DEVELOPMENT OF AN INTELLIGENT MOBILE-BASED EMERGENCY DETECTION AND AUTOMATED SOS SYSTEM

BY

OBINALI CHUKWUEMEKA

GOU/U22/CSC/923

DEPARTMENT OF COMPUTER SCIENCE AND MATHEMATICS, FACULTY OF NATURAL SCIENCES AND ENVIRONMENTAL STUDIES,

GODFREY OKOYE UNIVERSITY, ENUGU STATE. **IN PARTIAL FULFILLMENT OF THE AWARD OF BACHELOR OF**

SCIENCE (B.SC), DEGREE IN COMPUTER SCIENCE.

SUPERVISOR: ENGR DR LUCY I. EZIGBO

JANUARY, 2026.

APPROVAL PAGE

This research, titled "**Development of An Intelligent Mobile-Based Emergency Detection and Automated SOS System**" has been assessed and approved by the Department of Computer Science and Mathematics Committees in the Faculty of Natural Sciences and Environmental Studies, Godfrey Okoye University, Enugu, Nigeria.

Supervisor	Signature	Date
External Examiner External Examiner	Signature	Date
Dr. Frank Okebanama HOD, Department of Computer Science and Mathematics	Signature	Date
Prof. N. M. Ozofor Dean, Faculty of Computing And Information Technology.	Signature	Date

CERTIFICATION

I certify that I completed the research included therein and that it was primarily based on my observations and investigation. Consequently, I will fully accept the University's decision if I am found to have cheated on the assignment.

_____

OBINALI CHUKWUEMEKA
GOU/U22/CSC/923

DEDICATION

This project is dedicated to God Almighty for His grace and guidance throughout this study. It is also committed to my parents and family members, whose constant support, encouragement, and prayers contributed immensely to the successful completion of this work.

ACKNOWLEDGMENTS

I want to express my sincere gratitude to my project supervisor, Engr. Dr. Lucy I. Ezigbo, for her guidance, constructive criticism, and professional support throughout the course of this project. Her insights and supervision greatly contributed to the successful completion of this work.

I also extend my appreciation to the Head of the Department, Dr Mr. Franklyn Okebanama, Faculty Dean, Prof. Ifeyinwa Angela Ajah and other lecturers in the Department of Computer Science for the knowledge, discipline, and academic foundation they have imparted during my course of study.

My profound gratitude goes to my parents, Mr. and Mrs. Obinali, for their constant love, encouragement, financial support, and prayers, which sustained me throughout this academic journey. I am equally grateful to my sister, Obinali Annabel, for her moral support and words of encouragement, especially during challenging moments.

Finally, I acknowledge all friends, colleagues, and individuals who contributed in one way or another, through advice, motivation, or assistance, to the successful completion of this project. I am deeply grateful to everyone who played a role, directly or indirectly, in making this work a success.

ABSTRACT

The increasing prevalence of insecurity, violent crimes, accidents, and sudden medical emergencies has heightened the need for effective and timely emergency response systems, particularly in environments where access to immediate help is limited. Traditional emergency response methods largely depend on manual actions such as phone calls or panic buttons, which may be ineffective when victims are incapacitated, restrained, unconscious, or unable to react promptly. This challenge underscores the need for intelligent systems capable of autonomously detecting danger and initiating emergency responses without relying solely on user intervention.

The core problem this study addresses is the inadequacy of existing safety solutions for situations where users cannot manually trigger emergency alerts. Delays in communication during such incidents significantly reduce the chances of rescue and survival. The study seeks to answer how artificial intelligence and smartphone sensor data can be used to automatically detect dangerous situations and promptly notify emergency contacts.

To address this problem, the study adopts a system design and implementation methodology. The proposed **Development of an Intelligent Mobile-Based Emergency Detection and Automated SOS System** leverages smartphone sensors, including motion, location, and sound data, combined with artificial intelligence techniques for anomaly and danger detection. A backend system is developed using Python-based frameworks to process data, manage alerts, and coordinate automated emergency messaging to predefined contacts. Simulated data and controlled test scenarios are used to evaluate system performance.

The results of the study demonstrate that the system can successfully detect abnormal or potentially dangerous situations and automatically transmit SOS alerts containing real-time location information. The prototype shows improved responsiveness compared to manual-only emergency systems and effectively reduces reliance on direct user interaction during critical moments.

The study contributes to the field of intelligent safety systems by presenting a practical and scalable approach to automated emergency detection and response. The findings highlight the potential of AI-driven solutions to enhance personal safety and emergency preparedness, while providing a foundation for future research, system optimization, and possible real-world deployment.

Table of Contents

Chapter Three: System Analysis and Design

Chapter Four: System Implementation

List of Tables

List of Figures

CHAPTER ONE
INTRODUCTION

Background of the study

Emergencies like insecurity, crime, road accidents, and sudden medical emergencies are becoming more prevalent, and these situations could be fatal or life-threatening. People in developing countries mostly experience these problems, and getting immediate emergency services may take forever, be unavailable, or difficult to access. Situations range from assault, kidnapping, road traffic crashes, or even sudden cardiac arrests. In most situations, a victim of an emergency may not be able to call for help due to unconsciousness or inability to move around, such as in cases of restraint or mental trauma.

When people do not report emergencies and get help away it really lowers the chances of survival during emergencies like assault and road traffic crashes. This is what studies have found out. Researchers such as Adeyemi and Okonkwo in 2024 and Li and their team in 2025 have looked into road traffic crashes and assault. They found out that getting prompt attention is very important for survival, during these emergencies. Smartphones and mobile applications are widely used despite most emergency prevention systems requiring physical user inputs, such as pressing the panic button or dialing an emergency number. This limitation will only allow the devices to work when the users are conscious enough to use them during emergencies. Hence, several efforts have been directed towards creating intelligent devices that can autonomously sense abnormal or risky situations and send an alert message or notify the appropriate authority.

Development of an Intelligent Mobile-Based Emergency Detection and Automated SOS System tackles this problem by leveraging smartphone sensors, artificial intelligence, and real-time communication to automatically detect emergencies and notify relevant parties. The system is always checking things like how a device's

moving, which way it is facing where it is and what sounds it is picking up. It is looking for anything that seems wrong or dangerous. If it finds something like that it will send out an alarm away. This alarm is like an SOS message that says something is wrong and tells where the user is. It sends this message to people or groups that the user has chosen to help in an emergency. There is an advancement towards autonomous emergency detection, which is also the trend towards automatic detection without manual initiation.

1.1Statement of the Problem

The research highlights the benefits of AI-driven approaches to improve individual safety and creates the basis for future developments towards automating the process of responding to emergency situations. The research focuses on inefficiency of the existing approach to handling the situation which involves conducting certain manual tasks, such as making a phone call. With more and more danger areas emerging, as well as the fact that people may not have enough capability to respond to danger themselves, it becomes necessary to have an intelligent system capable of recognizing an emergency independently using smartphone sensors.

1.2 Aim and Objectives of the Study

Aim

The aim of this project is the Development of An Intelligent Mobile-Based Emergency Detection and Automated Sos System

Objectives

In pursuit of the above-stated objective, the following specific objectives are set out:

1. Research and analyze existing alerting systems in order to ascertain the limitations especially in terms of automatic detection and response to danger.

2. Analysis of data from sensors in smartphones including motion, sound and location data as pertains to identification of danger and emergencies.

3. Designing an intelligent danger detection model based on AI, that can automatically detect any form of danger posed by behavior of users among others

4. Development of a mobile application for constant monitoring using the phone's sensors for user safety

5. Implementing the capability of automated sending of SOS messages along with live tracking and multimedia information (image/audio)

6. Evaluate the system performance in terms of danger detection and sending alerts

1.3 Significance of the Study

The current research relies on personal security enhancement, the effectiveness of emergency response processes, and intelligent mobile technology advances. By solving the main weaknesses of existing emergency systems, in particular, their necessity to be manually initiated, the suggested system allows one to respond to threatening situations in a more reliable and effective manner. Namely, if a person falls unconscious, is tied up, or cannot react because of fear or shock, the system's capability to detect danger automatically will improve the likelihood of receiving prompt help and thus survive.

First of all, the suggested system presents numerous advantages from a practical perspective. From a user's point of view, the suggested system becomes a constant personal safety tool capable of providing continuous monitoring of the situation and reacting to it without any need for any additional actions. It is particularly important for vulnerable groups such as students, travelers, lone employees, and older people. The presence of various channels of communication ensures that emergency messages will be sent through several channels including SMS, Internet communication channels, and push notifications. As recent studies show, the use of redundant communication strategies improves the probability of successful delivery of emergency messages considerably (Okafor & Adeniran, 2024;

Moreover, the system has a lot of use for security agencies, health professionals, and law enforcement organizations as well. The system provides real-time information about the location of an individual, along with the context information like sound recordings and activity pattern which will improve the situational awareness and result in better decision-making during an emergency situation (Al Mudawi, N., Azmat, U., Alazeb, A., Alhasson, H. F., Alabdullah, B., Rahman, H., Liu, H., & Jalal, A. (2025).; Zhong, Z., & Liu, B. (2025)).

1.4 Scope of the Study

The current project is focused on the creation of the mobile security system that operates on the Android platform and ensures constant monitoring, automatic detection and quick

response in case of emergency situations. This system should work primarily on smartphones through the use of sensors, artificial intelligence, and automatic communication. In particular, the project will have such major functions as:

1. **Continuous sensor data collection and preprocessing:** Real-time sensing refers to sensing that is done continuously using sensors such as accelerometer, gyroscope, microphone, and camera available in the smartphones. This makes sure that any abnormal movement or sound pattern will be detected by the system. The preprocessing of data helps in eliminating noise from the data.

2. **AI-driven danger detection:** The present project is concentrated on developing the mobile security system which will work in the environment of the Android platform and guarantee continuous monitoring, detection, and immediate response to emergencies.

3. **Automated SOS alert transmission:** Once the emergency alert is raised, the system will automatically send out SOS alerts to predefined emergency contacts without relying on any form of manual input.

4. **Real-time GPS location tracking:** It constantly keeps track of the user's geographic position, thereby supplying precise and updated geographical details to those responding to the call. This feature proves vital when there is a need for immediate action during any emergency situation like kidnapping, accident, or a health emergency.

5. **Audio and location recording:** This system has the capability to record whatever is happening in the surroundings of a person and where he or she is at the time something wrong happens. This is done to help people get an understanding of the situation and to analyze it later on. The recording of audio and location takes place when there is an emergency.

6. **Backend services for routing and event logging:** The server or cloud-based services are used in the system to route the alert message in the event of an emergency. This is accompanied by the storage and recording of information about the occurrence of the event.

1.5 Limitations of the Study

The research recognizes certain limitations which can hamper the full functionality of the system, despite the benefits of the latter in terms of ensuring personal safety and emergency management:

1) **Quality of sensors and availability of data:** Precision is influenced by the smartphone sensors (accelerometer, gyroscope, microphone, camera). The poor quality of sensors and lack of data can decrease the accuracy of detection, leading to the false alarm or inability to raise an alert (Okeke & Adeyemi, 2024).

2) **Increased battery consumption:** Continuous data collection and analysis via artificial intelligence consume more energy than standard operations, thus requiring additional optimization and sufficient battery capacity to ensure longer usage.

3) **Requirement for stable network connection:** Internet-based alerts presuppose the reliable internet connection. Lack of such connection can result in ineffective alerts.

4) **Restrictions of mobile OS**: The modern operating systems of smartphones have restrictions on background operations in order to preserve system resources and ensure users' privacy (Mohsin, A. S. M., & Muyeed, M. A. (2024)).

5) **Privacy issues:** Collection of audio data, image data and location data implies potential breaches of privacy and necessity to comply with the rules of data collection and storage (Ibrahim et al., 2024; Adebola & Yusuf, 2025).

1.7 Definition of Terms

1. **Danger Analyzer:** Artificial intelligence system responsible for analyzing data from sensors to detect any dangers or unusual events, through machine learning.

2. **SOS Messenger:** The component responsible for generating and sending an alert about danger or emergency to contacts, security agencies, and other predefined receivers.

3. **Sensor Fusion:** Merging of different data coming from various sensors like accelerometers, gyroscopes, GPS, microphones to increase detection accuracy.

4. **Anomaly Detection**: Detecting any deviation in the usual behavior of the person by analyzing data collected from sensors, which might be indicative of emergencies.

5. Real-time Monitoring: Collection, processing, and analyzing of data coming from sensors and environment in real-time to detect any emergencies and threats.

6. Automated Alert Transmission: Sending notifications and evidence to emergency contacts through several mediums automatically and without any human interaction.

7. Contextual Evidence Collection: Automatic recording of data related to the context like audio, images, or activities in case of emergencies.

8. Proactive Threat Prediction: The capability of the AI to predict emergencies based on the analysis of behavior and sensor data trends.

9. Emergency Response Integration: The capability to work with other systems and agencies in order to ensure proper response after receiving alerts.

10. Mobile-based Safety System: Personal safety and security application based on a smartphone.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

The increase in demand for intelligence in personal security leads to mobile emergency detection and automatic alerting systems. The usage of smartphones as intelligent sensors allows detection, monitoring, and immediate alerts, yet most of the systems are reactive, triggered-based, and lack predictive capability making their emergencies less effective (Okeke & Adeyemi, 2024; Zhong & Liu, 2025).

The current chapter analyzes recent studies, papers, and reports related to mobile safety systems, AI anomaly detection, sensor fusion, automated alerts, and evidence collection. Emphasis is given to efficiency of the current approach including development of emergency alerts, real-time monitoring using phone sensors, AI threat detection, and multi-channel alerts. Detected shortcomings become a foundation for the suggested research, which involves an autonomous AI-powered system that identifies threats, provides automatic alerts, and evidence collection. That is why an Intelligent Mobile-Based Emergency Detection and Automated SOS System has to be developed (Okafor & Adeniran, 2024; Al Mudawi et al., 2025).).).

2.2 Theoretical Background

Designing and developing an Intelligent Mobile-Based Emergency Detection and Automated SOS System requires a combination of several important theoretical approaches, principles, and technological ideas borrowed from mobile computing, artificial intelligence, sensor technology, and human-computer interaction. An insight into the theoretical background provides a better understanding of the mechanism, rationale behind various decisions, and integration of technologies for detecting emergencies automatically and reliably. In particular, the discussed system combines several theories, principles, and other conceptual aspects.

First of all, the system makes use of smartphone sensors, which are responsible for tracking user activities and conditions of environment in order to detect anomalies. Such sensors generate a stream of data related to movements, orientations, acoustics, and geolocation that are vital for determining whether there is some sort of abnormality and danger to a user. The idea of sensor fusion allows collecting and analyzing all such data in an intelligent way in order to increase the accuracy of detection and decrease the number of false positives.

Second, artificial intelligence (AI) and machine learning (ML) act as the mechanism of decision making. With help of ML, the system learns normal patterns of user behavior and detects anomalies to predict emergencies before they happen. It is worth noting that such an approach differs significantly from traditional safety applications which typically have no proactive nature and do not provide any prediction of future emergencies.

Third, the system includes automated communication and alerting mechanisms, which make sure that the notification is sent to appropriate persons (e.g., emergency contacts or security companies). Using multi-channel alerting solutions, the system eliminates the possibility of problems associated with communication networks and devices, thus increasing the probability of emergency response.

Finally, the last part of the system involves capturing contextual evidence such as sounds, images, and logs about the incident. In this way, emergency response teams and law enforcement would be able to understand what happened and make sure of the truthfulness of their reports.

All these aspects of the theoretical background – sensors, anomaly detection using AI, automated communication, and evidence collection – allow us designing and implementing a holistic, intelligent, and proactive safety solution.

2.2.1 Mobile-Based Emergency Detection Systems – Overview

The use of mobile devices for the detection of emergencies is one of the essential security technologies in our time. Traditional approaches (manually triggered alarms, phone calls, panic buttons), which require user activity, are prone to failure in cases of violent actions, incidents, and health problems in case there is an existing technology (Zhong & Liu, 2025). The smartphone is an all-in-one device capable of monitoring, analyzing and automatically triggering alerts through accelerometer, gyroscope, GPS, microphone, and camera sensors without the need for additional gadgets (Okeke & Adeyemi, 2024; Mohsin & Muyeed, 2024).

Recent studies focus on predictive security technologies based on artificial intelligence, which are aimed at detecting anomalies in order to recognize any danger independently from user activity (Kundu et al.). Sensor fusion is the basis of current research (Sakalauskas & Vaiciulyte, 2025). Current mobile security applications are able to detect the location of the smartphone and its owner for the selected organizations and people. Various channels of communication (SMS, push notifications, Internet messaging) make

it possible to provide effective information transfer in cases of limited network coverage (Okafor & Adeniran, 2024).

2.2.2 Smartphone Sensors and Data Acquisition

The modern smartphone uses sensors such as an accelerometer, gyroscope, GPS, microphone, and camera, which give information about user activities and surroundings. The sensor theory tells about the processes through which devices gather information about movements, orientations, and sounds. Such information allows the creation of safety applications that can analyze activities, find threats, and react to them (John et al., 2024). Sensor fusion refers to the process when data from various sensors is collected in order to improve accuracy and lower the chances of mistakes during detection of any danger (Zhong & Liu, 2025). Motion sensors detect activities and movements of users and track their motion or velocity; accelerometers detect acceleration while gyroscopes detect rotations. GPS detects user location in order to locate him/her if there is a need for help. Microphones and cameras help to evaluate the environment, find distress signals, and understand the surroundings, but information processing may pose challenges. Sensor fusion helps to solve the problem of noise and inaccuracy of single sensors, but problems such as high power consumption, noise, and different performances of devices remain.

2.2.3 Artificial Intelligence and Anomaly Detection

Artificial Intelligence (AI), especially machine learning, plays a vital role in processing the gathered information and detecting any deviation from the norm, which may lead to an anomaly or potential threat. The process of identifying deviations in order to detect potential danger is known as anomaly detection. Through training on the past datasets, the model learns user behavior and detects the deviations that allow initiating preventive actions without the involvement of humans (Kundu, S., Mallik, M., Saha, J., & Chowdhury, C. ; Adeyemi et al., 2025).

Machine learning is one of the key areas in the field of AI, which allows systems to learn from data rather than programming them entirely. Machine learning is divided into supervised, unsupervised, and reinforcement learning. In the case of supervised learning, datasets used for training have labels that allow finding out patterns. Unsupervised learning is used to find patterns or abnormalities in the absence of labeled datasets. It is particularly important in emergency detection as it finds abnormalities regardless of predefined scenarios.

