ADVANCED HUMAN LIFE DETECTION SYSTEM FOR DISASTER RESCUE USING MULTI-MODAL SENSOR FUSION

A PROJECT REPORT

Submitted by

ARUL PRAKASH G	**(711621106002)**
MAHARAJA S	**(711621106019)**
VIJAYARAGAVAN S	**(711621106038)**

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING (B.E)

in

DEPARTMENT OF

ELECTRONICS AND COMMUNICATION ENGINEERING

KATHIR COLLEGE OF ENGINEERING, COIMBATORE

(Autonomous)

ANNA UNIVERSITY: CHENNAI 600 025

MAY 2025

KATHIR COLLEGE OF ENGINEERING, COIMBATORE

(Autonomous)

BONAFIDE CERTIFICATE

Certified that this project report **"ADVANCED HUMAN LIFE DETECTION SYSTEM FOR DISASTER RESCUE USING MULTI-MODAL SENSOR FUSION "**is the bonafide work of

ARUL PRAKASH G	**(711621106002)**
MAHARAJA S	**(711621106019)**
VIJAYARAGAVAN S	**(711621106038)**

who carried out the project work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of other dissertation on the basics of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

……………………………………

Dr. N.K. Anushkannan, M.E, Ph.D.,
HEAD OF THE DEPARTMENT
Professor & Head,
Department of ECE,
Kathir College of Engineering,
Neelambur,
Coimbatore-641062.

SIGNATURE

…………………………………

Mr.R.Nandhakumar,M.E,
SUPERVISOR
Assistant Professor,
Department of ECE,
Kathir College of Engineering,
Neelambur,
Coimbatore-641062.

The candidates with university register number ………………………….. was examined in the Project viva-voce examination held on………………

……………..………

INTERNAL EXAMINER

…………………………..

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

ACKNOWLEDGEMENT

The success of a work depends on a team and cooperation. We take this opportunity to express our gratitude and thanks to everyone who helped us in our project. We would like to thank the management for excellent facilities and the constant support provided by them to complete this project.

It is indeed our great honor bounded duty to thank our Respected Chairman **Thiru. E. S. KATHIR,** for his academic interest shown towards the students. We are indebted to our beloved Secretary **Tmt. LAVANYA KATHIR,** for motivating and providing us with all facilities. We wish to express our sincere regards and deep sense of gratitude to **Dr. R. UDAIYAKUMAR, M.E, Ph.D.,** Principal, for the excellent facilities and encouragement provided during the course of the study and project.

We are indebted to **Dr. N. K. ANUSHKANNAN, M.E, Ph.D.,** Professor & Head/ECE, for having permitted us to carry out this project and giving the complete freedom to utilize the resources of the department. We express our sincere thanks to our project guide **Mr. R. NANDHAKUMAR, M.E,**Assistant Professor /ECE, for his valuable guidance and encouragement given to us for this project.

We solemnly express our thanks to all the teaching and non-teaching staff of the Electronics and Communications Engineering Department, family and friends for their valuable support which inspired us to work on this project.

ABSTRACT

ABSTRACT

Disaster rescue operations often face significant challenges in rapidly and accurately locating victims trapped under debris, as traditional and single-sensor methods are limited by environmental noise and obstructions. To address these challenges, this project presents a portable, real-time human life detection system that combines a 24GHz mmWave radar sensor, a high-sensitivity microphone, and an accelerometer to capture micro-movements, faint sounds, and vibrations associated with trapped survivors. These sensors are interfaced through an MCP3008 ADC and managed by a Raspberry Pi 4, which serves as the core processing and control unit.

Advanced signal processing techniques, such as Fast Fourier Transform (FFT) and Discrete Wavelet Transform (DWT), are used to extract meaningful features from the raw sensor data. Embedded machine learning algorithms, including Support Vector Machine (SVM) and Random Forest, analyze these features to robustly classify human presence and minimize false alarms, even in noisy and cluttered environments. Real-time alerts are delivered via an LCD, LEDs, and a buzzer, providing immediate feedback to rescue teams. The system is designed for low power consumption and portability, making it ideal for rapid deployment in disaster zones. Experimental results demonstrate that the multi-sensor fusion approach significantly improves detection accuracy and reliability compared to conventional single-sensor solutions.

TABLE OF CONTENTS

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF FIGURES

LIST OF TABLES

LIST OF TABLES

LIST OF ABBREVIATIONS

LIST OF ABBREVIATION

ADC	-	Analog-to-Digital Converter
DWT	-	Discrete Wavelet Transform
FFT	-	Fast Fourier Transform
GPIO	-	General Purpose Input/Output
I2C	-	Inter-Integrated Circuit
LCD	-	Liquid Crystal Display
LED	-	Light Emitting Diode
LoRa	-	Long Range
ML	-	Machine Learning
Pi	-	Raspberry Pi
RMS	-	Root Mean Square
SVM	-	Support Vector Machine
SPI	-	Serial Peripheral Interface
UART	-	Universal Asynchronous Receiver-Transmitter
ZCR	-	Zero Crossing Rate
THD	-	Total Harmonic Distortion
OS	-	Operating System
IoT	-	Internet of Things

CHAPTER 1

INTRODUCTION

INTRODUCTION

1.1 BACKGROUND

Natural disasters such as earthquakes, landslides, and building collapses often result in people being trapped under rubble and debris. Rapid detection of trapped survivors is critical to saving lives and reducing casualties. Traditional search and rescue operations rely on manual searches, trained dogs, or single-sensor devices, which are often time-consuming, resource-intensive, and limited by environmental factors such as dust, smoke, and unstable debris. Rescue workers searching for survivors in a collapsed building or disaster scenario, highlighting the urgency as shown in Figure 1.1

Figure 1.1 Disaster rescue operation highlighting the need for rapid human life detection

Recent advances in embedded systems and machine learning have opened new avenues for developing intelligent, portable, and robust human life detection systems. These systems aim to provide real-time, accurate detection of human presence by fusing data from multiple sensors, enabling rescue teams to locate survivors quickly and efficiently.

