Filing Date	:NA :NA	
(87) International Publication No	: NA	
(61) Patent of Addition to Application Number Filing Date	:NA :NA	
(62) Divisional to Application Number Filing Date	:NA :NA	

(57) Abstract :

Vehicle BlackBox Model Integrated with IoT and Deep Learning to Save Lives ABSTRACT Road traffic accidents (RTAs) are becoming more common nowadays, as evidenced by the fact that the number of accidents is increasing on a daily basis. Many a times an accident goes unnoticed for hours before help comes in. Due to all these factors, there is a high rate of mortality among the accident victims. In addition to this there is delay in the ambulance reaching the hospital due to the traffic congestion between accident location and hospital which increases the chances of the death of victim. The lack of immediate help to save a life is one of the most prominent causes of death in a road traffic accident. Some various methods and techniques can assist in minimizing the frequency of traffic road accidents and saving lives. There is a need to introduce a system to reduce the loss of life due to accidents and the time taken by the ambulance to reach the hospital. This will result in lives being saved and injured persons being rescued. In this project, to this end, propose an accident reporting, ambulance positioning, alert ambulance and rescue system named Vehicle BlackBox Model based on IoT and Deep Learning for Intelligent Transportation System in urban environments. The proposed project to shorten the response time for ambulance arrival at the scene of a road accident by employing an unsupervised generative clustering approach with Variational Deep Embedding (VaDE) sounds innovative and promising. The proposed method is categorized into two phases: accident detection, based on the accident detected location predict the ambulance to arrive the spot, alert the ambulance driver to reach the spot, real-time alerts to family members, hospitals, and traffic departments, facilitating route clearance for expedited ambulance travel. Another phase of the proposed system is to position the ambulance based on the data set given to the VaDE. This proposed system contributes to reducing the loss of life in road traffic accidents, minimizing response times, and improving overall emergency response efficiency. The proposed system can be achieved through IOT and Deep Learning.

No. of Pages : 14 No. of Claims : 3