2.2.4 Automated Alerts and Communication Systems

Automated alerts are necessary for proper dissemination of information after the detection of an emergency, particularly in safety applications, where automated alerts are triggered upon the detection of harm, distress, or inability to alert others (Okafor & Adeniran, 2024). Event-driven communications collect location, time, and context and forward this information to contacts or emergency services.

The important development is the multi-channel delivery via SMS, internet messaging, push notifications, and e-mail in order to deliver messages through all channels regardless of failures in some channels. This method increases speed and delivery probability (Mohsin, A. S. M., & Muyeed, M. A. (2024)). The backend service provides routing, queuing, logging, and data synchronization (Al Mudawi, N., Azmat, U., Alazeb, A., Alhasson, H. F., Alabdullah, B., Rahman, H., Liu, H., & Jalal, A. (2025).).

Problems include network problems, device limitations, and delivery restrictions. In case of poor network, messages may be delayed or blocked. Channels are prioritized and attempts of message transmission are done dynamically.

2.2.5 Evidence Collection and Contextual Data

The collection of contextual data is increasingly important for the rapid response and analysis. Contextual evidence includes voice data, imagery, timestamps, and system logs, which provide not only notification but also additional information (Al Mudawi, N., Azmat, U., Alazeb, A., Alhasson, H. F., Alabdullah, B., Rahman, H., Liu, H., & Jalal, A. (2025).; Zhong, Z., & Liu, B. (2025)).

Collection of contextual data allows increasing situational awareness: voice data provides distress signals or environmental noises; imagery provides information about

the surrounding environment, attackers, or condition of the person being attacked. This data helps to evaluate situation and make decisions rapidly.

Contextual data is also important in order to investigate incident after it occurs. Recorded data can be used for evidentiary purposes, whereas activity logs provide a timeline of the incident.

2.3 Review of Related Works

Researches in the area of mobile applications with sensor-based detection of threats aim at improving personal safety. For instance, Motion-sensor-based and GPS-based emergency notifications tend to be quicker than manual notifications, yet the threshold-based method is prone to misidentifying regular behavior as an emergency.

(Zhou, H., Zhang, X., Feng, Y., Zhang, T., & Xiong, L. (2025)) suggest usage of AI in safety monitoring with machine learning application to detection of changes in user behavior. The supervised learning algorithms can identify various emergencies but require a large amount of labeled data for training and work poorly in variable environment where behavior changes. Multi-channel emergency communications use SMS, app-based notifications, and cloud-based delivery to enhance reliability on poor networks. However, the systems typically offer only delivery services without any automatic threat detection; this is where this project comes in. The context-aware emergency system by Ibrahim et al. (2025) is audio/video-based, thus involving privacy issues and stringent access rights. It does not have any predictive capacity as it depends entirely on user activation.

Community based mobile apps for safety typically send a user's location to nearby contacts and authorities during an emergency situation, however, most of them require manual operation and lack automated threat detection.

(Kundu, S., Mallik, M., Saha, J., & Chowdhury, C. (2024)) investigated machine learning algorithms to detect activity and fall in smartphone users using Android smartphone sensors. Machine learning algorithms applied were SVM, RF, Naive Bayes, and KNN, while RF and SVM proved to be the most effective. The development of multimodal fusion system by (John, A., Cardiff, B., & John, D. (2024)) for public danger monitoring is through fusion of behavioral analysis and speech emotion detection. The authors noted that fusion of many sources of information surpasses single-sensor systems in detection of abnormal behaviors. Nevertheless, the system is designed for public surveillance and does not have an automatic alarm through mobile phones. While improvements have been made to sensor-based monitoring, AI-based emergency detection, improved communication and data gathering, such innovations have not

covered all areas of concern. These include user activation, prediction capability, and data fusion.

2.4 Summary of Literature Review

Table 2.1. Summary of Literature Review

S/N	Author(s) & Year	Objective	Methodology	Key Findings	Research Gap
1	Al-Risi et al. (2024)	Review of smartphone-based fall detection among disabled and elderly populations.	Comprehensive review on mobile sensors and machine learning.	Mobile sensors and machine learning improve accuracy of fall detection.	The paper is a review; there is no emergency system in place.
2	Kundu, S., Mallik, M., Saha, J., & Chowdhury, C. (2024)	HAR and fall classification algorithms based on smartphones were designed.	Random Forest, SVM, KNN, and Naïve Bayes classifiers were compared with respect to accelerometer and gyroscope signals.	Random Forest Classification results in terms of performance and efficiency.	Only activity classification was considered in this research, no SOS alarm generation or cloud management.
3	Wang et al. (2024)	A real-time framework of fall detection using smartphones along with secondary validation through Wi-Fi has been developed.	This involves the use of IMUs of smartphones together with Wi-Fi CSI and deep learning.	It improves the accuracy of fall detection as well as minimizes false positives.	The drawback of the system is that extra Wi-Fi facilities have to be added.
4	Suffoletto et al. (2025)	Fall-risk predictions based on mobility data from smartphones	Predictive modelling using mobility data	Mobility data from smartphones enhanced fall-	Focus on fall-risk prediction, not emergency responses.

S/N	Author(s) & Year	Objective	Methodology	Key Findings	Research Gap
		after discharge from the ED	from smartphones.	risk prediction compared to traditional screening methods.	
5	Nguyen et al. (2025)	Enhancement of Random Forest-based fall detector by genetic algorithm hyperparameter tuning.	Optimally-tuned RF algorithm for mobile sensors-based dataset.	Hyperparameter tuning improved the precision of the algorithm.	The algorithm detects falls only and does not incorporate any emergency response modules.
6	Guo & Nakayama (2025)	Proposed enhanced technique for feature extraction of smartphone fall detection.	Machine learning features extracted from sensors on a smartphone.	Selected features increased efficiency in classification with computational efficiency.	Does not consider emergency alerts or intelligence systems.
7	Goodarzi et al. (2025)	Developed a light-weight mobile fall-detection model for elderly care.	Utilized activity recognition using neural networks from smartphones' accelerometers.	Provided accurate fall detection in real-time applicable to mobile devices.	Fall detection only, no multi-modal emergency detection.
8	Frontiers in Artificial Intelligence (2024)	Studied artificial intelligence models for activity recognition in wearables.	Comparison was done between CNN, SVM, Random Forest, and KNN.	Random Forest combined efficiency with accuracy.	Only concerned about activity recognition, not full emergency response.

S/N	Author(s) & Year	Objective	Methodology	Key Findings	Research Gap
9	Frontiers in Bioengineering and Biotechnology (2025)	Presented a framework for multimodal intelligent biosensor to detect falls and monitor health.	Multimodal sensor integration with deep learning techniques.	The combination of modalities greatly increased reliability of fall detection.	Health monitoring application-oriented framework without SOS from smartphone.
10	González-Aguirre et al. (2025)	Proposed a privacy-preserving multimodal fall detection and alerting system.	Thermal camera, wearable sensors, voice assistant, and machine learning.	Multimodal approach decreased false alarms but maintained privacy.	Needs special sensing hardware; not a simple smartphone solution.

2.5 Research Gap

Some advances have been made in the area of intelligent emergency detection through the use of smartphones, ML algorithms, and sensor based activity recognition for emergency detection, activity recognition, communication and multimodal sensors. ML approaches such as Random Forests and Deep learning algorithms improve activity recognition accuracy. However, most of the literature is devoted to the processes of managing the emergency itself. Most of the applications rely on manual SOS activation that will not work in case if the person becomes unconscious. Multimodal sensors can improve performance; however, they may require additional hardware or complex algorithms that cannot be used continuously in mobile devices. Very few works consider combination of light-weight ML algorithms, continuous sensing, automatic SOS activation, and mobile offline/on-line data processing. Moreover, very few papers consider the features such as distress voice, movement abnormalities, car accidents and multimodal sensors fusion. The proposed solution tries to fill this gap.

CHAPTER THREE

SYSTEM ANALYSIS

3.0 Introduction

This chapter explains the methodology, system analysis, and design used in the Development of an Intelligent Mobile-Based Emergency Detection and Automated SOS System. It covers both technical and theoretical aspects and highlights the systematic approach used to meet the aims and objectives of Chapter One.

It begins by reviewing existing systems and their limitations, then sets objectives for the new system. Then it outlines the system requirements, discusses the design methodology, and explains system modeling with UML diagrams. The chapter concludes by presenting the system design, which covers inputs, outputs, the database, and the overall architecture. According to a recent article by Hichan Moon, Hyosoon Park, and Jiwon Seo (August 2024), the HELPS system is a novel approach to positioning and searching for emergency location services that works for all mobile phone users, including those with older feature phones, supporting the system's technical credibility.

3.1 System Analysis

System analysis is very important in the process of system development as it analyzes the problem environment, specifies the requirements and defines the objectives for the development of the system. System analysis aims at analyzing the existing systems, evaluating their strengths and weaknesses, and defining the key problems that the system being developed aims at solving (Pressman, R. S., & Maxim, B. R. (2020).). There are many mobile emergency alerting systems currently available. Such technologies enable individuals to send out emergency alerts to their predefined contacts or to the police whenever they are in a risky situation. Modern smartphones have panic button or SOS functionality through which emergency alerts along with location can be sent.

3.1.1 Analysis of the Existing System

Mobile technologies have been integrated into emergency response systems. Currently, most of the mobile phones have inbuilt emergency alerts. Many security apps allow users to initiate distress alerts in cases of hazards.

Most of the alerts have to be manually initiated. Alert is initiated through pressing a panic button, gesture (like shaking phone) or pressing a certain sequence of keys. The predefined message containing location is sent out to the concerned person (Mohsin, A. S. M., & Muyeed, M. A. (2024)).

Primary users:
i. Mobile users initiating alerts in case of danger.
ii. Emergency contacts receiving alerts.
iii. Security agencies or the responder based on the configuration of the system.

Process: users enroll into the application and configure emergency contacts. After configuring, application runs in the background until alert is activated. They are mostly reactive, in that they depend on user action in order to initiate alerts. They depend on manual activation and basic intelligent technology. This makes them to lack any form of proactive safety monitoring or automatic initiation of emergencies.

3.1.2 Limitations of the Existing System

Although these apps are becoming more common in recent years, there are a number of limitations which prevent their efficient operation during emergencies and make it necessary to create an intelligent automatic detection system. Some of the limitations of these apps include the following:

1. **Manually activated alerts:** The activation involves pressing a button or performing a particular gesture, which becomes impossible if a user is tied up, unconscious, hurt, or scared, thereby resulting in delayed communication to the emergency contacts.

2. **No intelligent processing of data:** Though phones have many sensors (accelerometer, gyroscope, GPS, microphone) to detect any abnormalities, most apps do not involve automatic data processing or machine learning .

3. **Inadequate data for evidence:** The apps send limited data (location) and a text message only and fail to provide scene information (voice message, pictures, environmental data) for situation analysis.

4. **Installation problems:** Some apps are not easily installed; some apps need an Internet connection or subscription, but that is not possible in the absence of a good network. Our app intends to be easy-to-use, easy-to-install, works well even without Internet, and helps users with disabilities (Barriers and best practices for inclusive emergency alerts and warnings, 2025).

3.1.3 Objectives of the Proposed System

To address the main aim of the proposed system and achieve the required results, the following specific objectives have been set up; each of them aims to overcome one or several limitations of existing systems:

1. **Research and analyze existing alerting systems**

It is necessary to conduct research because most of them have serious deficiencies related to the absence of automation, slow reactions, and the absence of intelligent capabilities. The analysis helps to determine the best practices and avoid mistakes to get rid of inefficiency and implement innovation.

2. **Analyze data from sensors to detect abnormal situations**

The existing systems do not use sensors enough; therefore, detection may become inaccurate or impossible. It is possible to use accelerometer, gyroscope, microphone, and GPS data to monitor behavior and detect such events as falls or distress sounds.

3. **Design an intelligent danger detection model based on AI**

There is no intelligent and automated danger assessment in current systems. This solution allows classifying the sensor data as safe or dangerous and conducting the automated danger detection and reaction to it.

4. **Develop a mobile application for constant monitoring**

It is possible that some existing systems work only when activated by the user. The developed mobile application works constantly and monitors user activity; therefore, the emergency alarm is triggered independently of user activity.

5. **Implement the capability of automated sending of SOS messages**

Existing systems suffer from delays caused by the manual activation of alarms. The system will automatically send distress messages to contacts defined by the user after the danger detection.

7. Evaluate the system performance

Real-life evaluation is often not conducted; therefore, the reliability of the systems is put at risk. The tests will be conducted for simulated emergency situations.

3.1.4 Analysis of the Proposed System

The proposed Intelligent Mobile-Based Emergency Detection and Automated SOS System is a highly advanced system designed for the purpose of providing safety using a smartphone. While many emergency alert systems rely on manual activation by a user to raise an alarm (Emergency Alert System, 2023), the proposed solution utilizes an alternative approach where sensors (accelerometer, gyroscope, microphone, GPS) and artificial intelligence techniques (neural networks, decision trees), as well as communication protocols, are used for automatic detection and sending a notification in case of emergency situations.

Continuous monitoring of activities carried out by a user is performed with the help of a set of sensors, and corresponding data on movement, orientation, sounds around a person, and location are recorded. These data are further analyzed with the help of machine learning algorithms and rule-based logical analysis to determine whether there is something unusual (abrupt falling down, unusual activity, cries for help, or change in location), which could indicate an emergency situation.

The detection process might be affected by the difficulty in determining what situations could be regarded as an actual emergency and not just some other extraordinary events, which would lead to false alerts. Another problem might arise due to sensor noises, variability of user behavior, and quality of training data. Thus, calibration of models would be necessary.

Upon detecting the threat, the system automatically sends an alert without any input from a user. This alert contains information on real-time location, message, and possibly additional information like audio recordings or images to designated emergency contacts or simulated security agencies.

3.2 System Requirements Specification (SRS)

System Requirements Specification (SRS) provides detailed descriptions of the functional and non-functional requirements of the proposed system. This is a detailed description of what the system can do and what performance standards it can provide under varied conditions. The document is fundamental in guiding subsequent system designs and evaluations. As regards the above-mentioned proposal, “Development of an Intelligent Mobile-Based Emergency Detection and Automated SOS System,” the requirements will be split into functional and non-functional requirements. Functional

requirements are descriptions of the functionality of the proposed system while non-functional requirements are descriptions of qualities that the system needs to meet. Describing the requirements helps in matching user expectations and operating efficiently in emergency situations (Pressman, R. S., & Maxim, B. R. (2020).; Almeida et al., 2024).

3.2.1 Functional Requirements

Functional requirements specify the basic functionalities that the system will undertake to fulfill its objectives. The following are some of the functionalities describing the interactions, processing, and reactions of the system.

1. **Continuous Sensor Data Monitoring:** The system will continuously monitor data from sensors found in smartphones. **S**ensors **such as** Accelerometer**,** Gyroscope**,** Microphone**, and** GPS will be monitored in real time**.** This functionality helps identify abnormal patterns indicating the possibility of any dangers .

2. **Abnormal Activity Identification:** The system will analyze the sensor data to determine if there are any abnormal or risky activities. Examples of such abnormal or risky activities **are;** Falls, Movement activity abnormali**ty,** Presence of noise relating to distress **and** Changes in l**o**catio**n.** The above functionality allows the automatic identification of emergency situations without the need for any user intervention (Mohsin, A. S. M., & Muyeed, M. A. (2024)).

3. **Intelligent Danger Classification Using AI:** The system will use a machine learning approach to classify the users' activities**,** based on their state **of being, as** Safe**,** Suspicious **or** Danger**.** This classification functionality will help in decision making based on different danger states (Sakalauskas, L., & Vaiciulyte, I. (2025); Kundu, S., Mallik, M., Saha, J., & Chowdhury, C.)**.**

4. **Automatic SOS Triggering:** The system will automatically trigger an SOS signal in case of a dangerous situation. This reduces the need for users to manually activate an alert in case of emergencies especially when they are unconscious.

5. **Real-Time Location Tracking**: The system will capture and send real-time GPS location of the user during an emergency situation.

6. **Emergency Message Communication:** The system will send alert messages to predefined emergency contacts. The messages should have the **c**urrent location of the user**, the t**ime of the event **and s**tatus of the alert**.** Th**is** construction ensures that necessary information is communicated appropriately and effectively.

7. **Additional Evidence Capture and Transmission**: The system will capture and transmit additional information regarding an emergency situation including: Recorded audio data

8. **User Registration and Profile Configuration**
The system will allow users to:

1. Create an account
2. Add and configure emergency contacts
3. System configuration settings

9. Alert Logging and Data Storage

The system shall store records of emergency events, including timestamp, location and sensor data. This allows for post-event analysis and system evaluation.

10. Manual SOS Activation (Optional Feature)

Although the system is primarily automated, it shall provide an optional manual SOS trigger as a backup. This ensures redundancy in cases where automatic detection fails.

3.2.2 Non-Functional Requirements

Non-functional requirements define the quality properties of the system, which describe the effectiveness of the system's performance in certain situations.

1. Performance Requirements: The system should operate in real-time and not have any delays in detecting danger and sending an alert.

i. Response time of SOS alert: ≤ 5 seconds (Moeller et al., 2024, pp. 135–138)
ii. Sensor work without any delay

Performance requirements are crucial in emergency systems because delays can lead to serious consequences (Mohsin, A. S. M., & Muyeed, M. A. (2024)).

2. Reliability Requirements: The system should have no issues or crashes while functioning.

i. Constant monitoring feature
ii. Reliable alert delivery

Reliability provides assurance of system performance in critical situations.

3. Usability Requirements: The system should have an easily understandable and intuitive interface.

i. Intuitive navigation
ii. User interaction is kept at a minimum
iii. Notifications of alerts

Usability requirements are vital because users may find themselves in stressful situations when using the system.

4. Security Requirements: The system should protect sensitive data about the user, including:

i. Geolocation
ii. Personal information
iii. Emergency contacts

The measures include but are not limited to:

i. Encryption of data
ii. Authentication security
iii. Access control

Security protection of user information is necessary to prevent the misuse and violation of privacy .

5. Scalability Requirements: The system should allow further development in the future, such as:

i. Integration with wearable technology
ii. Installation of additional sensors
iii. Increase in the number of users

Scalability allows the further development of the system in the future.