This project focuses on designing and implementing a multi-sensor embedded system that integrates radar, acoustic, and vibration sensors with embedded machine learning algorithms to detect human life signs such as breathing, movement, and faint sounds under debris. The system is intended to be low-cost, portable, and capable of real-time operation in harsh disaster environments.

1.2 EMBEDDED SYSTEMS IN HUMAN LIFE DETECTION

An embedded system is a dedicated computing device designed to perform specific tasks within a larger system, often operating under real-time constraints. It typically consists of a microcontroller or microprocessor, memory, input/output interfaces, and specialized software or firmware optimized for the target application.Block diagram of a simple embedded system (with microcontroller, sensors, input/output) as shown in Figure 1.2

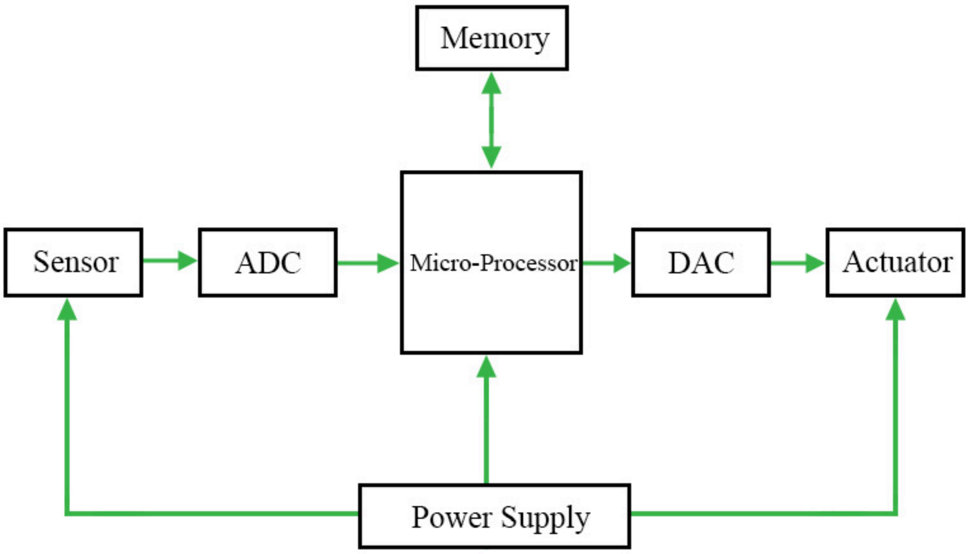

Figure 1.2 Block diagram of an embedded system for sensor-based applications

In the context of human life detection, embedded systems serve as the core processing units that acquire sensor data, perform signal processing, execute machine learning algorithms, and provide immediate feedback or alerts to rescuers. Their compact size, low power consumption, and ability to operate autonomously make them ideal for deployment in disaster zones.

1.2.1 Key Characteristics

Embedded systems used in life detection must possess several key characteristics as shown in Figure 1.3

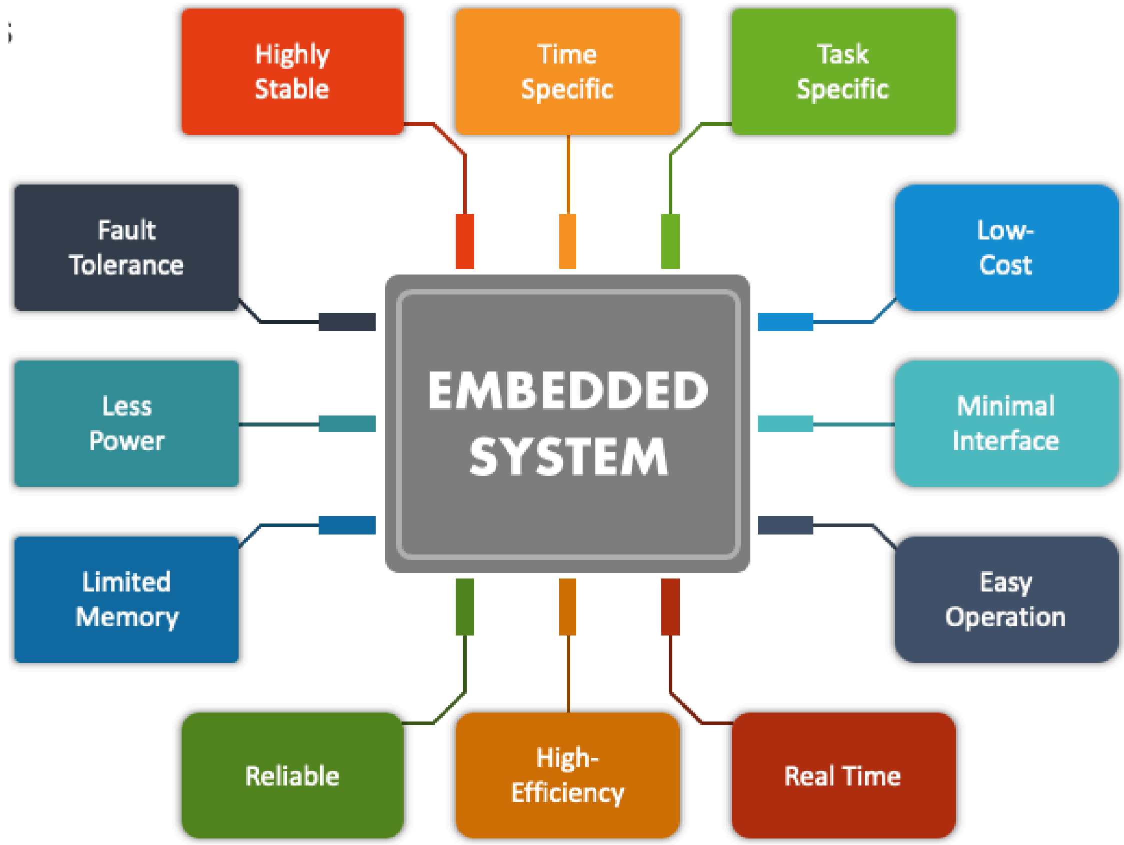

Figure 1.3 Key characteristics of embedded systems: real-time, low power, compact design

Real-Time Operation: The ability to process sensor inputs and generate alerts instantly is crucial for timely rescue.

Reliability and Robustness: Systems must operate reliably in challenging environments with dust, moisture, and electromagnetic interference.

Low Power Consumption: Battery-powered operation is often necessary in field deployments.

Compactness and Portability: Easy to carry and deploy in confined or unstable spaces.

Scalability and Flexibility: Ability to integrate multiple sensor types and adapt to different disaster scenarios.

1.2.2 Applications in Disaster Management

Embedded systems have been widely adopted in disaster management for structural health monitoring, environmental sensing, communication, and now increasingly for human life detection. Their ability to perform on-site data processing reduces dependency on remote servers or cloud infrastructure, which may be unavailable or unreliable during emergencies as shown in Figure 1.4

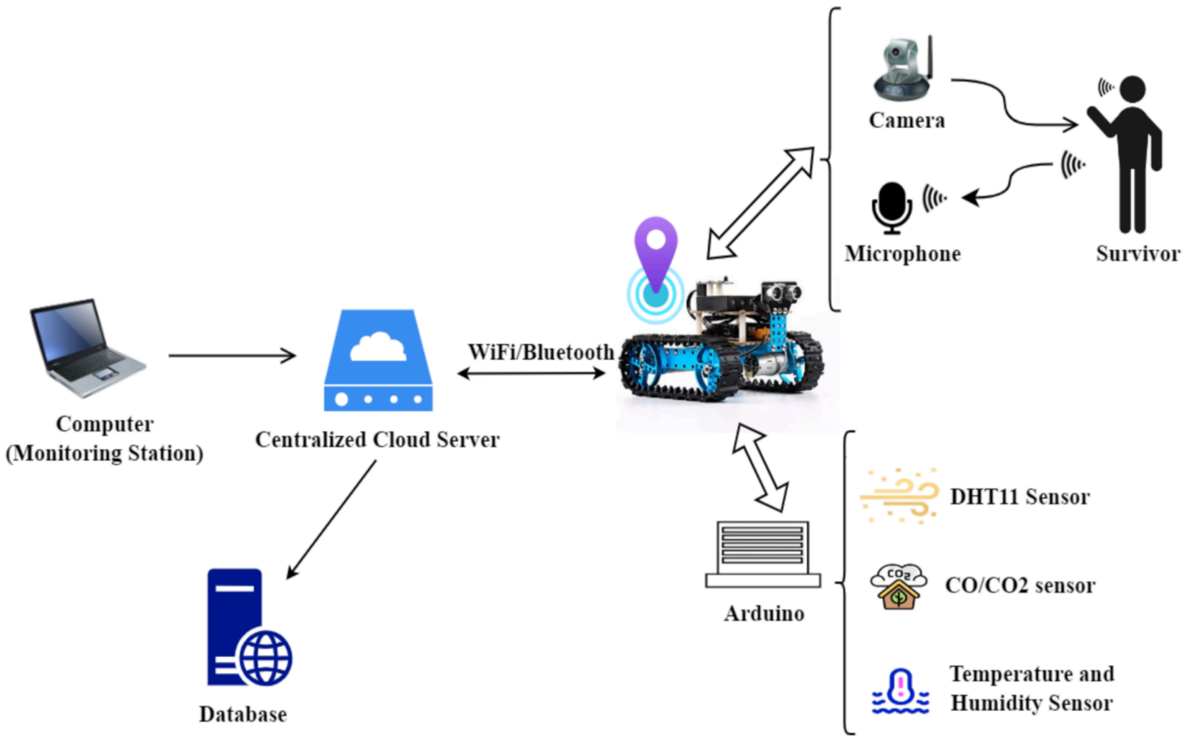

Figure 1.4 Applications of embedded systems in disaster management and rescue operations

1.3 MACHINE LEARNING FOR HUMAN LIFE DETECTION

Machine learning (ML) is a subset of artificial intelligence that enables systems to learn patterns from data and make decisions or predictions without explicit programming. ML algorithms can improve detection accuracy by learning complex patterns in sensor data and distinguishing between human presence and environmental noise.conceptual overview of Machine Learning as shown in Figure 1.5