6. Availability Requirements: The system should be constantly available, especially in emergencies.

i. 24/7 monitoring system
ii. Ability to operate in background

7. Maintainability Requirements: The system should be easy to update.

i. Modularity
ii. Easy to read codebase
iii. Easier debugging and upgrade

8. Compatibility Requirements: The system should be compatible with:

i. Android OS
ii. Diverse phone hardware configuration

3.3 System Design Methodology

System design methodology refers to the systematic approach used in modeling, design, and implementation of the suggested system. System design methodology enables the representation of system components and the way they interact. It also provides guidance on the way the implementation should be done. Choosing the appropriate design approach is very important in ensuring that the system is well structured and can meet all its requirements. In the development of the proposed system "DEVELOPMENT OF AN INTELLIGENT MOBILE-BASED EMERGENCY DETECTION AND AUTOMATED SOS SYSTEM," two different approaches have been used:

i. Object-oriented analysis and design (OOAD) for modeling of the system
ii. Iterative Development Model for the implementation of the system

A. Object-Oriented Analysis and Design (OOAD): The design process employs the principles of OODA (Object-Oriented Analysis and Design) to depict the system as a set of interacting objects, which represent tangible entities with data and operations (Pressman, R. S., & Maxim, B. R. (2020).). In accordance with OOA&D, the system will be structured based on the entities used in emergency detection and response:

1) User Object: contains user information and emergency contacts.

2) Sensor Data Object: contains data generated by the sensors (accelerometer, gyroscope, microphone, GPS) of smartphones.

3) Danger Detection Object: performs analysis of the data using logic rules and ML.

4) SOS Manager Object: generates and triggers an emergency alert.

5) Notification Service Object: sends alerts either to the contacts of emergency services or simulated organizations.

B. Iterative Development Model as the Implementation Approach

Aside from modeling, the Iterative Development Model was also used as the implementation approach of the system. This method develops the system incrementally through planning, design, implementation and testing in every stage of development. Project iterations included:

1) System Design and Interface Development: design of the overall architecture and interface of the system.

2) Sensor Integration Module: data acquisition from the accelerometer, gyroscope, microphone and GPS sensors.

3) Danger Detection Module: abnormal behavior detection using rule based and artificial intelligence techniques.

4) SOS Alert Module: automated alert generation and sending of SOS alerts.

5) System Testing and Optimization: performance testing under simulated emergencies and detection accuracy refinement.

3.3.1 Methodology Justification

The choice of OOAD and the Iterative Development Model were due to:

1. Proper System Representation: OOAD represents the system by breaking down the whole system into objects that represent real world entities. Entities such as user profiles, sensor data handlers, danger detection, SOS service and other modules can be represented as objects.

2. Modularity and Maintainability: OOAD achieves modularity through independent modules performing separate functions. In case of any change in any of the modules, there will be no impact on any of the modules (danger detection will not impact SOS alert or user interface). The significance of this is the requirement for frequent modifications in AI-based danger detection.

3. Scalability and Possible Future Developments: Some of the future developments in the system may involve incorporation of wearable devices, sensors and number of

users. OOAD enables new modules to be added without making changes to the existing modules as technology advances .

4. Iteration and Continuous Improvement (Flexibility): The Iterative Development process involves iteration among design, implementation, testing, and evaluation **to** improve design. Iterations are needed in case of complex modules such as sensors and AI. Danger detection modules can also be improved using test results.

5. Early Detection of Errors: Early detection and correction of errors are possible with continuous testing during development. Emergency detection systems in real time may result in both false positives and negatives. Targeted testing of various modules (including sensors, AI detection system, and SOS communication) greatly minimizes failures (Mohsin, A. S. M., & Muyeed, M. A. (2024)).

6. Ability to Design Complex, Multiple Components: The system involves use of smart phone sensors, AI, real time data analysis, and alerts generation process. OOAD offers design approach through modeling of system components whereas iterative model facilitates systematic development and integration. This helps in management of interdependence and coherent system functioning.

7. UML Compatible System Modeling: OOAD is compatible with UML through provision of diagrams for system structure and behavior. Use case, activity, and class diagrams help in designing and documentation of the system functionalities and interactions. They also help in defending the project since system functioning becomes clear through illustration.

3.3.2 Data Collection Methods

The following describes data collection methods used for design, modeling, and implementation of the suggested system. The data for design, modeling, and implementation of INTELLIGENT MOBILE-BASED EMERGENCY DETECTION AND AUTOMATED SOS SYSTEM have been collected through various methods to ensure validity, reliability, and effectiveness of the system based on primary and secondary sources.

1. Literature Review (Secondary Data)

The bulk of secondary data is based on the review of academic literature: academic journals, conference papers, online materials and technical documentation to map out

existing emergency detection systems, mobile safety applications, and approaches to AI monitoring. Results:

i. Mapping of existing architectures;
ii. Identification of research gaps;
iii. Requirements and objectives definition.

2. Dataset Collection for AI Model Development

To develop the AI danger-detection module, several datasets, as well as synthetic datasets have been created to simulate real-life emergencies (falls, abnormal activities and distress). The diversity of dataset improves detection performance (Sakalauskas, L., & Vaiciulyte, I. (2025); Kundu, S., Mallik, M., Saha, J., & Chowdhury, C.). Some of the data sets used, including:

i. Smartphone sensor data (accelerometer and gyroscope);
ii. Data about human activities;
iii. Emergency scenarios simulation data.

3. Observation Method

This method is used for studying user behavior while interacting with mobile device in emergencies, specifically:

i. Behavior during normal vs abnormal activity;
ii. Common smartphone usage;
iii. Potentially dangerous situations (falling down, panic).

4. System Simulation and Testing Data

System simulation resulted in actual testing data to evaluate the performance of the danger-detection and automatic SOS module. Situations simulated include:

i. Fall detection;
ii. Detection of loud sounds;
iii. Sudden movement patterns.

5. Technical Documentation and Developer Resources

Supplementary data comes from developer resources and technical documentation for Android platform, smartphone sensors API and machine learning frameworks.

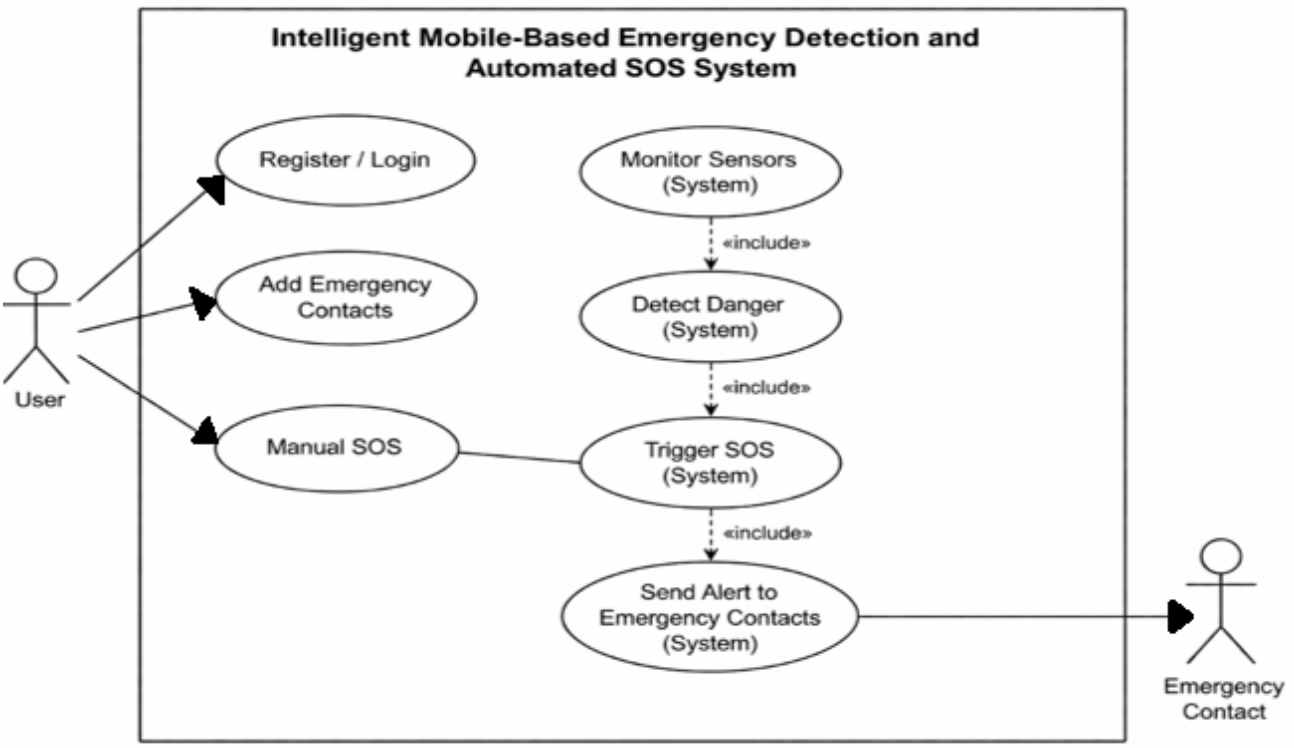
3.4 Modeling of System with UML

UML is the visual language used to model the structure, behavior, and interaction of the components in software. The UML modeling technique enables the OOAD to visualize functionality and architecture of the software.

3.4.1 Use Case Diagram

The use case diagram demonstrates how external actors communicate with the system, with emphasis on the hybrid nature and increased interaction compared to existing emergency systems. For the Intelligent Mobile-Based Emergency Detection and Automated SOS System, the user will be the main actor during initialization and emergency contacts will be secondary actors. After initialization, the system operates automatically. The major use cases in the system include: User Registration, Adding Emergency Contacts, Sensing, Danger Detection, and Automated Sending of SOS Signals. Possible use case diagram:

Fig 3.1: Use Case Diagram

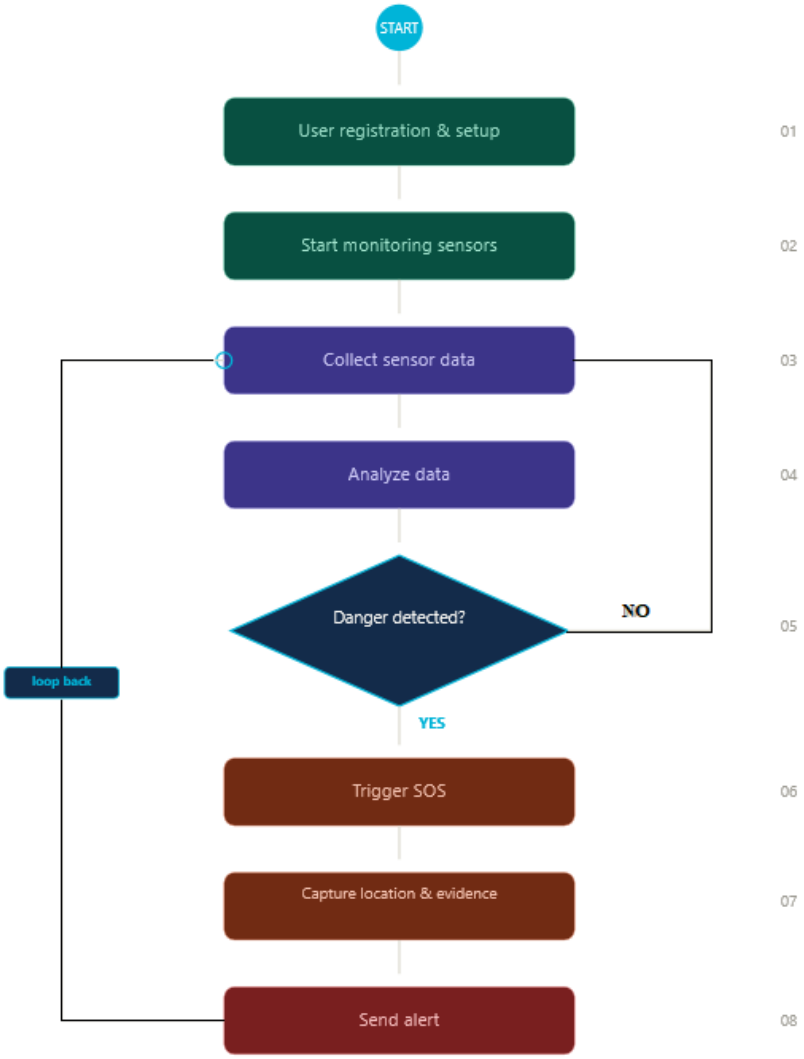


3.4.2 Activity Diagram

This diagram presents a flowchart of how the Intelligent Mobile-Based Emergency Detection and Automated SOS System operates from start to SOS notification.

From the very beginning, when the app starts initializing and background monitoring, the process is continuously carried out to distinguish between normal and abnormal behavior. If no abnormality is observed, then the monitoring proceeds. In case there's any abnormality (voice of distress, fall, unusual movements, or car accident), the problem will be determined using the random forest classifier. Once the confirmation of the emergency is received, it's time to collect evidence (GPS coordinates and available audio/images). The collected information serves as a basis for the generation of an SOS message to send it to emergency contacts via the communication module. After that, the event is recorded in the local SQLite database and the Firebase cloud database.

Fig 3.2: Activity Diagram

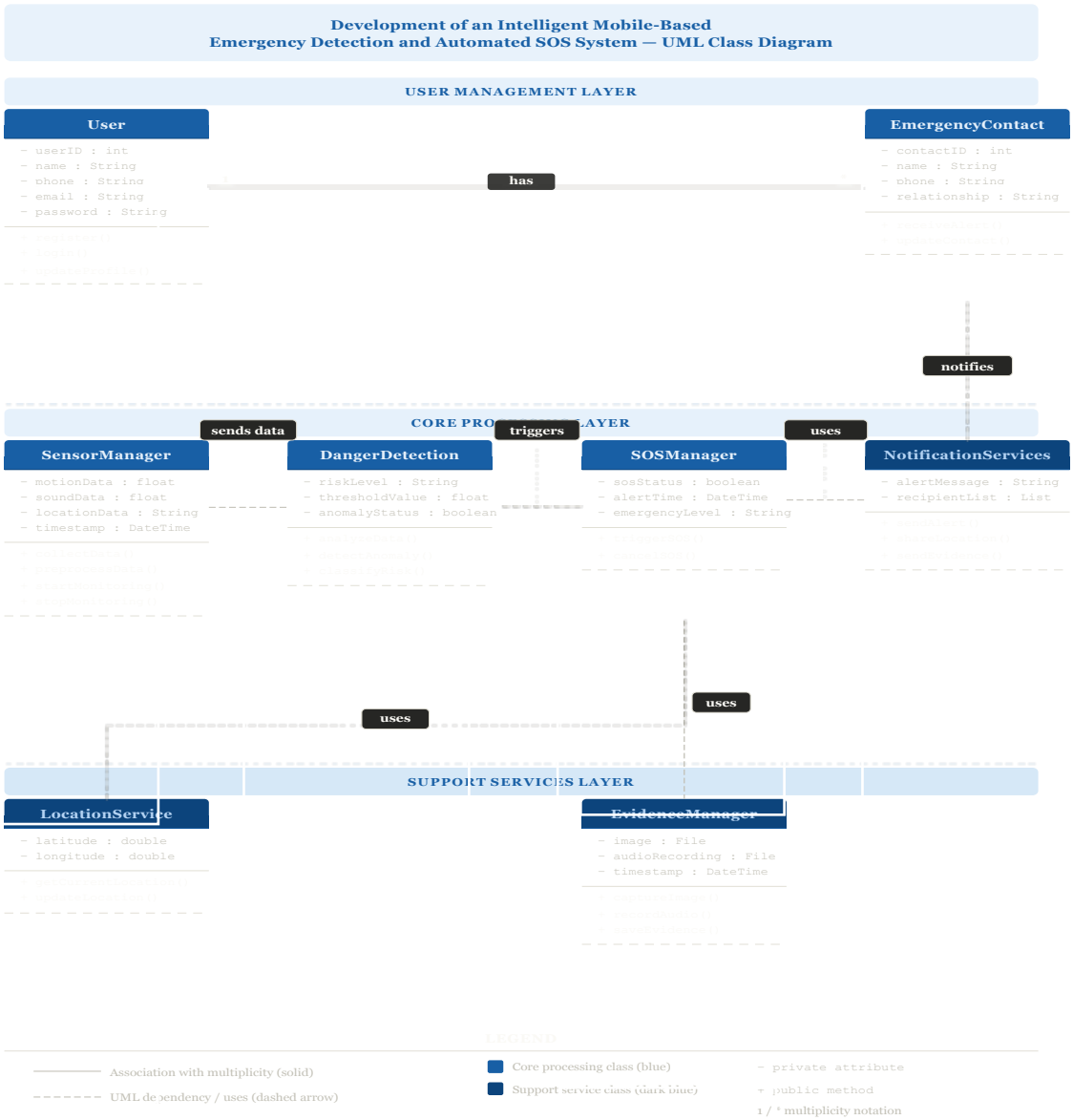


3.4.3 Class Diagram

In the Class Diagram, you will see the architectural design of the Intelligent Mobile-Based Emergency Detection and Automated SOS System that includes the different

classes with attributes and methods. These are the classes used to perform activities related to emergency detection, user management, monitoring of the sensor data, Machine Learning processing, evidence management, and automated alerting of an emergency. The most important classes here include User, Emergency Contact, Sensor Data, Danger Event, Evidence, Alert History, and the Machine Learning Classification Module.

Fig 3.3: Class Diagram



3.5 System Design

In this stage, an actual design will be developed using the requirements and specifications in a way that describes how the system works. For the Intelligent Mobile

Based Emergency Detection and Automated SOS System, the design is based on functional as well as non-functional requirements and includes the use of inputs, processing of data from the sensors, emergency handling, sending messages to emergency contacts and module interconnections.