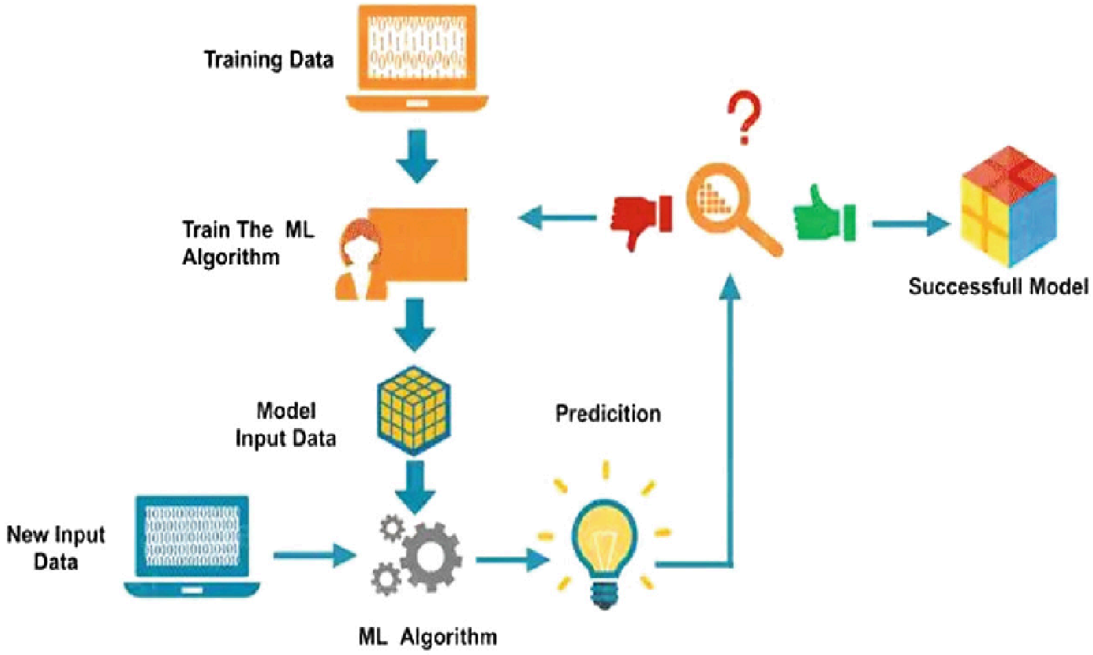

Figure 1.5 Conceptual overview of machine learning in embedded systems

1.3.1 Characteristics Relevant to Embedded Systems

Data-Driven: ML models improve with more data, enabling adaptation to diverse environments.

Non-Linear Pattern Recognition: Capable of identifying subtle signals such as micro-movements or faint sounds.

Lightweight Models: Embedded ML focuses on efficient algorithms suitable for resource-constrained hardware.

Real-Time Inference: Enables instant decision-making critical for rescue operations.

1.3.2 Applications in Life Detection

ML techniques such as Support Vector Machines (SVM), Random Forests, and Neural Networks have been successfully applied to classify sensor data for life detection. These models analyze features extracted from radar signals, audio recordings, and vibration data to detect signs of life with high accuracy.

1.4 MULTI-SENSOR FUSION: TECHNOLOGY AND METHODOLOGY

Multi-sensor fusion combines data from diverse sensors to improve the accuracy and reliability of human life detection in complex environments. By integrating radar, acoustic, and vibration sensors, the system leverages complementary strengths to overcome individual sensor limitations. Advanced signal processing and machine learning techniques are employed to analyze fused data in real time, enabling robust detection under debris and noisy conditions.

1.4.1 Sensor Modalities

The proposed system integrates three complementary sensors:

24 GHz mmWave Radar (HLK-LD1125H): Detects micro-movements such as breathing and heartbeat through debris.

MAX4466 Microphone: Captures faint acoustic signals like tapping or calls for help.

MPU-6050 Accelerometer: Senses vibrations and knocks indicative of human activity.

1.4.2 Data Acquisition and Signal Processing

The process of data acquisition and signal processing is fundamental for transforming raw sensor outputs into actionable information in a human life detection system. Initially, analog signals from various sensors-such as the radar module and high-sensitivity microphone-are digitized using the MCP3008 analog-to-digital converter (ADC). This ADC provides 10-bit resolution and supports multiple input channels, enabling simultaneous high-resolution digitization of signals from different sensors. The digitized data is then fed into the Raspberry Pi 4, which acts as the central processing hub for subsequent operations.

Through this integrated pipeline-comprising synchronized data acquisition, targeted noise reduction, and sophisticated signal processing-the system is able to reliably extract vital features from complex, real-world sensor data. This robust approach is essential for achieving accurate, real-time detection of human life signs in challenging disaster environments as shown in Figure 1.6 Signal processing pipeline for extracting features from sensor data.

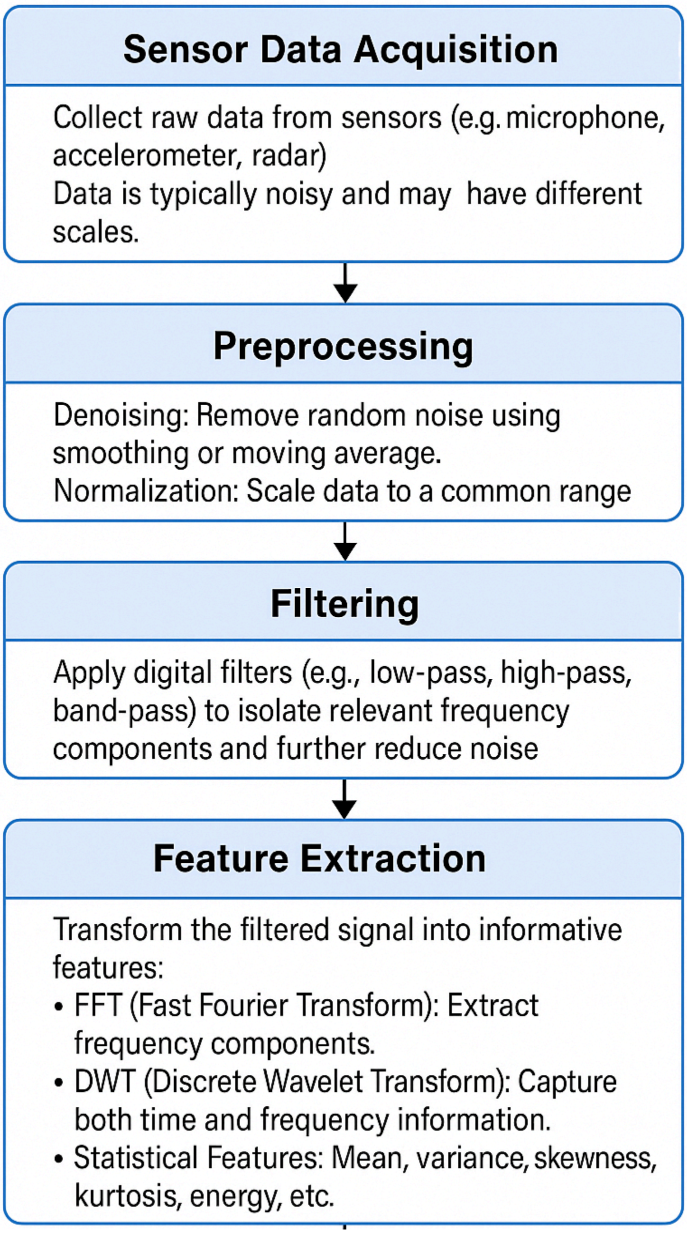

Figure 1.6 Signal processing pipeline for extracting features from sensor data.

1.4.3 Machine Learning Integration

Extracted features are input to embedded ML classifiers (SVM, Random Forest) running on the Raspberry Pi. The models are trained to distinguish human presence signals from environmental noise and false triggers.

1.4.4 Real-Time Alerts and User Interface

The system provides immediate feedback via a 16x2 LCD display, buzzer, and LED indicators, enabling rescuers to quickly identify detected survivors without external communication infrastructure.

1.5 SYSTEM ARCHITECTURE AND COMPONENTS

The system integrates radar, microphone, and accelerometer sensors with a Raspberry Pi for real-time human life detection. Immediate alerts are provided via LCD, buzzer, and LEDs, making the solution portable and effective for disaster rescue.

1.5.1 Hardware Components and Software Tools

The system integrates the HLK-LD1125H radar for micro-movement detection, MAX4466 microphone for capturing faint sounds, and MPU-6050 accelerometer for sensing vibrations, with the MCP3008 ADC converting analog signals to digital. All sensor data is processed by the Raspberry Pi 4 Model B, which also manages real-time alerts through a 16x2 LCD, buzzer, and LED indicators. The software runs on Raspberry Pi OS and is developed in Python, utilizing NumPy and SciPy for signal processing and Scikit-learn for machine learning tasks, ensuring efficient data handling and reliable human detection during disaster rescue operations.

1.6 IMPACT AND SIGNIFICANCE OF THE PROJECT

The proposed embedded multi-sensor life detection system addresses critical gaps in current rescue technologies by providing:

Enhanced Detection Accuracy: Multi-sensor fusion reduces false positives and missed detections.

Real-Time Operation: Instant alerts speed up rescue efforts.

Portability and Cost-Effectiveness: Affordable, compact design suitable for rapid deployment.

Environmental Robustness: Effective in noisy, cluttered, and obstructed disaster environments.

CHAPTER 2

LITERATURE SURVEY

LITERATURE SURVEY

2.1 INTRODUCTION

Human life detection during disasters like earthquakes and landslides is difficult due to the slow and limited effectiveness of manual methods. Recent research shows that combining radar, acoustic, and vibration sensors with machine learning significantly improves accuracy and reduces false positives. This review highlights current system strengths and gaps, guiding the design of a more accurate and responsive multi-sensor detection system for real-world rescue scenarios.

2.2 EXISTING TECHNOLOGIES

Karthikeyan et al. (2018) [1] utilized UWB radar coupled with data mining techniques to detect micro-movements indicative of life. The high resolution of UWB radar enables non-contact detection; however, challenges remain regarding signal attenuation in dense debris and system cost.

Zhang et al. (2018) [2] combined CO_2 sensors, thermal cameras, and microphones with Support Vector Machine (SVM) classifiers to improve detection accuracy. This multi-modal approach reduced false positives but was limited by environmental noise and sensor reliability.

Balbin et al. (2019) [3] developed a system combining pulse sensors and force sensing resistors integrated with wireless communication for detecting buried victims. Their system demonstrated the feasibility of physiological signal detection but was limited by environmental factors and sensor placement.

Wulandari et al. (2020) [4] used thermal imaging on mobile robots, extracting Histogram of Oriented Gradients (HOG) features and applying machine learning classifiers for victim detection. The system was computationally efficient but suffered reduced accuracy in cluttered environments.

Dong et al. (2021) [6] integrated thermal imaging with unmanned aerial vehicles (UAVs) and deep learning to scan large disaster areas for survivors. Although effective in open environments, thermal imaging is less reliable when victims are deeply buried or obscured by debris.

Aldana-Franco et al. (2021) [7] developed an intelligent search system using biometric sensors and artificial neural networks (ANN), enabling precise detection and reducing rescuer risk. Embedded machine learning allowed real-time processing on portable devices.

Bagaskara et al. (2021) [5] showed that Ground-Penetrating Radar (GPR) can effectively detect buried human bodies by identifying amplitude contrast anomalies in radar signals, especially for fresher remains. Using a 700 MHz antenna and MATLAB-based noise reduction, their cemetery simulations demonstrated GPR's potential for improving disaster victim recovery in scenarios like landslides and earthquakes.

Ahmed et al. (2022) [8] proposed a Frequency-Modulated Continuous Wave (FMCW) radar system integrated with convolutional neural networks (CNN) for real-time human activity recognition. Their approach showed robustness to environmental variations but was limited to predefined activities and sensitive to radar positioning.

Murali et al. (2024) [9] proposed a real-time detection system integrating motion detectors, infrared cameras, and ground-penetrating radar (GPR) sensors. Their multi-sensor fusion approach enhanced detection accuracy but required complex hardware and infrastructure.

Niyaz et al. (2024) [10] employed Very High Frequency (VHF) and Ultra High Frequency (UHF) electromagnetic waves analyzed by artificial neural networks (ANN) for life detection, achieving high accuracy in simulated debris conditions, though with complexity and frequency limitations.

2.3 COMPARATIVE ANALYSIS

Table 2.1 Comparative Analysis of Human Life Detection Technologies in Disaster Rescue.