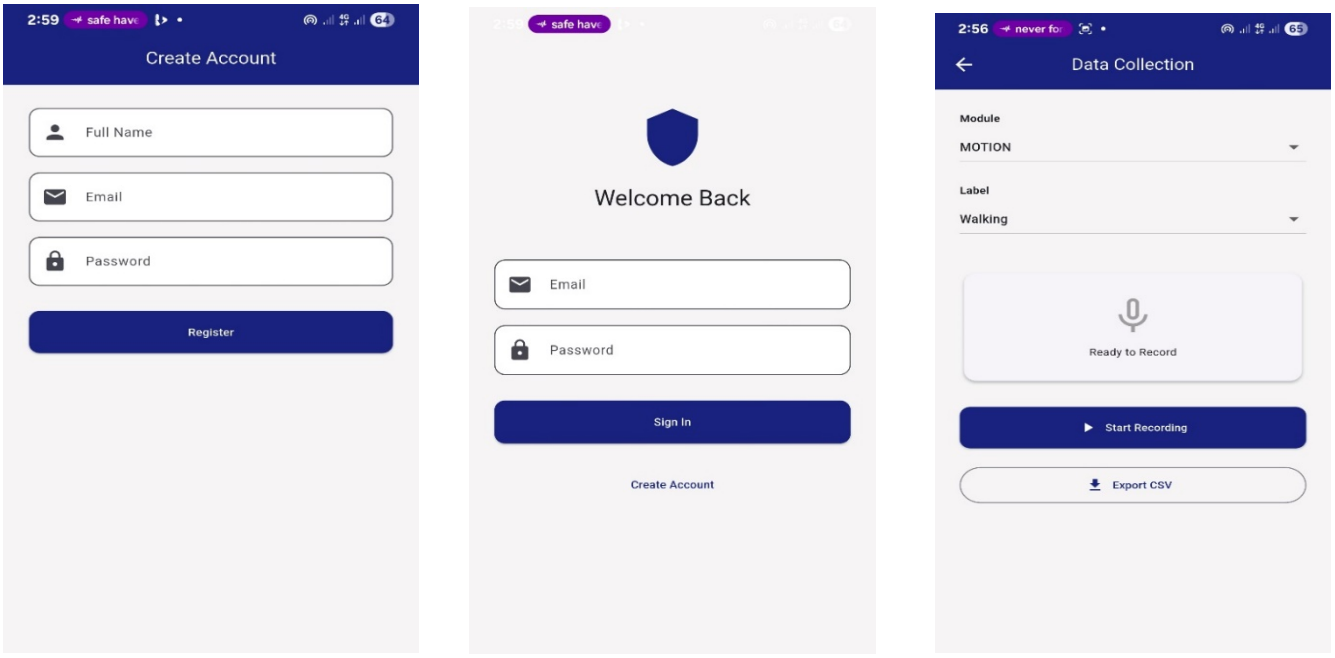
Important Design Components:

i. Input Design
ii. Output Design
iii. Database Design
iv. System Design

3.5.1 Input Design

Input design deals with the way the data is inputted into the system, and it should be done in such a way that it is accurate, efficient, secure, and easy to process. Effective input design makes the system more user-friendly and increases its performance (Pressman, R. S., & Maxim, B. R. (2020).). The Intelligent Mobile-Based Emergency Detection and Automated SOS System will have two kinds of input: manual input from users and automatic sensor data collected from the smartphone. The validation of all the input data will help avoid inaccuracies.

Fig 3.4: Input Design

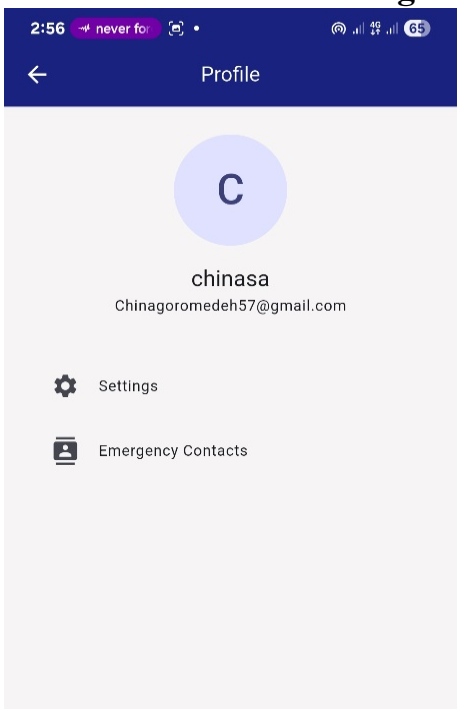


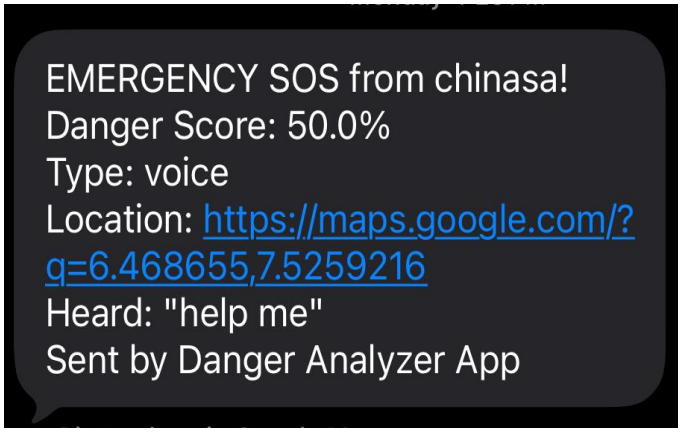
3.5 .2 Output Design

Output design refers to the ways processed information is displayed to users and other external parties. This design seeks to display the information meaningfully, clearly,

accurately, and effectively, thus enhancing decision making and providing important information (Pressman, R. S., & Maxim, B. R. (2020).). In the case of the Intelligent Mobile-Based Emergency Detection and Automated SOS System, the outputs will aid real-time emergency communication, intelligent alerts regarding threats, feedback and monitoring, and reporting in the system. In an emergency scenario, the outputs will be instant, noticeable, and easily comprehensible.

Fig 3.5: Output Design





3.5.3 Database Design

Database design involves organizing, storing, retrieving, and managing of data in order to maintain its consistency, integrity, security, and availability. In the Intelligent Mobile Based Emergency Detection and Automated SOS System, there is a need to store data related to users, sensors, emergencies, and alerts; therefore, the database design is crucial for the system.

3.5.3.1 Hybrid Database Architecture Overview

This architecture incorporates the following storage mechanisms:

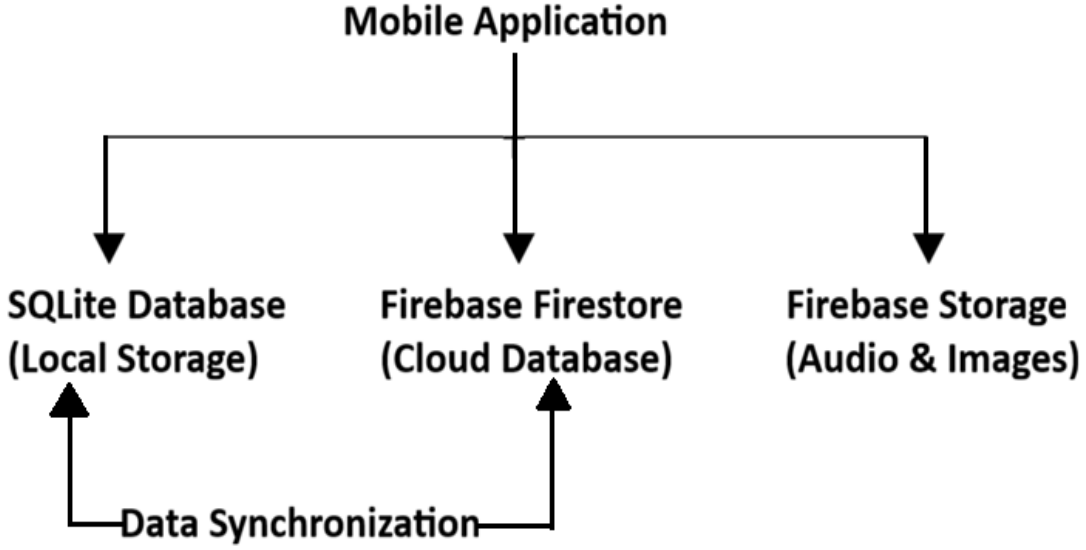
1) SQLite Local Database
2) Firebase Firestore Database
3) Firebase Storage

1) **SQLite:** Offline Android-based data storage for user records, emergency contacts, sensors, and temporary emergency events.

2) **Firebase Firestore:** Remote data storage, cloud storage, and scalability in the future using synchronized records.

3) **Firebase Storage:** Multimedia data (audios and images) storage from emergencies. Firestore stores the reference and metadata and actual files in the Firebase Storage.

Figure 3.6: Hybrid Database Architecture of the Proposed System



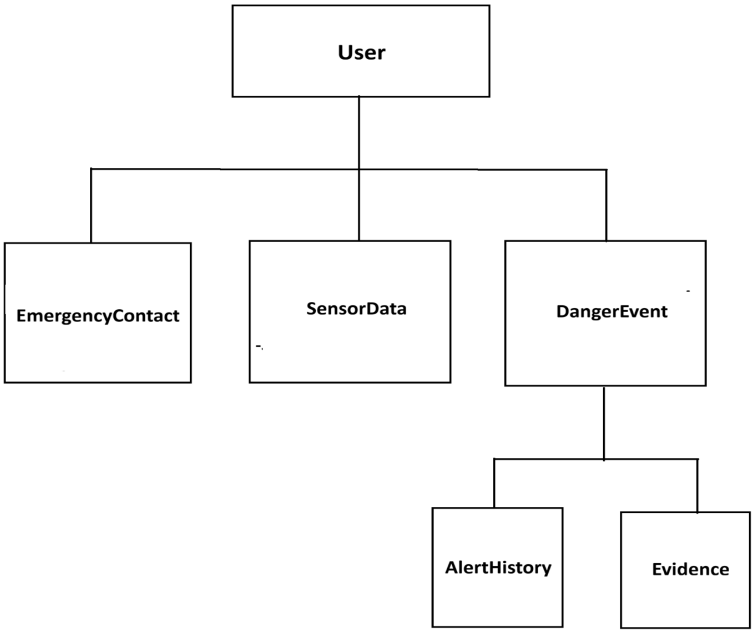
3.5.3.2 Entity-Relationship Modeling (ERD)

Entity-Relationship Diagram (ERD) provides a conceptual representation of data entities in the system along with their relationships. The main entities identified for our proposed system are:

1. User
2. Emergency Contact
3. Sensor Data
4. Danger Event
5. Alert History
6. Evidence

These entities provide support in storing and managing information related to emergency monitoring. The conceptual relationships between these entities can be given as:

Fig 3.7: Entity Relationship Diagram



3.5.3.3 Relational Schema (SQLite Database)

The SQLite database employs a relational structure with interconnected tables. Every table consists of attributes that describe the type of data stored in the system.

User Table: The User table contains data about the registered users of the application.

Table 3.1 User Entity Schema

Field Name	Data-Type	Key
userID	INTEGER(10)	Primary Key
fullName	TEXT(100)	
email	TEXT(100)	Unique
phoneNumber	NUMBER(11)	
passwordHash	TEXT(10)	
registrationDate	DATETIME(10)	

EmergencyContact Table: Holds emergency contacts associated with users.

Table 3.2 Emergency Contact Entity Schema

Field Name	Data Type	Key
contactID	INTEGER(10)	Primary Key
userID	INTEGER(10)	Foreign Key

contactName	TEXT(100)	
relationship	TEXT(50)	
phoneNumber	NUMBER(11)	

SensorData Table: Holds sensor data from smartphones.

Table 3.3 Alert History Entity Schema

Field Name	**Data Type**	**Key**
sensorID	INTEGER(10)	Primary Key
userID	INTEGER(10)	Foreign Key
sensorType	TEXT(50)	
sensorValue	REAL(10)	
timestamp	DATETIME(10)	

DangerEvent Table: Used for storing emergencies detected by the AI module.

Table 3.4 DangerEvent Entity Schema

Field Name	**Data Type**	**Key**
eventID	INTEGER(10)	Primary Key
userID	INTEGER(10)	Foreign Key
dangerType	TEXT(50)	
confidenceScore	REAL(10)	
eventTime	DATETIME(10)	
Status	TEXT(10)	

Note: confidenceScore indicates AI certainty of prediction.

AlertHistory Table: Used for storing emergency alerts made and sent by the system.

Table 3.5 Alert History Entity Schema

Field Name	**Data Type**	**Key**
alertID	INTEGER(10)	Primary Key
eventID	INTEGER(10)	Foreign Key
Recipient	TEXT(50)	
alertTime	DATETIME(10)	
deliveryStatus	TEXT(10)	

Evidence Table: Used for storing metadata of emergency evidence.

Table 3.6 Evidence Entity Schema

Field Name	Data Type	Key
- evidenceID	INTEGER(10)	Primary Key
- eventID	INTEGER(10)	Foreign Key
- evidenceType	TEXT(50)	
- filePath	TEXT(100)	
- captureTime	DATETIME(10)	

The actual evidence is stored in Firebase Storage, but only paths to them are stored in the DB.

3.5.3.4 Firebase Collection Schema

SQLite works with relational tables; Firestore of Firebase works with documents in collections. The suggested Firebase schema will have the following collections:

Users Collection: stores the synchronized user information

Users:

1. UserID
2. FullName
3. Email
4. PhoneNumber
5. Registration Date
6. Account Status

Emergency Contacts Collection: stores the cloud-based emergency contacts

EmergencyContacts:

1. ContactID
2. UserID
3. Contact Name
4. Relationship
5. PhoneNumber

DangerEvents Collection: stores AI-generated emergency events

DangerEvent:

1. EventID
2. UserID
3. Danger Type
4. Confidence Score

5. Timestamp
6. Status

AlertHistory Collection: stores cloud-synchronized alert history
AlertHistory:
1. AlertID
2. EventID
3. Recipient
4. Alert Time
5. Delivery Status

Evidence Collection: Stores metadata for multimedia evidence in Firebase Storage
Evidence:
1. EvidenceID
2. EventID
3. Evidence Type
4. File URL
5. Capture Time

3.5.3.5 Relationships Between Entities

The proposed database uses several one-to-many relationships as follows:

i. **User → EmergencyContact**
A single user can have many emergency contacts.
Relationship: 1: M

ii. **User → SensorData**
A single user can make many sensor data entries during system operations.
Relationship: 1: M

iii. **User → DangerEvent**
A single user may go through many detected danger events.
Relationship: 1: M

iv. **DangerEvent → AlertHistory**
A detected danger event may lead to multiple alert records.
Relationship: 1: M

v. **DangerEvent → Evidence**

A danger event may hold multiple evidences.
Relationship: 1: M

3.5.3.6 Data Storage Structure

The proposed system uses a multi-layered data storage mechanism to ensure effective management of data. There are roughly three data storage layers in the proposed system, namely:

A. Local Storage Layer: This data storage layer is developed by using SQLite that ensures offline capability and fast data access. It is used for storing:

i. User profiles
ii. Emergency contacts
iii. Sensor readings
iv. Stored copies of emergency events
v. Temporary evidence references

B. Cloud Storage Layer: This data storage layer is provided via the use of Firebase Firestore. It is used for storing:

i. Synchronized user records
ii. Alert history records
iii. Danger event records
iv. Metadata of emergency evidence

C. Multimedia Storage Layer (Firebase Storage): This helps prevent database size overload. It is used for storing:

i. Audio evidence
ii. Taken pictures
iii. Future video evidence

3.5.3.7 Data Access Mechanism

The system adopts a layer data access model in which requests first access SQLite data immediately but get synchronized later via Firebase Firestore whenever there is an internet connection.

3.5.3.9 Security Measures of the Database

The system processes sensitive data including user identity, GPS, emergency alerts, evidence files. Therefore, there are security measures implemented in the database:

i. User Authentication and Authorization
ii. Password Hashing and Encryption
iii. Secure Firebase Rules
iv. Controlled access to emergency data
v. Encryption during synchronization
vi. Restricted access to evidence files

3.5.4 System Architecture Design

The system architecture describes the parts that make up the system, how they are organized, and how they interact. Good architecture will allow for scaling, maintenance, reliability, and efficiency because of the clear definition of each part's role (Pressman, R. S., & Maxim, B. R. (2020).). In this case, an architecture consisting of five layers on the mobile phone was selected to accommodate usage on mobile devices and incorporate mobile sensors and AI. The five tiers in this architecture are:

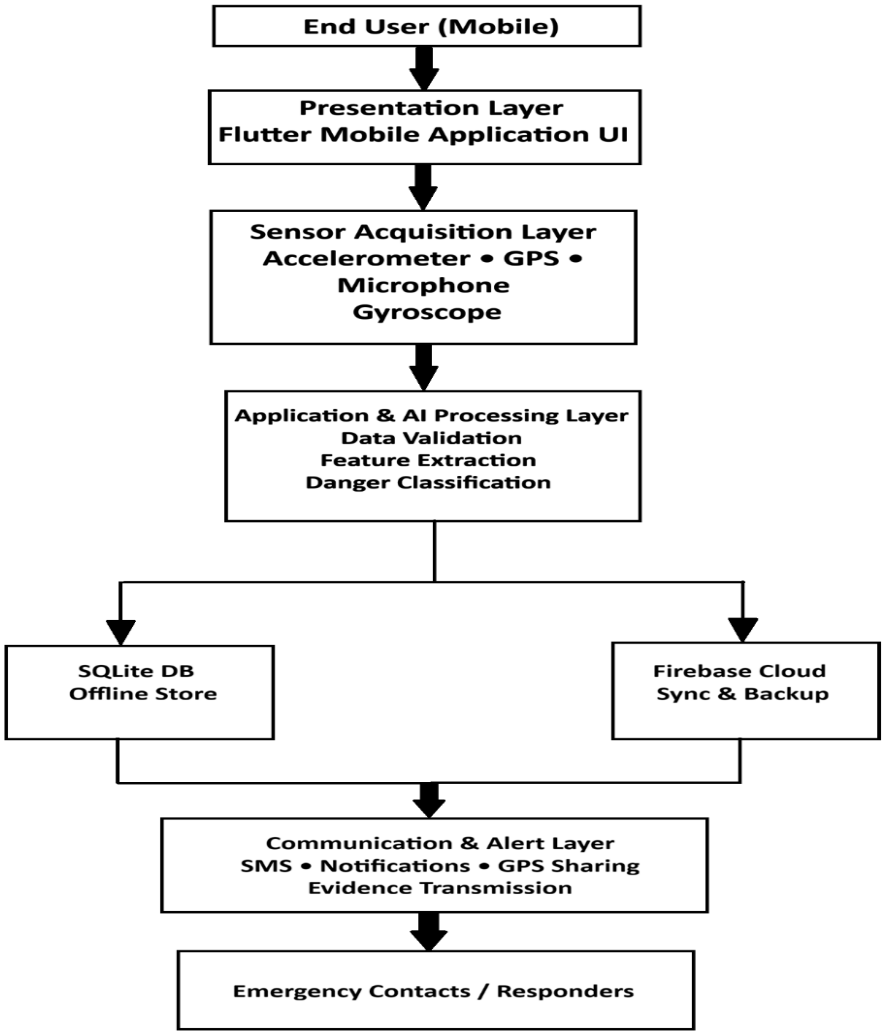
i. Presentation (Frontend)
ii. Sensor Acquisition
iii. Application and AI Processing
iv. Database
v. Communication and External Services

3.5.4.1 High-Level System Architecture

The architectural design explains how the data from sensors is collected, processed through AI module, stored into the database, and used to raise alarm messages whenever there is any threat. The mobile multi-tier architecture allows for:

i. Real-time danger monitoring
ii. Artificial Intelligence based detection
iii. Offline and Online operation
iv. Secure storage
v. Automated communication

Fig 3.8: System Architecture design



A. Presentation (Frontend) Layer

The presentation layer is the user interface portion of the application and is designed as a mobile application using the Flutter framework. This is where the graphical user interfaces for configuring the app, monitoring actions, and feedback are made. Simple to use and easy to navigate, it facilitates fast decision-making during emergency situations. The frontend manages:

1. User sign-up and login
2. Managing emergency contacts
3. Configuration of the system
4. Manually triggering SOS
5. Monitoring interface
6. Notifications and alerts

B. Sensor Acquisition Layer

This layer gathers real-time data from phone sensors which forms the basis for intelligent detection. This layer consists of:

1. **Accelerometer:** Measures movement, falls, and unusual activities.
2. **Gyroscope:** Monitors orientation and rotation of the phone.
3. **GPS:** Supplies coordinates for real-time location and emergency sharing.
4. **Microphone:** Picks up ambient noises such as screams and cries for help to detect distress. Real-time data collection is crucial in monitoring and intelligent detection of emergencies .