Reference	Technology	Method	Algorithm	Key Features	Strengths	Limitations
Karthikeyan et al. (2018)[1]	UWB radar	Data mining	Nil	Detects vital signs	Non-contact, high resolution	Expensive, low SNR in debris
Zhang et al. (2018)[2]	CO_2, Thermal, Mic	Multi-sensor fusion, Voice	SVM	Sensor fusion, voice recognition	Reduces false positives, narrows search	Gas sensors unreliable, thermal blocked
Balbin et al. (2019)[3]	Pulse, FSR, XBee, GPS	Real-time wireless, GPS loc.	Nil	Heartbeat/pressure detection	Real-time, supports rescue ops	GPS unreliable indoors, placement critical
Wulandari et al. (2020)[4]	Robot, Thermal Imaging	Feature extraction	HOG, SVM, RF	FLIR camera, autonomous	Efficient, low computation	Terrain/thermal limits, non-radiometric
Bagaskara et al. (2021)[5]	GPR	Amplitude contrast analysis	Nil	Subsurface imaging	Non-invasive, buried body detection	Costly, soil/moisture sensitive, complex
Dong et al. (2021)[6]	UAV, Thermal Imaging	Deep learning, Real-time	CNN	UAV scan, thermal camera	Wide-area, automated, rapid coverage	Weather/obstacle sensitive, battery limits
Aldana-Franco et al. (2021)[7]	Biometric sensors, IoT	Remote monitoring	ANN	Temp, CO_2, humidity, distance	Precise, reduces rescuer risk	Network dependent, comms may fail
Ahmed et al. (2022)[8]	FMCW radar	Activity recognition	CNN	Human activity detection	Robust, real-time, non-invasive	Predefined activities, radar position sensor.
Murali et al. (2024)[9]	5G multi-sensor	Sensor fusion, 5G data transfer	Nil	Real-time, multimodal, subsurface map	Accurate, real-time	Needs 5G, high cost, power consumption
Niyaz et al. (2024)[10]	VHF/UHF	Frequency analysis	ANN	EM simulation, respiration detection	High accuracy in simulation	Frequency constraints, simulated only

2.4 Problem Statement

Traditional disaster rescue methods are often slow and unreliable in detecting victims trapped under debris, mainly due to the limitations of single-sensor systems and challenging environments. Existing solutions struggle with low sensitivity in noisy conditions and frequently fail to detect deeply buried or unconscious victims. The lack of real-time data processing and alert mechanisms further delays critical rescue actions, reducing survival chances. There is a clear need for a portable, real-time system that can accurately identify human presence beneath rubble, minimize false alarms, and provide immediate alerts to rescue teams.

2.5 Solution

This project introduces a portable, battery-powered human life detection system that uses mmWave radar, a sensitive microphone, and a three-axis accelerometer to sense motion, sound, and vibrations from trapped victims. Real-time sensor data is processed on a Raspberry Pi 4 using advanced signal processing (FFT, DWT) and machine learning (SVM, Random Forest) to accurately distinguish human presence from background noise. Alerts are delivered instantly via LCD, LEDs, and buzzer, enabling rapid, reliable rescue response in disaster environments while minimizing false alarms. The system is user-friendly and designed for quick deployment, making it highly practical for use in diverse and challenging field conditions.

2.9 Summary

This chapter reviewed various sensor-based human life detection technologies, highlighting their methods, strengths, and limitations. While radar, acoustic, and thermal sensors each have unique benefits, they also face environmental and operational challenges. Multi-sensor fusion with embedded machine learning stands out as a promising approach, offering improved accuracy and reliability for real-time victim detection in disaster scenarios.

CHAPTER 3

PROPOSED SYSTEM

PROPOSED SYSTEM

3.1 INTRODUCTION

The proposed system is an advanced, portable human life detection solution designed specifically for disaster rescue operations. Unlike traditional methods that rely on a single sensing modality and often suffer from slow response times and high false alarm rates, this system integrates radar, acoustic, and vibration sensing with embedded signal processing and machine learning. The multi-modal sensor fusion approach leverages the strengths of each sensor type, enabling the system to detect human presence even in challenging environments such as collapsed buildings, landslides, and earthquake zones. Real-time data acquisition, robust feature extraction, and intelligent classification ensure that rescue teams receive accurate and timely alerts, thereby increasing the chances of saving lives during critical rescue missions.

3.2 BLOCK DIAGRAM OF THE PROPOSED SYSTEM

The architecture of the proposed system is best visualized through its block diagram shown in Figure 3.1 which illustrates the flow of power and data from the sensors to the processing unit and finally to the user interface and alert mechanisms.

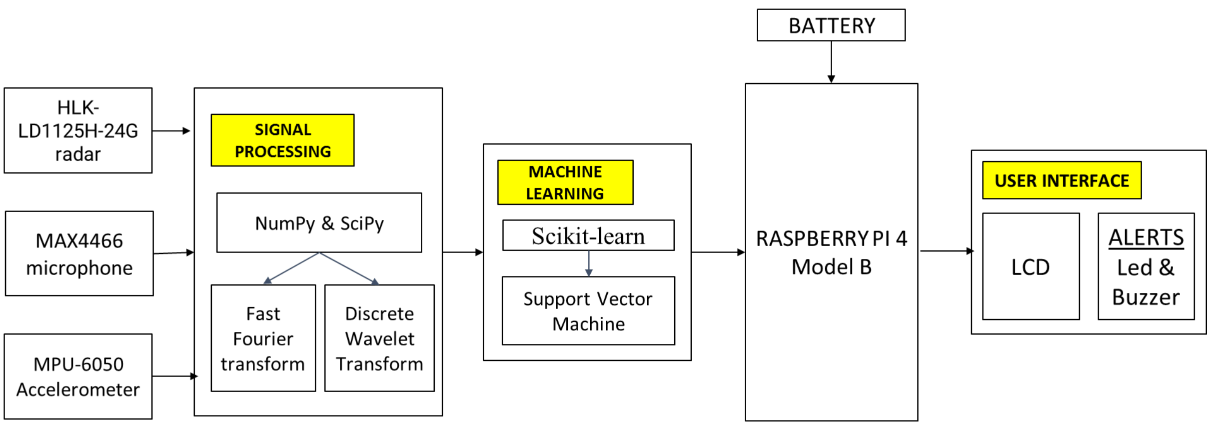

Figure 3.1 System Block Diagram

In this diagram, the battery serves as the primary power source for all components, ensuring system portability and independence from external power supplies. Sensor data from the radar, microphone, and accelerometer is funneled into the Raspberry Pi 4, which performs all signal processing, feature extraction, and machine learning inference. The outputs are then communicated to the user through a combination of LCD display, LEDs, and buzzer alerts.