C. Application and AI Processing Layer

This layer is responsible for providing intelligence: processing, analyzing, decision-making, and emergencies triggering. This layer consists of:

1. Data validation: Purges invalid/erroneous/irrelevant data to improve model input.
2. Data preprocessing: Prepares sensor data through normalization and noise reduction.
3. Feature extraction: Extracts meaningful patterns in sensor data representing normal/abnormal activities.
4. AI-based danger detection: Classifies events into categories of Safe, Suspicious, and Dangerous, and makes decision.
5. Decision-making: Decisions based on AI output to continue monitoring, ask for confirmation, or trigger an emergency – more advanced than manual-only emergency apps.

D. Database Layer

The database layer holds, retrieves, manages, and synchronizes all system information. The system uses a combination of databases architecture through SQLite and Firebase in order to increase reliability and availability.

1. **SQLite Database:** Local only, which allows the system to function offline. It serves as a local database for:

   i. User profile information

ii. Emergency contacts storage

iii. Sensor logs storage

iv. Offline events logging

v. Evidence references temporarily

2. **Firebase Firestore:** Cloud database for:

   i. Synchronization in real time

   ii. Backup in the cloud

   iii. Access remotely to the data

   iv. Alert history

   v. Emergency records management long-term

3. **Firebase Storage:** Multimedia evidence from emergencies storage (audio, images, media in the future). The layers above provide improved reliability, accessibility, and disaster recovery.

E. Communication and External Service Layer

This layer allows interacting with external technology for emergency response purposes, such as:

1. **GPS Service API:** For providing real-time location:

   i. User location tracking
   ii. Sharing the location
   iii. Localization of the emergency

2. **SMS Service API:** For maintaining alerts under limited connectivity by sending:

   i. SOS messages
   ii. Location information
   iii. Alerts about emergency

3. **Firebase Services API:** Secure data management and synchronization through:

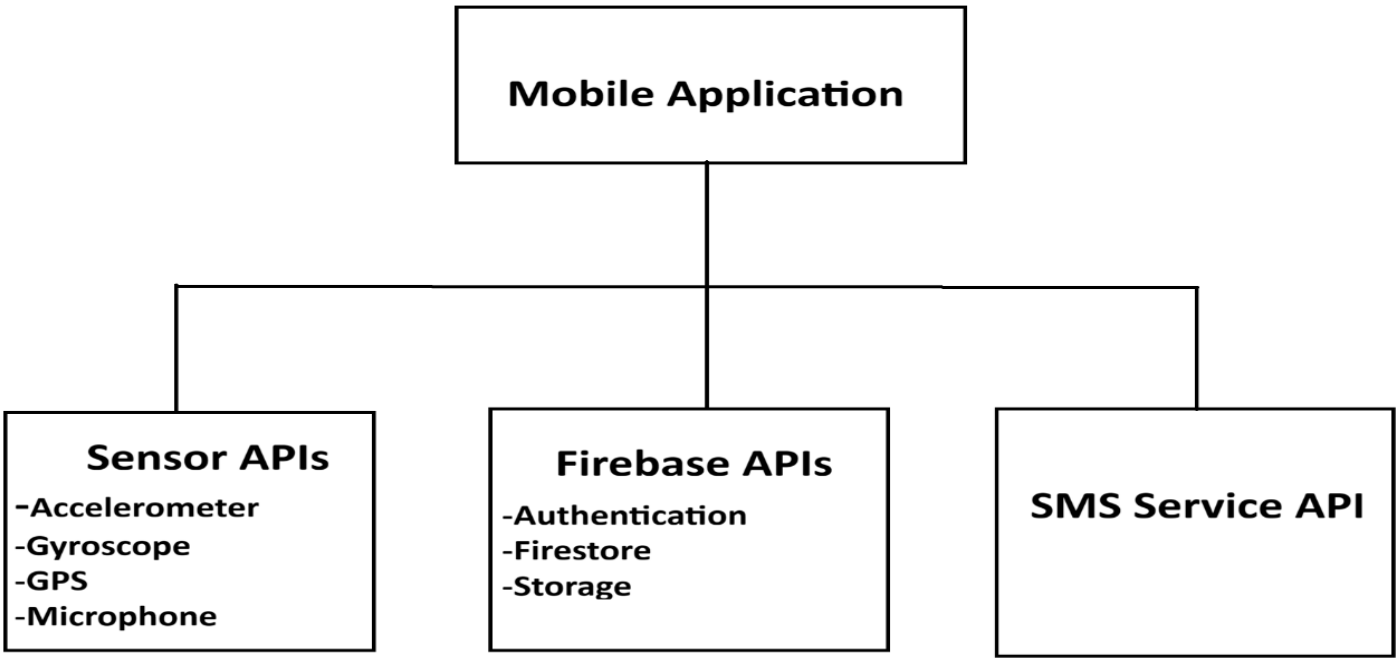
   i. Firebase Authentication
   ii. Firebase Firestore

iii. Firebase Storage

4. **Device Sensor APIs**: Interface for accessing the hardware in the device:

   i. Accelerometer
   ii. Gyroscope
   iii. GPS coordinates
   iv. Microphone

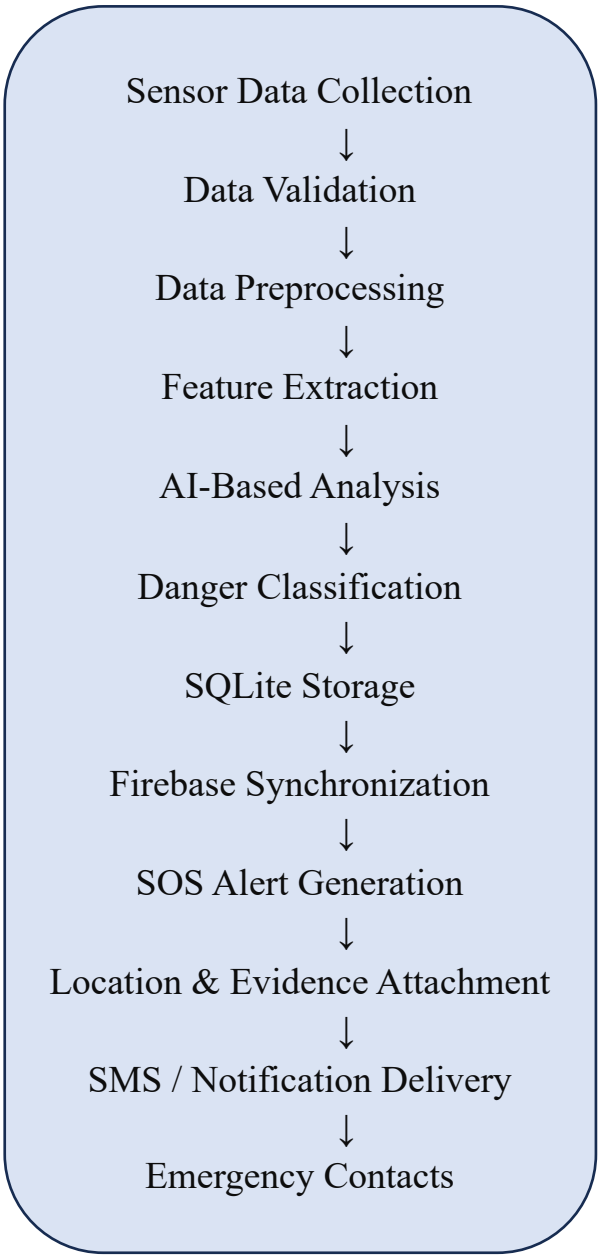
Fig 3.9: The Relationship Between the Application and External Services



3.5.4.2 System Data Flow Architecture

The flow of information within the proposed architecture follows a sequential process beginning with sensor acquisition and ending with emergency notification delivery. The data flow process is illustrated below.

Fig 3.10: Data Flow Process Diagram



CHAPTER FOUR
SYSTEM IMPLEMENTATION

4.0 Introduction

This chapter details the implementation of Intelligent Mobile-Based Emergency Detection and Automated SOS System. It includes environment, tools, languages, frameworks, dataset, ML algorithm, and procedures needed for implementing the Intelligent Mobile Based Emergency Detection and Automated SOS system.

Mobile-based system identifies user emergencies through sensors and generates alerts automatically. Multimodal identification of events such as distress voice, abnormal motion, fall, vehicle crash, and sensor fusion is done to identify user emergencies. Random Forest Classifier is chosen for event classification due to its effectiveness and ability to avoid over-fitting which is good for sensor-based data.

The Random Forest Classifier is trained using a combination of public data set and data collected from smartphones' sensors. Tools used for developing the application include Flutter, Python, SQLite, and Firebase and Smartphone Service for delivering emergency messages. Random Forest Classifier is popular in activity recognition in handling multidimensional sensor data.

4.1 Development Environment and Tools

The implementation of the system required an integration of software tools, ML framework, mobile application technologies, database and cloud services that are compatible, scalable, efficient and suitable for mobile AI.

4.1.1 Programming Languages and Libraries

In this system design, several programming languages and auxiliary libraries are utilized in the development of various components that make up the architectural framework.

A. Flutter (Dart Programming Language)

Mobile application is implemented through the use of Flutter 3.41.9 cross-platform application development framework provided by Google. Dart is the primary programming language used in the development of mobile application interfaces, monitoring of sensors, emergency contacts and communications. According to Google, Flutter provides developers with an easy way to create highly-performing applications from one code base. It was chosen because it provides:

i. Compatibility across platforms
ii. Faster development cycles
iii. User interface components
iv. Performance
v. Firebase services integration

Flutter Packages Used

The following are the Flutter plugins used in the app:

1. **Sensors Plus:** Accelerometer/Gyroscope sensors
2. **Geolocator:** GPS functionality
3. **Firebase Authentication:** for user authentication
4. **Cloud Fire Store:** Cloud Database
5. **Firebase Storage:** Media storage
6. **Flutter Local Notifications:** Notification functionality
7. **Permission Handler:** Runtime Permissions
8. **Telephony/sms:** SOS messages

B. Python Programming

Python programming language is often used in artificial intelligence/machine learning due to simplicity and available libraries (Russell, S. J, & Norvig, P. 2021). The development of the machine learning component was done using Python 3.12.8 because of:

i. powerful support for machine learning
ii. rich scientific library
iii. preprocessing tools
iv. big community
v. simple integration with ML frameworks

Python Libraries

The following Python libraries have been employed in developing the proposed system:

i. Scikit-Learn: Random Forest classifier for activity recognition and emergency detection (Pedregosa, F., Varoquaux, G., Gramfort, A., et al. (2011)). because of its fast and accurate algorithms, efficient data processing, and evaluation and classification modules.

ii. NumPy: Library used for performing numerical computation, mathematics, and feature extraction.
iii. Pandas: Importing and managing datasets.
iv. Matplotlib: Plotting of graphs and creation of matrices

C. Machine Learning Framework

Random Forest Classifier

In the developed system, a Random Forest Classifier is used. It's an ensemble of multiple decision trees that increases the accuracy and reduces overfitting while providing fast prediction on moderate datasets (Breiman, L. (2001).).

4.1.2 Development Platform and Hardware Needs

This project needed different platforms and hardware for the development, testing, and deployment of applications and ML experimentation.

Software Development Platforms

1. **Android Studio:** IDE that can be used for developing, testing and debugging applications on the Android platform.
   i. App development using flutter
   ii. Android SDKs management
   iii. Testing via emulator
   iv. Debugging
   v. Builds generation

2. Visual Studio Code: an easy-to-use lightweight editor that will help increase efficiency; secondary IDE for:

   i. Coding
   ii. Flutter development
   iii. Python scripting
   iv. Project management
   v. Git integration

3. Firebase Console: allows scalability and synchronization in the cloud; useful for:

   i. Cloud Firestore database
   ii. Authentication
   iii. Cloud storage

iv. Monitoring application

4. Hardware Requirements: Configuration of hardware needed for development:

1. **Processor:** Intel Core i5 or better
2. **RAM:** 8 GB minimum
3. **Storage:** 256 GB minimum (SSD)
4. **Operating System:** Windows 10/11
5. **Mobile Phone:** Android Smartphone
6. **Internet Connection:** Needed for Firebase Syncing

A smartphone is used as the platform for validation of data collection, machine learning prediction, GPS localization, and SOS functionality.

Environment justification

The described combination provides an intelligent mobile application framework. Flutter allows quick application development, Python enables experimentation in machine learning area, Scikit-Learn provides the efficient implementation of the Random Forest classifier, SQLite guarantees offline access, and Firebase provides the scalable cloud services. Together, all of these components enable real-time emergency detection, efficient data storage, reliable SOS feature, and future extension of the system with additional emergency modules.

4.2 Dataset description and preprocessing

4.2.1 SisFall dataset

SisFall is a publicly available fall-detection dataset for performance evaluation of the system. The dataset contains sensor readings for young and elderly people performing their daily activities and falling:

a. Sensors: accelerometer, gyroscope
b. Falls: multiple falls scenarios
c. Activities: multiple Activities of Daily Living (ADL)

Activity examples within the dataset:

a. **Normal:** walking, sitting, standing, lying down, stairs up/down
b. **Falls:** forward, backward, side, syncope, walk and fall

SisFall provides numerous motion examples in controlled conditions, which makes it suitable for training of the machine learning models to detect normal behavior from falling-related movements. Participants with elderly age increase the practical value of the system for emergency monitoring. The following tasks can be solved using SisFall dataset:

i. Training the fall-detection model
ii. Motion anomaly analysis
iii. Pattern of impact recognition
iv. Extraction of sensor features

4.2.2 MobiAct Dataset

MobiAct is one more publicly available dataset for HAR and fall detection problem with focus on smartphones and, thus, is highly relevant to a mobile application. It contains:

a. Sensor readings from: accelerometer, gyroscope, and orientation
b. Users' activity labels

The two types of activities presented in MobiAct:

a. **ADL (Activities of Daily Living):** These activities involve normal user behavior.
   Examples of ADL: walking, jogging, standing, sitting, stairs up/down, jumping.

b. **Fall Activities:** These represent emergency situations.
   Examples include: Front falls, Back falls, Side falls, Obstacle-induced falls

The MobiAct dataset is similar to the contemporary smartphones' sensor readings, and this will improve the model's performance on mobile devices. It highly includes:

i. Detection of motion anomalies.
ii. Fall identification.
iii. Study of smartphone sensors' behavior.
iv. Activity recognition learning.

4.2.3 Custom Smartphone Validation Dataset

Although public datasets are useful, they do not entirely mimic real-life situations. A custom smartphone validation dataset was created in order to complement the public datasets and improve flexibility, using the sensors from the built application.

Data collected from the sensors:

a. Accelerometer: movement intensity, abrupt shocks, motion transitions.
b. Gyroscope: device orientation, rotations, angular velocities.
c. GPS: user location, speeds, movements.

Activities conducted in order to collect the dataset included:

Table 4.1: Custom Dataset

Activity Type	**Activity Functionality**
Walking	Normal activity
Sitting	Normal activity
Standing	Normal activity
Stair climbing	Normal activity
Running	Normal activity
Falls	Emergency activity
Impacts	Emergency activity
Motion change detection	Motion anomaly detection
Vehicle motion scenarios	Crash detection assistance

Uses of the data included:

i. Model verification;
ii. System testing; and
iii. Performance evaluation in real-life situations.

4.2.4 Distress Voice Dataset

As detecting distress voice is the primary objective of the proposed system, an exclusive audio dataset was developed to train and verify the emergency voice identification system in a controlled environment. It includes:

A. Distress voice signals: "Help me!"**,** "Please help!"**,** Distress sounds, etc.

B. non-distress voice signals: Normal conversation**,** Ambient noise**,** Environmental sound, House noises and Outdoor noises

Inclusion of non-distressed samples aims at making the ML classifier able to differentiate between an emergency situation and other environmental noises. Audio was recorded under various conditions—silent environment, indoor environment, outdoor environment, and environment with some level of background noise—so as to enhance model robustness. Data used includes the detection of distress-voice detection, emergency classification via audio, and multi-modal sensor fusion. Incorporating audio adds intelligence to the system by facilitating the emergency detection even without motion information.

4.2.5 Data Integration and Multi-Modal Fusion

For the first time, the system makes use of more than one source of data rather than utilizing just one sensor. Various types of data are integrated to enhance the accuracy of emergency detection. Sources that have been integrated for the purpose include Microphone Audio, Accelerometer, Gyroscope, Contextual Activity Features, and Multi-modal Emergency Dataset. The integration of data will enable the model to examine emergencies based on various indicators. Consider the examples in Table 4.4.

Table 4.2: Scenario and Sensor Indicator Table

Situation	Sensor Alerts
Fall Incident	Impact & Orientation Shift
Distant Distress Signal	Special Audio Alert
Car Accident	Deceleration & Impact
Movement Disruption	Unusual Sensor Movement
Critical Emergency	Eventual Combination of Alerts

4.2.6 Data Preprocessing

The raw sensor data contain noise, incompleteness, redundancies, and inconsistencies in scale. Thus, the preprocessing step is used to improve the data and the model itself. Preprocessing includes the following components:

A. Data Cleaning

Removal of invalid instances (records without sensor data, corrupted instances, duplications, incomplete samples of activities). The goal is to increase training stability and minimize misclassifications.

B. Noise Filtering

Use of filtering algorithms for the sensor data and feature consistency improvement to avoid false emergencies due to noise.

C. Data Normalization

The values are normalized due to different value scales (e.g., accelerometer values of -20 to 20, gyroscope values of -500 to 500). The normalization helps to increase the classification stability and accuracy.

D. Data Augmentation

- Data augmentation algorithms are used to increase data variability and achieve better generalization in the case of the use of time-series and audio data.

Augmentation of Motion Sensors (Accelerometers/Gyroscopes)

i. Adding low-level noise
ii. Scaling the signal
iii. Time shift
iv. Window slicing with overlap

Augmentation of Distress Voice

i. Background noise addition
ii. Volume change
iii. Time stretching
iv. Pitch change

Augmentation approaches adopted for sensor data include:

i. Introduction of random noise
ii. Time shifting
iii. Scaling of the signal
iv. Window slicing

For the distressed voice audio samples, the following augmentation approaches have been adopted:

i. Adding background noise
ii. Volume change
iii. Time stretching
iv. Pitch alteration

4.2.7 Feature Extraction

Feature extraction transforms the raw sensory data into input variables that can be used in the machine learning algorithm. These features represent the activities and environment of interest for the model. For user movements, the following features have been extracted:

1) Accelerometer:

i. Mean, maximum and minimum acceleration
ii. Standard deviation
iii. Signal magnitude area

2**) Gyroscope:**

i. Angular velocity
ii. Rotation variance
iii. Orientation change

3) **GPS:**

i. Speed of the user
ii. Distance covered by the user
iii. Movement consistency

4) **Audio:**

i. Sound intensity
ii. Sound energy
iii. Sound frequency
iv. Audio time series

4.2.8 Dataset Partitioning

Once data was preprocessed, feature-extracted, and augmented, the concatenated dataset was partitioned into training, validation, and testing sets. Below is the dataset partitioning described in Table 4.1.

Table 4.3: Training, Validation, and Testing Split Ratio

Data Subset	Proportion	Usage
Training Dataset	70%	Model learning and pattern recognition
Validation Dataset	15%	Hyperparameter tuning and model optimization
Testing Dataset	15%	Model performance evaluation

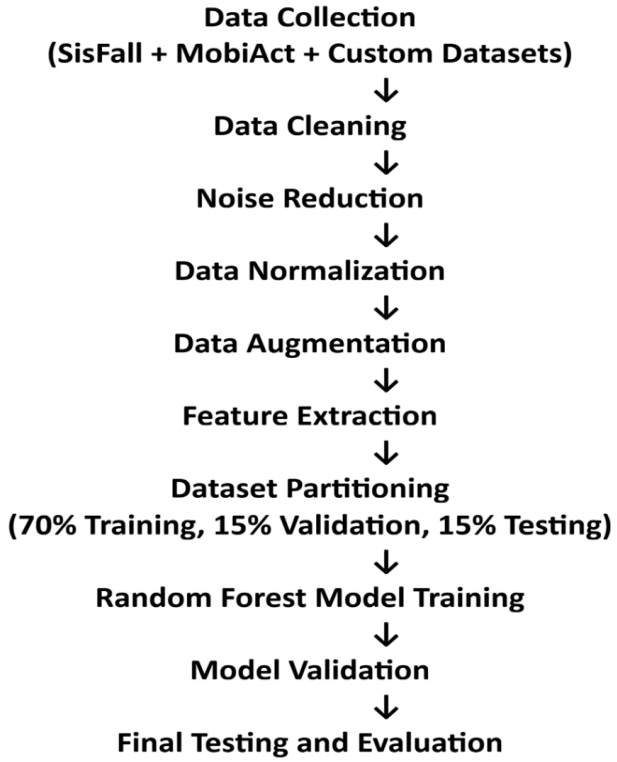
The training dataset (70%) was applied to train the Random Forest classifier and let the model learn connections between sensor features and emergencies.

The validation dataset (15%) was applied while building a machine learning model to assess intermediate model performance, optimize parameters, and avoid overfitting. The usage of the validation dataset allows evaluating the machine learning model using data that were not seen by the model at the training stage.

The testing dataset (15%) was intended solely for evaluating model performance at the final stage. This dataset was not used for the training and validation phases.

The use of training-validation-testing technique increases the accuracy of performance estimation and is widely accepted approach in machine learning systems development (Géron, 2022). It is possible to evaluate the performance of the proposed system more accurately using this approach.

Fig 4.1: Data Preparation Workflow



4.3 Model Architecture and Training

The success of an emergency detection algorithm depends on the ability of the model to differentiate between normal and emergency situations. Following data preprocessing and feature extraction (Chapter 4.2), the machine learning classifier recognizes the patterns of emergencies from the merged multi-modal features. The application is capable of recognizing whether the input is an emergency or not using the Random Forest Classifier trained on sensor and audio features from the SisFall, MobiAct, Custom Smartphone Validation Data Set, and Custom Distress Voice Data Set.

In comparison with deep learning techniques, Random Forest ensures high accuracy, immunity to overfitting, and efficient processing in constrained settings, thus being the suitable technique for smartphone emergency recognition (Breiman, L. (2001).).

4.3.1 Architecture of Random Forest Classifier

The proposed emergency detector uses the technique of building multiple decision trees as an ensemble to improve the prediction accuracy. During training, several decision trees are built and their results are combined using the majority voting during classification. The classifier architecture consists of three key components:

Step 1: Features Input Layer

Input features from acceleration, gyroscopic sensors, GPS motion data, distress voices, and other activities to know the user's physiology and situation.

Step 2: Random Forest Decision Layer

Use all features in a Random Forest classification algorithm. The multiple decision tree ensemble constructed randomly will give an output of one of the following:

i. Regular activity
ii. Distress voice found
iii. Anomalous motion
iv. Fall detected
v. Collision detected
vi. Emergency situation

Step 3: Classification Output Layer

By majority voting across the tree ensemble, Emergency Detected or No Emergency Detected can be obtained and sent to the emergency response layer for initiating SOS operations.

Fig 4.2: Conceptual Random Forest Architecture

4.3.2 The Rationale Behind the Selection of Random Forest

Technical and practical factors are combined here. The emergency detection algorithm requires input from heterogeneous data collected from various sensors. Random Forest deals well with mixed data type features without requiring much transformation. Its ensemble-based architecture makes overfitting less likely due to training on different data samples. Additionally, it is suitable for medium-sized datasets.

The main advantages of Random Forest are:

i. Good classification performance
ii. Efficient predictions
iii. Limited overfitting
iv. Tolerance to noise
v. Simplicity
vi. Transparency for scholarly assessment

4.3.3 Multi-Modal Feature Representation

In order to increase the robustness of the emergency detection process, the developed model relies on multi-modal feature representation. In contrast to other methods which use a single sensor for data collection, our model uses multiple information sources such as Motion Features, Audio Features, GPS Features and Sensor Fusion Features.

4.3.4 Model Training Process

Our model training process follows the classic machine learning procedure.

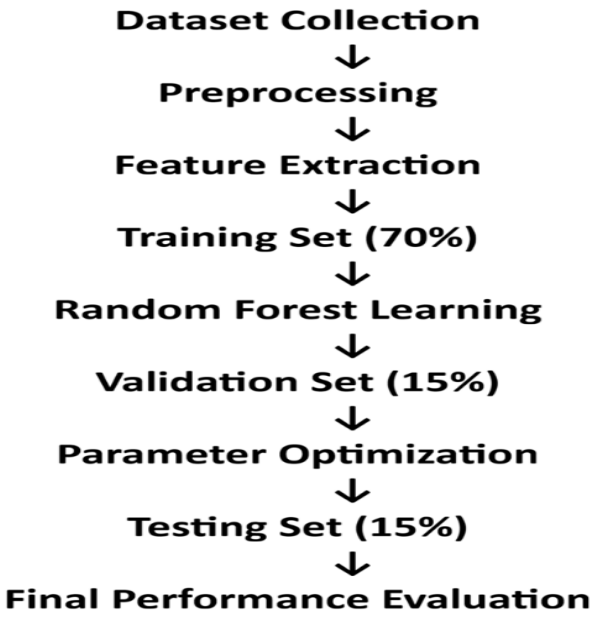
A. Dataset preparation: Before starting the training, the prepared datasets were subjected to: Data cleaning, Normalization, Augmentation and Feature extraction

B. Training: While training, the Random Forest algorithm learns how different features are related to activity labels. Training dataset contains 70% of all data and includes both emergency and normal events. Some examples of training labels are: Normal activity, Distress voice event, Motion anomaly event, Fall event and Vehicle crash event.

C. Validation: Validation is performed by using 15% of all data for parameter tuning, monitoring of the algorithm performance and generalization capability. Validation helps to prevent overfitting.

D. Testing: 15% of the data left for testing phase which allows us to evaluate how the model will perform when applied to new unseen data in the real world.

Fig 4.3: Training Workflow



Model Training Parameters

While deep learning models require parameters such as epochs, batches, optimization and loss functions to control the process of learning, Random Forest classifiers make use of tree-based parameters for the same. The model parameters used in this study are listed in Table 4.2 below.

Table 4.4: Random Forest Training Parameters

Hyperparameter	Value	Functionality
Number of Trees (n_estimators)	100	The more trees we have, the better will be predictions and less variation will occur.
Maximum Depth (max_depth)	15	Controls depth to avoid overfitting but still allow learning.
Minimum Samples to Split (depth))	5	Number of minimum samples required for splitting a node.
Minimum Samples per Leaf (min_samples_leaf)	2	At least two samples should exist at the end node.
Maximum Features (max_features)	sqrt	Square root of features should be used at each split to increase diversity.
Bootstrap	True	Each tree is trained on different subset of data using bootstrapping.
Random State	42	Allows reproducibility between runs.

4.3.5 Overview of Model Architecture

This model allows intelligent recognition of distressing voices, abnormal motions, falls, and car accidents, with alerts issued automatically. The machine learning model architecture built for the proposed system is outlined in the diagram below:

Fig 4.4: Machine Learning Architecture

4.4 System Implementation and Modules

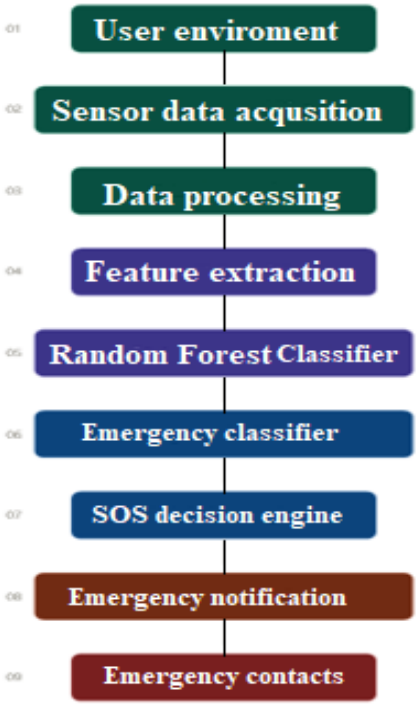
System implementation transforms the design of Chapter 3 into a working app of intelligent emergency detection and SOS alert generation. The Intelligent Mobile-Based Emergency Detection and Automated SOS System is developed using Flutter for UI development, Python with Scikit-Learn for ML, Firebase and SQLite for data management, and sensors of smartphones for monitoring. A module-based structure assigns unique functions to individual modules for achieving better performance of system maintenance, scalability, fault isolation, and further development. There are several methods of detecting emergencies included: distress voice detection, motion

abnormalities detection, fall detection, vehicular accident detection, GPS tracking, and automated SOS communication.

4.4.1 Overall System Implementation

This system is mobile-oriented. The smartphone serves as the source of data and the responder in case of any emergencies. It constantly monitors its sensors and context, which include: Microphone, Accelerometer, Gyroscope, GPS and Predefined settings for an emergency. Emergency detection is performed using the ML engine. After detecting the emergency, predefined response protocol is executed. The process of implementation is described below.

Fig 4.5: Overall Implementation Workflow Diagram



4.4.2 Module of User Registration and Authentication

This module deals with user identity, access control, and user authentication. It employs Firebase Authentication, Flutter Forms, and validation methods in order to make sure that the application is used only by the authorized individuals.

Functions:

1. Account registration
2. Login into account
3. Forgotten password

4. Credential verification
5. User profile management

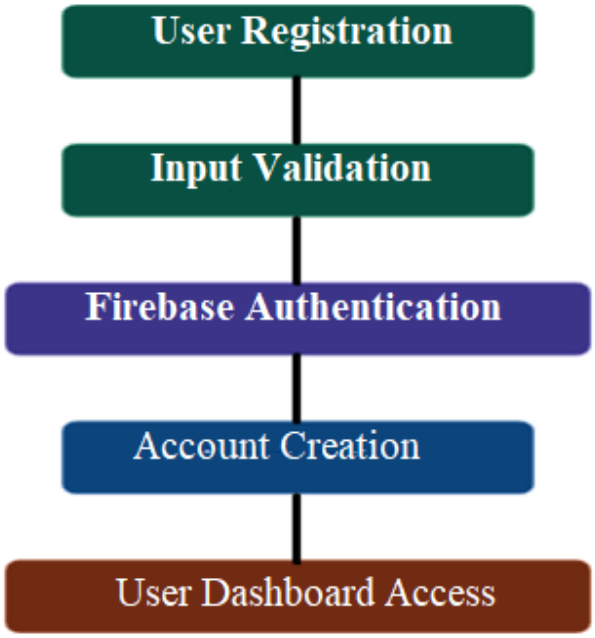
User Registration: When registering users, input validation occurs prior to saving data in Firebase authentication, based on information from table 4.5.

Table 4.5: User Registration Table

Full Name	Identifier for the user
Email Address	Login identifier
Phone Number	For emergencies
Password	For securing the account

Authentication Process: This step enhances security by validating user identity.

Fig 4.6: Authentication Workflow Diagram



Emergency Contact Management Module

The Emergency Contact Management Module allows users to register their trusted friends whose duty is receiving an emergency notification after detecting a hazard. The module holds contact data locally on SQLite and keeps the synchronization of the contact list with Firebase Firestore.

Module Functions

Users can:

1. Create emergency contacts.
2. Change emergency contacts.
3. Delete emergency contacts.
4. Prioritize contact lists.
5. Synchronize contacts in cloud storage.

Data Storage
The module stores the following data:

Table 4.6: Data Storage Field

Field	Purpose
Contact Name	Identification
Phone Number	Notification channel
Relationship	Category of contacts
Priority Level	Priority level of notification

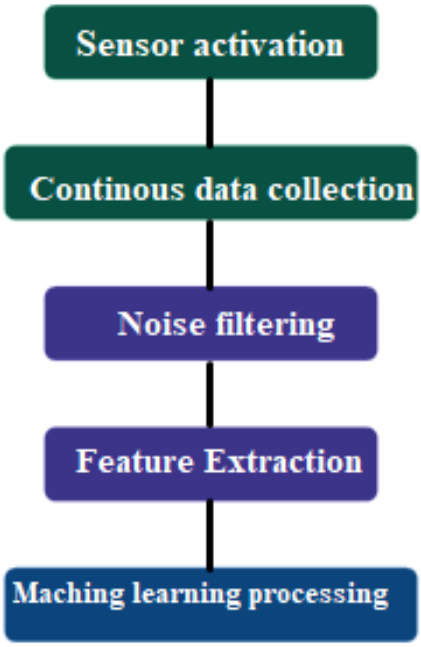
Significance of the Module
Emergency contacts become the main recipients of:
1. SOS messages.
2. GPS location.
3. Emergency evidence.
4. Distress alerts.

4.4.3 Sensor Monitoring and Data Acquisition Module

This component obtains live smartphone sensor data for emergency detection by means of the ML engine. Being implemented with Flutter background services, it monitors smartphone sensors even when the application is not running. Situation awareness includes combining data from accelerometers and gyroscopes (motion), GPS (location), and microphone (distress voice samples). Fall detection has been included; other modalities like voice can be easily added. Data preprocessing involves timestamp adjustment, noise reduction, verification of sensor availability, elimination of incomplete records, normalization, and generation of robust feature vectors for the Random Forest classifier. To improve energy efficiency, there are provisions for

adaptive monitoring, efficient polling, optimal background use, and elimination of redundant data to enable real-time detection. Post-processing includes transferring data to the Intelligent Emergency Detection Module. The timestamps and location data are saved locally in SQLite and synchronized with Firebase, if possible, to make decisions as quickly as possible. In other words, this module acquires and supplies the system with live data (Al Mudawi, N., Azmat, U., Alazeb, A., Alhasson, H. F., Alabdullah, B., Rahman, H., Liu, H., & Jalal, A. (2025).

Fig 4.7: Sensor Monitoring Workflow Diagram

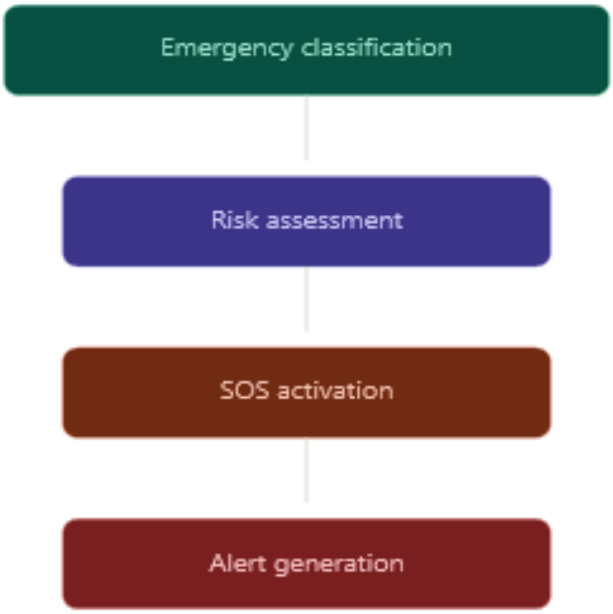


4.4.4 Intelligent Emergency Detection Module

The Intelligent Emergency Detection Module acts as the intelligence center of the Intelligent Mobile-Based Emergency Detection and Automatic SOS System. It processes information from the sensors to detect signs of emergencies, currently focused on fall detection using a Random Forest classifier model trained on a hybrid data set. This prototype provides a base for further development of additional features such as detection of distress voice and vehicle crash. Using feature vectors based on data obtained from several sensors, the module capitalizes on feature dependencies and relationships to enable fast and accurate detection. A verification process is used to prevent false alarms before taking any actions to generate an SOS signal. Its modular

structure enables additional features integration with minor changes in the code. The system stands out because of its real-time processing capabilities and machine learning, among others. Using real-time sensor analytics, machine learning for activity recognition, intelligent decision-making, and a scalable modular design, the module allows for the detection of emergencies while ensuring that it can be expanded into a full-fledged multimodal emergency detection system in the future (John, A., Cardiff, B., & John, D. (2024)).