3.3 BLOCK DIAGRAM EXPLANATION

The block diagram outlines the overall structure and data flow of the proposed system. It shows how each sensor and component connects to the processing unit, illustrating the sequence from data collection and signal processing to machine learning-based detection and user alerts. The following sections briefly describe the function of each block in the system.

3.3.1 Sensors

The HLK-LD1125H-24G is a 24GHz mmWave radar sensor capable of detecting human presence and micro-movements such as breathing, heartbeat, or slight body motions, even through dense debris. Its high-frequency electromagnetic waves penetrate obstacles, allowing for non-contact detection in scenarios where visual or acoustic signals may be blocked. The radar outputs digital signals indicating both the presence of movement and the estimated range to the detected object. Communication with the Raspberry Pi is handled via a UART/Serial interface, ensuring reliable and fast data transfer. The radar's robustness to environmental noise and its ability to sense through obstructions make it indispensable for disaster rescue operations.

The MAX4466 electret microphone is designed to capture faint acoustic signals, such as tapping, knocking, or cries from victims trapped under debris. Its high sensitivity and adjustable gain allow it to detect low-amplitude sounds that other microphones might miss. The analog output from the microphone is digitized using the MCP3008 10-bit Analog-to-Digital Converter (ADC), which provides accurate digital representation of the sound waveforms. This digitized data is then processed by the Raspberry Pi for further analysis. The microphone's wide frequency response (20Hz–20kHz) ensures that both low and high-frequency sounds are captured, increasing the likelihood of detecting human-generated noises.

The MPU-6050 is a 3-axis accelerometer and gyroscope module that senses vibrations and movements caused by human activity or knocking under rubble. It provides real-time acceleration data along the X, Y, and Z axes via an I2C communication interface. The sensor's high sampling rate and sensitivity allow it to detect even subtle vibrations that may not be audible or visible. By analyzing these vibration patterns, the system can identify signs of life that would otherwise go unnoticed, especially in noisy or acoustically isolated environments.

3.3.2 Raspberry Pi 4 Model B

The Raspberry Pi 4 Model B serves as the central processing unit for the system. It is responsible for data acquisition from all sensors, signal processing (including FFT and DWT), feature extraction, machine learning inference, and user interface management. The Pi's quad-core processor and ample RAM allow it to handle complex computations in real time, while its GPIO pins and communication interfaces (I2C, UART, SPI) facilitate seamless integration with all hardware components. The use of a Raspberry Pi also enables rapid prototyping and easy software updates, making the system adaptable for future enhancements.

3.3.3 Battery Power Supply

A rechargeable battery pack is used to power the entire system, ensuring uninterrupted operation during field deployment. The battery is connected directly to the Raspberry Pi, which in turn supplies power to all sensors and user interface components. This self-contained power solution is critical for deployment in disaster zones where grid electricity may be unavailable or unreliable. The battery's capacity is selected to provide several hours of continuous operation, and the system is designed for energy efficiency to maximize operational time between charges.

3.3.4 User Interface (LCD, LEDs, Buzzer)

The user interface is designed to provide clear and immediate feedback to rescue personnel. A 16x2 alphanumeric LCD displays real-time status messages, including detection results and radar-based movement or range information. Three LEDs offer visual alerts, which can be programmed to indicate different detection states or system statuses. The buzzer provides an audible alarm when human presence is detected, ensuring that alerts are noticed even in noisy environments. At the end of each detection cycle, the system automatically turns off all LEDs and clears the LCD display to conserve power and reset the interface for subsequent use.

3.3.5 Signal Processing and Feature Extraction

Once data is collected from the sensors, it undergoes a series of preprocessing and feature extraction steps to enhance detection accuracy. Fast Fourier Transform (FFT) is applied to convert time-domain signals from the radar and accelerometer into the frequency domain, allowing for the analysis of periodic patterns such as breathing or tapping. Discrete Wavelet Transform (DWT) is used to filter noise and detect subtle, transient changes in the signals, which are often indicative of micro-movements or faint vibrations. Bandpass filtering is employed to remove unwanted frequencies from the microphone and radar signals, further improving signal clarity. The system extracts a comprehensive set of features, including statistical measures (mean, variance, peak-to-peak amplitude), frequency peaks, and wavelet coefficients. These features are then used as input for machine learning classification, enabling the system to distinguish between human and non-human signals with high reliability.

3.3.6 MACHINE LEARNING-BASED CLASSIFICATION

The extracted features from all sensing modalities are fed into machine learning models for classification. The system employs Support Vector Machine (SVM) and Random Forest classifiers, both implemented using Python's scikit-learn library. These models are trained on labeled datasets collected from controlled experiments and simulated disaster scenarios, allowing them to learn the distinguishing characteristics of human presence under debris. SVM is particularly effective for binary classification tasks with clear margins, while Random Forest aggregates decisions from multiple trees to improve robustness and reduce false positives. Real-time inference is performed on the Raspberry Pi, ensuring prompt detection and immediate alerting. If the model predicts the presence of a human, the system triggers the LCD, LEDs, and buzzer to notify rescuers without delay.