Fig 4.8: Emergency Decision Workflow



4.4.5 Emergency Response and Notification Module

This module comes into action once the Intelligent Emergency Detection Module detects the occurrence of an emergency. It transforms ML predictions into actions by sending alerts to pre-defined emergency contacts immediately to provide faster help in case of any life-threatening situation.

Process:

i. Validate the event to avoid the generation of false alarms due to sensor noise. In case confidence of prediction exceeds the defined threshold, generate an alert including emergency type, timestamp, user GPS coordinates, and additional monitoring context.
ii. Fetch the registered emergency contacts from local SQLite database and build the SOS message including user identification information, emergency category, location

information, and a link to Google Maps. In case the Internet connection is available, synchronize the alert information with Firebase Cloud Firestore. It will be easier to integrate with emergency services, healthcare, and security systems later through API connections.

Evidence collection and situational awareness:

Collect and save evidence of the incident that includes sensor readings, time stamps, location data, diagnostic information, etc. to help responders. Optionally, collect audio and multimedia evidence of distress detection and advanced recognition.

Reliability and offline mode support:

In case the Internet connection is not available, the emergency data is saved locally and synchronized with Firebase when Internet connection will become available. Hybrid storage solution provides robustness against data loss and synchronizes local and cloud databases. The system can also keep track of delivery and alerting history.

Implementation and data flow:

i. Receives classification results from the Intelligent Emergency Detection Module.
ii. Retrieves user profile and contacts from the User Management Module.
iii. Passes events and evidence to the Data Management and Synchronization Module for permanent storage.

Overview:

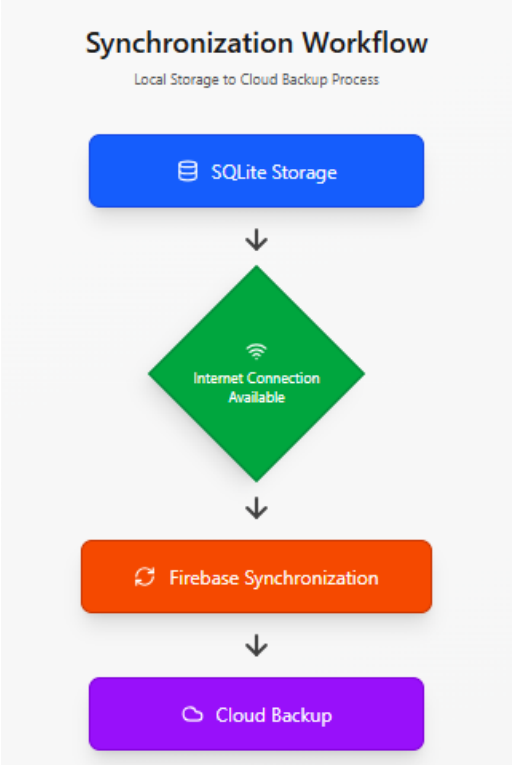
The module implements the automated functionality of SOS service which includes event validation, automatic alerts, location sharing, evidence management, offline and synchronization capabilities. The system is capable of further scaling with advanced communication technology integration and public safety services (Al Mudawi, N., Azmat, U., Alazeb, A., Alhasson, H. F., Alabdullah, B., Rahman, H., Liu, H., & Jalal, A. (2025).; John, A., Cardiff, B., & John, D. (2024)).

4.4.6 Data Management and Synchronization Module

The Data Management and Synchronization Module ensures secure storage, access, and synchronization of data for the Intelligent Mobile-Based Emergency Detection and Automated SOS System. It employs an SQLite local database for offline access and a Firebase Cloud Firestore for centralized storage. The local database holds user profiles, emergency contacts, and sensor data, allowing the application to function effectively

even without connectivity. Critical information is synchronized across authorized devices via Firestore, maintaining data integrity through asynchronous background services. Each incident is uniquely identified, connecting various records, while robust security measures protect data through input validation, authorization, and secure transmission protocols. The module integrates seamlessly with core components like User Management and Emergency Response.

Fig 4.9: Synchronization Workflow



4.5 Tests and Evaluations

Tests and evaluations review the system to determine whether the goals, requirements, and performance have been met. In regards to the Intelligent Mobile-Based Emergency Detection and Automated SOS System, tests were conducted for checking the validity of modules, integration, performance of ML models, and user interface operations. The key point was to check the Random Forest model's capability for identifying emergencies (distress calls, motion abnormalities, falls, and vehicle crashes) and to assure that the integration of all modules works fine.

The testing methods include:

i. Machine Learning Performance Evaluation
ii. Unit Testing

iii. Integration Testing
iv. User Interface Testing
v. Functional System Validation

4.5.1 Evaluation Metrics

Intelligent Mobile-based Emergency Detection and Automated SOS system was evaluated using traditional Machine Learning evaluation metrics. Accuracy, Precision, Recall, F-Score and Confusion matrix were used to measure the effectiveness, validity, and real-time nature using the Random Forest method to distinguish between normal and emergency events.

Table 4.7: Performance Metric Values

Metrics	Value
Accuracy	**95.1%**
Precision	**94.3%**
Recall	**96.0%**
F1-Score	**95.1%**

Accuracy: Accuracy is the percentage of total predictions made correctly in the test dataset for both emergency and normal classes. Although accuracy can imply the success of a machine learning model, one must consider other metrics too because of the possibility of class imbalance. The formula for accuracy is provided by Mohsin & Muyeed (2024).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Where:
TP = True positive
TN = True negative
FP = False positive
FN = False negative
Accuracy is high when the algorithm accurately classifies both emergencies and non-emergencies. In this case, accuracy reveals how well the system can classify the following:

i. Normal behavior
ii. Distress voice behavior
iii. Motion abnormality behavior

iv. Fall behavior
v. Vehicle crash behavior

Precision: The term precision refers to the percentage of the predicted emergencies that are actual. The precision of an algorithm indicates its reliability and reduces the number of false alerts. A high precision value indicates that there will be less false alarms, and this is important in order to prevent unnecessary panic.
Precision = TP / (TP + FP)

Example
The algorithm predicts 100 emergencies; 95 emergencies are actual while 5 of them are false alarms.
Precision = 95%

Recall: Recall, or sensitivity, refers to the proportion of actual emergencies detected by the system. It is very important in emergency detection due to possible delays in providing help. Recall is one of the most significant metrics because of the risks that come from missing emergencies.
Recall = TP / (TP + FN)

F1 Score: The F1 score is a combination of both precision and recall, as it calculates the harmonic mean of both metrics. This particular score is especially useful in evaluating classifiers that have been developed using imbalanced datasets, as it can give a misleading accuracy result. Formula:

$$F1 = \frac{2(Precision \times Recall)}{Precision + Recall}$$

F1-score
The F1 score provides a compromise between false positives and false negatives; the higher the F1 score, the more accurate the classification.

Confusion Matrix
This table gives a breakdown of how the Random Forest classifies activities, showing where the correct classifications are made and incorrect ones.
Table 4.8: Confusion Matrix

Actual / Predicted	Emergency	Non-Emergency
Emergency	TP	FN
Non-Emergency	FP	TN

A confusion matrix evaluates a classifier on the basis of the correct identification of true positives (emergency cases), true negatives (non-emergency cases), false positives (alarms with non-emergency cases), and false negatives (missing emergency cases). It reveals that the classifier is capable of detecting emergencies effectively with the right balance of detection and alarm generation. It supports the mobile surveillance technique of emergency detection (Al Mudawi, N., Azmat, U., Alazeb, A., Alhasson, H. F., Alabdullah, B., Rahman, H., Liu, H., & Jalal, A. (2025).; John, A., Cardiff, B., & John, D. (2024)).

4.5.2 System Testing

Testing was done on the system to ensure that the proposed application's modules were working properly either in isolation or when combined. The tests were conducted as follows:

1. Unit Testing
2. Integration Testing
3. Functional Testing

1. **Unit Testing:** Unit testing involves the process of evaluating each module of the software independently to check whether it is working correctly before integration.

Table 4.9: Test for User Registration Module

Test Scenario	**Data Used**	**Expected Output**
Valid Registration	Valid Details	Registered
Invalid Email	Wrong Format of Email	Error
Empty Field	Not Entering Required Field	Error Message

Result: Success

Table 4.10: Login Module Test

Test Case	**Input**	**Expected Result**
Correct Credentials	Email/Password	Successful Login
Incorrect Password	Incorrect Password	Login Failed

Empty Credentials	Missing Information	Validation Error

Result: Success

Table 4.11: Emergency Contact Module Test

Test Case	**Input**	**Expected Output**
Add Contact	Correct contact info	Contact added
Edit Contact	New contact info	Contact info updated
Delete Contact	Present contact	Contact deleted

Result: Success

Table 4.12: Test for GPS Module

Test Name	**Input**	**Expected Output**
Location Request	**GPS enabled**	**Retrieve coordinates**
GPS not Enabled	**GPS not available**	**Error message**

Result: Success

Table 4.13: Test for SOS Module

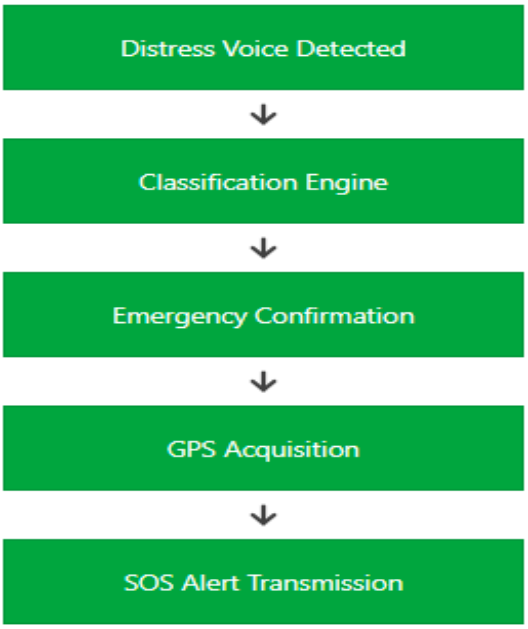
Test Name	**Input**	**Expected Output**
Emergency Alert	**Emergency**	**Generate SOS signal**
No Emergency Alert	**Not an emergency**	**Don’t send any alerts**

Result: Success

2. **Integration Testing**: Integration testing verified the interaction among modules and exchange of data after successful unit tests.

Test Scenario 1: Distress Voice Detection ➔ SOS Alert

Fig 4.10: Distress Voice Detection Workflow



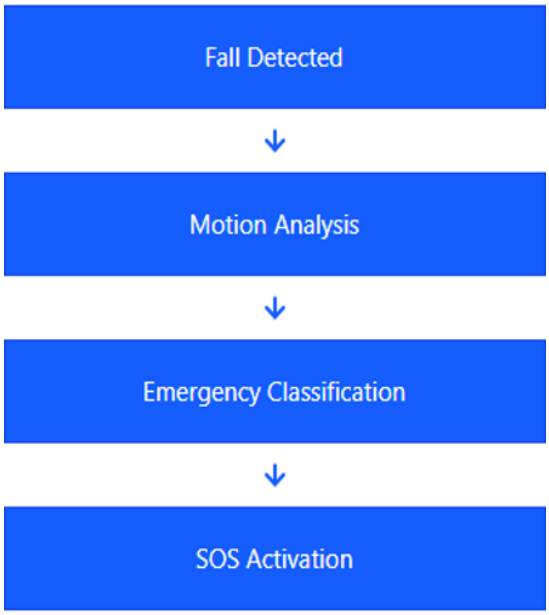
Expected Outcome:

i. Emergency identified
ii. Location determined
iii. Successfully sent alert

Outcome Status: Passed

Test Case 2: Fall Detection → Emergency Notification

Figure 4.11: Workflow for Fall Detection



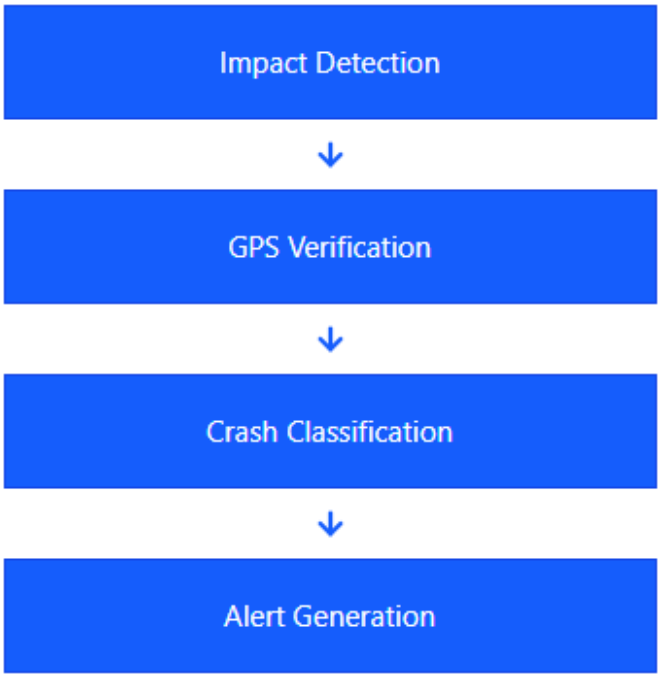
Expected Outcome:

i. Fall detected
ii. Emergency identified
iii. Generated notification

Outcome Status: Passed

Test Case 3: Vehicle Crash Detection → Contact Notification

Fig 4.12: Vehicle Crash Detection Workflow:



Expected Result:

i. Crash recognized
ii. Emergency contacts notified

Status: Passed

2. **Functional Testing:** Ensured that the system satisfies the functional requirements.

Table 4.14: Functional Requirement Verification

Functional requirement	Expected Function	Verification Result
User registration	Create account	Pass
User login	Authenticate user	Pass

Contact management	Contact storage	Pass
Distress detection	Emergency voice detection	Pass
Motion detection	Abnormal motion detection	Pass
Fall detection	Fall identification	Pass
Crash detection	Crash identification	Pass
GPS tracking	Coordinate acquisition	Pass
SOS notification	Emergency alert notification	Pass

4.5.3 Testing of User Interfaces

The user interface tests included ensuring consistency, usability, responsiveness, and navigability. In testing the user interfaces, the major areas of concern were layout consistency, validation, navigability, and usability.

Examples of UI testing carried out:

A. Testing of Registration Screen: To ensure that account creation is easy by providing:

i. Validating email address
ii. Requirement of password
iii. Validating required fields

Expected outcome: clear validation and successful registration
Result: Passed

B. Testing of Login Screen: The test involved:

i. Authentication credentials check
ii. Error handling
iii. Session management

Expected outcome: successful authentication and error messages
Result: Passed

C. Dashboard Testing: The dashboard is the central control panel for the following features:

i. Data display
ii. Navigation links
iii. Live status of sensors

Expected outcome: information display and navigation
Result: Passed

D. Emergency Contacts Screen Testing: For validating contacts adding, editing, and deleting

Expected outcome: proper contact management
Result: Passed

E. SOS Alert Screen Testing: For testing of the emergency alert interface, such as:

i. Generating alert
ii. Displaying location
iii. Preview of notification

Expected outcome: display of SOS information
Result: Passed

4.6 Results Presentation and Analysis

This part describes the outputs that result from the development of the Intelligent Mobile-Based Emergency Detection and Automated SOS System and provides an analysis of the system based on the objectives of this study. This section will show the practical working of the proposed system and how effectively it works towards detecting emergencies and initiating automated actions.

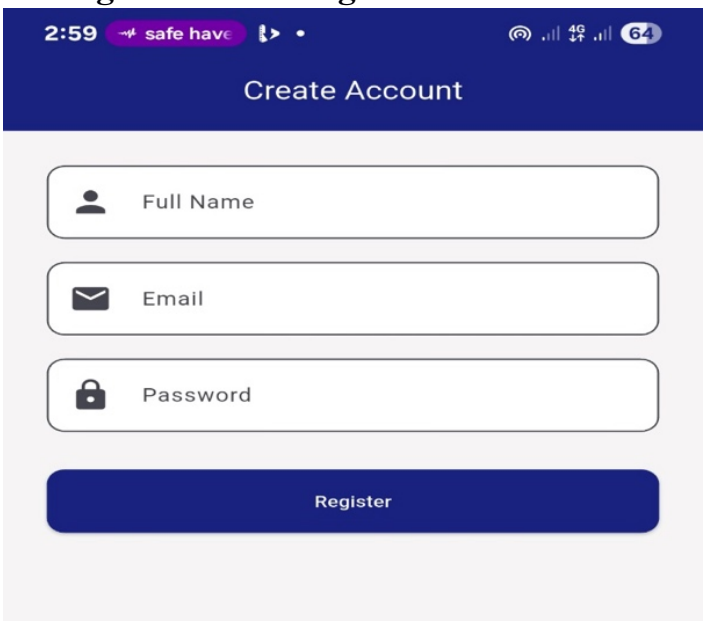
The findings in this part are obtained from the prototypical development of the intelligent mobile-based emergency detection and automated SOS system using Flutter, Python, Scikit-Learn, Firebase, SQLite, and smartphone sensors. The system was designed in a way that it detects emergencies through the processing of several data inputs like distress voices, abnormal motions, falls, and car accidents.

4.6.1 Sample Output and Interface Screens

The application provides interface for interaction of users, detection of emergencies, and sending automatic SOS alerts. The following interfaces have been provided along with their respective outputs.

Interface for User Registration: This serves as an initial interface where the users register themselves to get the service of personal emergency monitoring and SOS alerts.

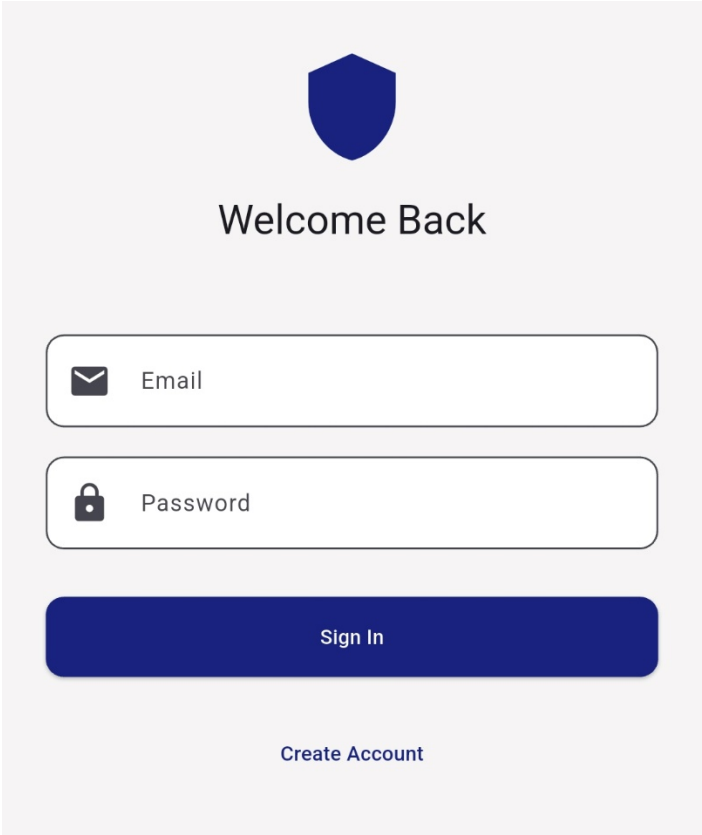
Fig 4.13: User Registration Interface



User Login Interface

The user is able to log into the system using the login interface.

Figure 4.14: User Login Interface

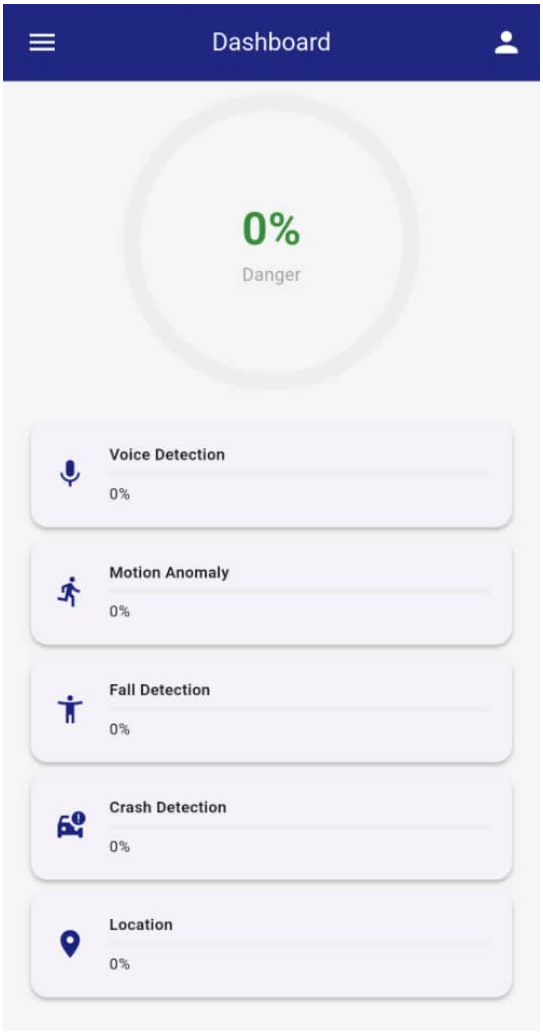


Dashboard Interface

The dashboard acts as the central control panel for the application, featuring the following capabilities:

i. Sensor Monitoring Status
ii. Emergency Detection Status
iii. Emergency Contact Details
iv. SOS Setting Configurations
v. Live Monitoring Details

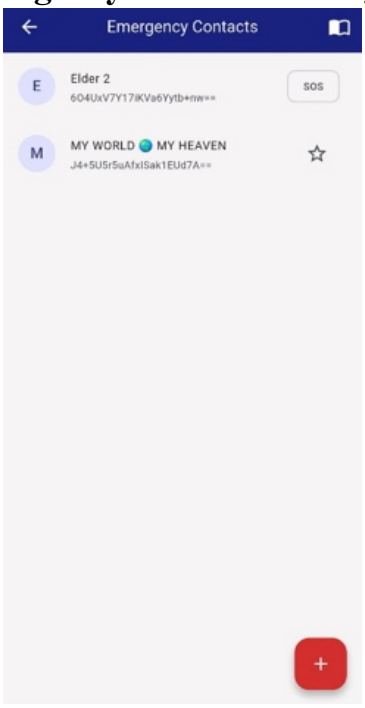
Fig 4.15: System Dashboard Interface



Emergency Contact Management Interface:

The Emergency Contact Interface enables the management of people who will be notified in case of emergencies.

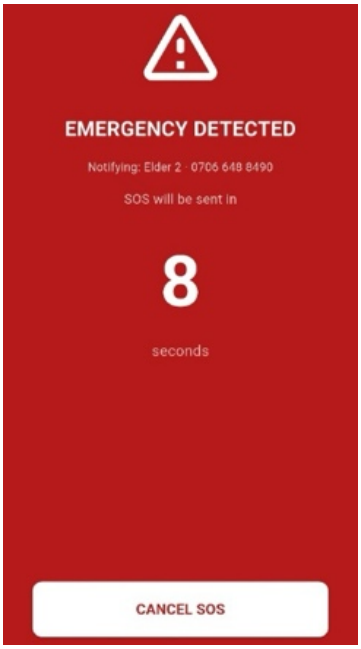
Fig 4.16: Emergency Contact Management Interface



Emergency Detection Interface:

The Emergency Detection Interface shows the output classification result of the Random Forest algorithm.

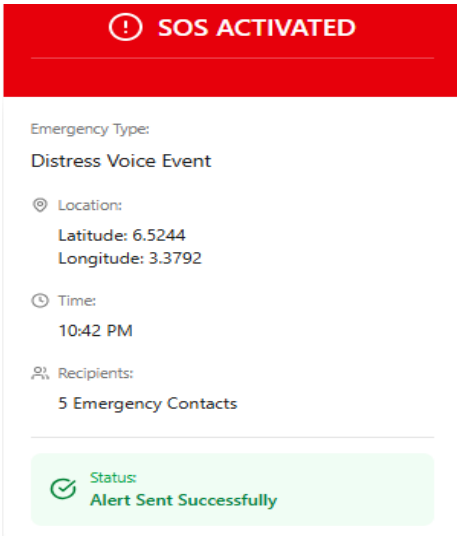
Fig 4.17: Emergency Detection Interface



Automated SOS Alert Interface:

In case of confirmed emergency, the SOS Alert Interface becomes operational.

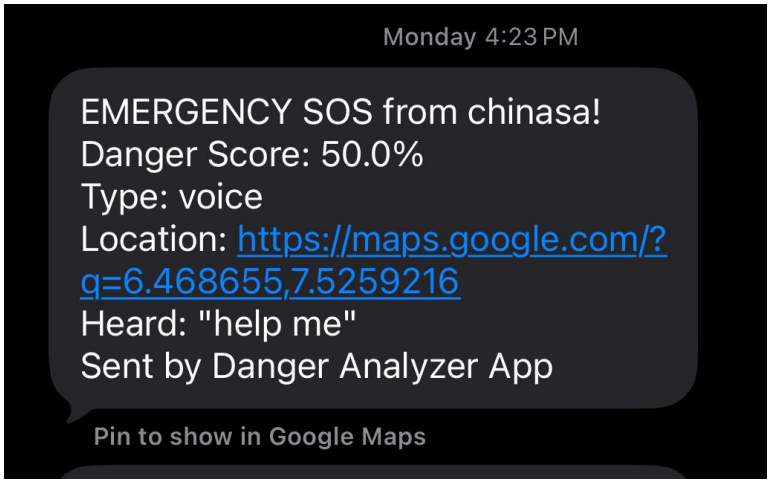
Fig 4.18: Automated SOS Alert Interface



Sample Emergency Notification Message

Below is an example of a typical SOS notification message.

Fig 4.19: Sample Emergency Notification Message interface



4.6.2.1 System Performance in General

The Intelligent Mobile-Based Emergency Detection and Automated SOS System effectively meets its objective of detecting emergencies and providing an automated response to it. The prototype combines smartphone sensors, machine learning, hybrid databases, and automation to create one single system that monitors the situation in real time. The performance is effective as observed during the tests of the prototype. A

Random Forest Classifier detected user actions and triggered the emergency response in case of a fall using sensor data. Offline functionality through SQLite and online database synchronization through Firebase Cloud Firestore contributed to effectiveness both when offline and online. Although this prototype currently detects only fall events, the modularity allows future implementation of distress voice detection, motion anomalies, car crashes, and sensor fusion from multiple sources. Generally speaking, the results provide a practical, scalable, and intelligent solution for real-time mobile-based emergency detection and SOS services.

4.6.2.2 ML Performance Analysis

Random Forest successfully classifies normal behavior versus emergencies based on the phone sensor readings. Based on the values presented in Section 4.5, namely accuracy, precision, recall, and F1-score, the classification algorithm demonstrates good results with balance between emergency classification and false alarm reduction.

A confusion matrix is consistent with these findings since the majority of activities are classified correctly without mistakes. Thus, one can conclude that selected features and Random Forest are appropriate for real-time emergency movements on smartphones. In general, the algorithm shows high accuracy and computational efficiency required for the prototype. Therefore, the proposed emergency detection and automated SOS system is reliable (Al Mudawi, N., Azmat, U., Alazeb, A., Alhasson, H. F., Alabdullah, B., Rahman, H., Liu, H., & Jalal, A. (2025).; John, A., Cardiff, B., & John, D. (2024)).

4.6.2.3 Functional Performance Analysis

Analysis indicates that the emergency detection system successfully merges all primary functionalities into one responsive system. The prototype was able to conduct constant sensor monitoring, real-time activity classification using Random Forest, emergency detection, location retrieval, and generation of SOS alerts according to specifications.

Fall detection system as the primary functionality was able to recognize the events of falling down using information from accelerometers and gyroscopes of the smartphone. After the detection of the event, the system initiated an emergency response that included alerting, location identification, and notifying emergency contacts according to the set up in advance.

Voice distress, motion anomalies, car crashes detection, and sensor fusion are not yet developed but possible in the further development stage due to the software architecture of the system.4.6.2.9 Limitations of the Proposed System

Despite its strengths, the system possesses certain limitations.

These include:

i. Dependence on smartphone sensor availability.
ii. Potential environmental noise interference.
iii. Battery consumption associated with continuous monitoring.
iv. Dependence on network connectivity for cloud synchronization.
v. Limited availability of real-world emergency datasets.

Future work may address these limitations through improved optimization techniques and expanded datasets.

4.6.2.4 Strengths, Weaknesses and Future Enhancements

The Intelligent Mobile-based Emergency Detection and Automated SOS system leverages smartphones' sensors, machine learning techniques, hybrid databases and automated SOS alerts to detect emergencies. The Random Forest model ensures accurate and efficient emergency classification with minimum consumption of resources of the Android mobile platform.

Nevertheless, there are certain weaknesses in the prototype of the application under analysis. The machine learning model is focused only on the problem of falls; further development will include distress voice, motion anomalies, car crashes, and other functions. Moreover, the system performance will depend on the quality of the sensors and environmental settings, as well as on the size of the validation dataset used.

In general, the described model has enough strengths and a good base for future scaling of the application to be utilized in various cases. The modular architecture will make it possible to introduce additional models and AI algorithms in the future.

CHAPTER FIVE

SUMMARY, CONCLUSION, AND RECOMMENDATIONS

5.1 Introduction

This chapter describes the summary, conclusions, and recommendations of the research study "Development of an Intelligent Mobile-Based Emergency Detection and Automated SOS System." In this chapter, the process followed to complete the research study is outlined through the identification of the activities undertaken in order to achieve the set goals. The chapter will further outline the conclusions drawn from the findings of the study with respect to how well the developed system met the stated aims and objectives. Recommendations are further made to guide other researchers, software developers, emergency response agencies, and other interested parties in developing their own intelligent emergency detection and response systems.

5.2 Summary of the Study

There has been an increase in instances of personal safety threats, accidents, medical emergencies, and delayed emergency responses, leading to an urgent need for the development of intelligent emergency detection systems that can automatically detect and alert the right people in case of any danger. Many of the current emergency alarm systems require manual activation by the victim who might not be in a position to do so in cases where there is an accident, fall, physical assault, or in case he/she has lost consciousness. The problem led to the development of the Intelligent Mobile-Based Emergency Detection and Automated SOS System that employs the use of smartphones' sensors and automated communication technologies to automatically detect emergencies and notify emergency contacts.

In Chapter One, the problem statement was presented in order to introduce the research problem. In addition, the chapter described the problems associated with existing systems as well as the need to come up with an intelligent system. In the same chapter, the aim of the study was clearly specified. Objectives were then outlined and the significance of the study was explained.

In Chapter Two, literature was reviewed on emergency detection systems, smartphone sensors, machine learning, mobile applications, and automated alerts. Fall detection, distress voice detection, activity recognition, and accident detection were covered, with particular focus on the Random Forest algorithm because of its efficiency, interpretability, and classification with sensors. From the review, there were numerous

ways of solving emergencies, but all focused on single emergency cases and lacked multi-modal detections.

In Chapter Three, system analysis and design were discussed. Emergency response methods currently employed depended on manually activating the process, were very slow, had few automations, and lacked multi-sensor integration. A new intelligent emergency detection system was then proposed and Functional and Non-Functional Requirements were provided in the form of System Requirements Specification (SRS). Agile approach was adopted when developing the system and Object-Oriented Analysis and Design was employed to model the system. Use Case, Activity, and Class UML diagrams provided an overview of the design.

Chapter Four addressed implementation and testing of the system. The tools used to develop the emergency detection app included Flutter 3.41.9 for mobile apps, Python 3.12.8 for ML, and Android Studio and Visual Studio Code as integrated development environments. In addition, Jupyter Notebook was used in training models. The emergency detector comprised a Random Forest Classifier using a hybrid dataset (SisFall, MobiAct) plus a custom validation set. Data preprocessing included cleaning, normalization, feature extraction, data augmentation, and train-test splitting. Modules that were developed were user authentication, emergency contacts management, distress voice detection, motion anomaly detection, fall detection, vehicle crash detection, GPS location detection, and automated SOS message creation and sending.

In conclusion, the study has been able to design an intelligent mobile-based emergency detection and automated SOS system that is capable of monitoring user activities, detecting potential emergencies and sending out automated SOS notifications with location details.

5.3 Conclusion

The objective of this research was to build an intelligent emergency detection and automated SOS system by identifying emergencies and sending SOS notifications. Based on findings, the proposed system succeeds in achieving the desired goals as outlined above. First, the analysis of current systems and their limitations was performed. Second, it involved the use of sensors (motion, GPS, sound) from smartphones to detect emergencies. Third, it included the development of an AI-based danger detection system using a Random Forest Classifier.

The evaluation of the developed system indicates that there is huge potential: detection of distress voice, detection of abnormal motions/falls, crash detection, GPS tracking, and automated sending of SOS notifications increases comprehensiveness. The

results obtained show that the Random Forest classifier can effectively differentiate between normal states and emergencies, thus allowing successful emergency detection.

Recommendations

With respect to the findings and results from the above experiment, the following recommendations could be made to improve the effectiveness and development of intelligent emergency detection systems.

Firstly, future studies should focus on increasing the training datasets. Increasing the real-world dataset of emergencies is expected to improve the robustness, generalizability, and classification performance of the machine learning model. Special attention needs to be paid to the production of different age ranges, environments, and scenarios.

Secondly, future versions of the system should apply sophisticated deep learning techniques, including CNNs, LSTM networks, and Transformers. Application of the mentioned techniques could bring improvement in distress voice recognition, activity recognition, and sensor fusion.

Thirdly, the distress voice detection part of the system should be able to recognize emergencies in several languages and dialects. In cases where there is a need to recognize emergencies in multilingual settings such as Nigeria, the ability of recognizing emergencies in English, Igbo, Hausa, Yoruba, and Nigerian Pidgin English is highly beneficial.

Fourthly, further versions of the system should be wearable devices such as smartwatches and fitness trackers. With wearable technology, more physiological data such as heart rate, blood oxygen saturation, and body temperature can be detected thus improving the accuracy of detecting emergencies.

Fifthly, integration of the system with emergency response services, hospitals, security agencies, and disaster management organizations needs to be done.

REFERENCE

Al Mudawi, N., Azmat, U., Alazeb, A., Alhasson, H. F., Alabdullah, B., Rahman, H., Liu, H., & Jalal, A. (2025). IoT powered RNN for improved human activity recognition with enhanced localization and classification. *Scientific Reports*, *15*, 10328.

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32.

John, A., Cardiff, B., & John, D. (2024). *A review on multisensory data fusion for wearable health monitoring*.

Kundu, S., Mallik, M., Saha, J., & Chowdhury, C. (2024). Smartphone based human activity recognition irrespective of usage behavior using deep learning technique. *International Journal of Information Technology*.

Mohsin, A. S. M., & Muyeed, M. A. (2024). IoT based smart emergency response system (SERS) for monitoring vehicle, home and health status. *Discover Internet of Things*, *4*(22).

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Pressman, R. S., & Maxim, B. R. (2020). *Software engineering: A practitioner's approach* (9th ed.). McGraw-Hill.

Russell, S. J., & Norvig, P. (2021). *Artificial intelligence: A modern approach* (4th ed.). Pearson.

Sakalauskas, L., & Vaiciulyte, I. (2025). Adaptive learning approach for human activity recognition using data from smartphone sensors. *Applied Sciences*, *15*(14), 7731.

Zhong, Z., & Liu, B. (2025). Efficient human activity recognition using machine learning and wearable sensor data. *Applied Sciences*, *15*(8), 4075.

Zhou, H., Zhang, X., Feng, Y., Zhang, T., & Xiong, L. (2025). Efficient human activity recognition on edge devices using DeepConv LSTM architectures. *Scientific Reports*, *15*, 13830.

Appendices

Appendix A
Code Listing

Appendix B
Mobile Application Screenshots

Appendix C
Database and Firebase Collections Schema

Appendix D
Random Forest Classifier and its Hyperparameter Configuration

Appendix E
UML Diagrams (Use Case, Activity and Class Diagram)

Appendix F
Emergency Detection Outputs and Example Test Cases

Appendix G
Dataset Description and Data Collection Tools