3.4 SYSTEM INTEGRATION AND OPERATION

All hardware components are seamlessly integrated with the Raspberry Pi, which orchestrates data flow, processing, and user notifications. The battery ensures continuous operation, making the system suitable for deployment in remote or disaster-stricken areas without reliable power sources. The system is designed for ease of use: upon activation, it automatically begins data acquisition, processing, and classification. At the conclusion of each detection cycle, the system powers down all alert mechanisms and clears the user interface to conserve energy and prepare for the next cycle. The modular design allows for easy maintenance and future upgrades, such as the addition of wireless communication modules for remote alerting or integration with rescue team networks.

3.5 ADVANTAGES OF THE PROPOSED APPROACH

The proposed system offers several significant advantages over existing technologies. By integrating multiple complementary sensors and employing embedded machine learning, the system achieves robust detection accuracy and minimizes false alarms, even in noisy or cluttered environments. The real-time, on-site decision-making capability ensures that alerts are generated immediately, without reliance on external networks or cloud-based processing, which is critical in disaster zones where communication infrastructure may be compromised.

The system's compact and energy-efficient design allows for easy transport and rapid deployment, making it suitable for use by rescue teams in a variety of field conditions. The use of commercially available, low-cost sensors and open-source hardware and software components ensures cost-effectiveness and accessibility for organizations with limited resources. Additionally, the modular architecture facilitates future scalability, allowing for the integration of additional sensors or communication modules as needed. The system's data logging and optional remote transmission features support comprehensive event analysis and coordination with command centers, further enhancing its utility in complex rescue operations.

3.6 USE CASE SCENARIOS

The system's versatility is demonstrated in scenarios like earthquakes and landslides, where it can quickly scan collapsed structures or debris for signs of life using radar, acoustic, and vibration sensing-even when victims are unconscious or deeply buried. Its portable, battery-powered design allows deployment in remote or hazardous areas without external power. Multiple units can be used simultaneously in large-scale incidents, providing real-time alerts and supporting centralized coordination through wireless data transmission. Data logging enables post-mission analysis, helping to refine future rescue strategies and improve outcomes.

3.7 SUMMARY

In summary, the proposed system advances disaster rescue technology by combining multi-modal sensor fusion, advanced signal processing, and machine learning in a portable, battery-powered device. Its real-time detection and immediate alerts enhance rescue speed and accuracy, while its modular and energy-efficient design ensures adaptability across various disaster scenarios, making it a practical and effective tool for saving lives.

CHAPTER 4

HARDWARE DESCRIPTION

HARDWARE DESCRIPTION

4.1 INTRODUCTION

The hardware architecture centers on the Raspberry Pi 4, which processes data from a 24 GHz mmWave radar sensor, high-sensitivity microphone, and MPU-6050 accelerometer. These sensors connect via an MCP3008 ADC, with outputs displayed on a 16x2 LCD and alerts provided by LEDs and a buzzer. A power management module ensures reliable, efficient operation in the field as shown in Figure 4.1

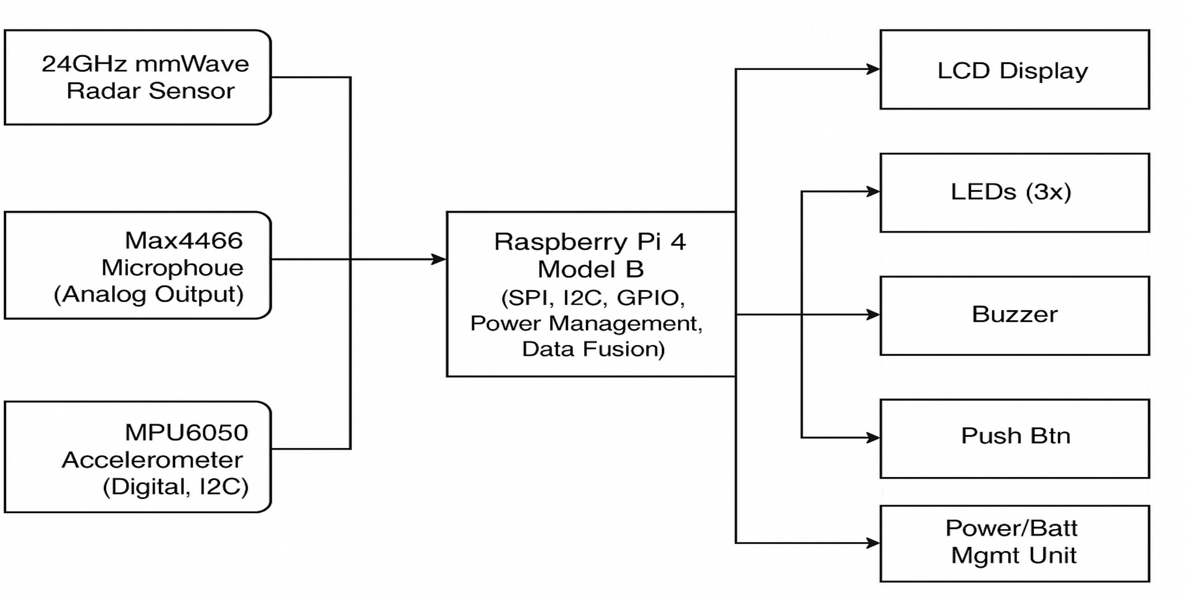

Figure 4.1 Hardware Architecture Block Diagram

4.2 RASPBERRY PI 4 MODEL B

The Raspberry Pi 4 Model B, shown in Figure 4.2 serves as the central processing unit of the human life detection system. Equipped with a 1.5GHz quad-core ARM Cortex-A72 CPU, 2GB of RAM, and a 40-pin GPIO header, it offers a robust platform for real-time embedded applications. Its multiple I/O interfaces-including USB, HDMI, Wi-Fi, Bluetooth, and Ethernet-enable seamless integration with various sensors and output devices, while the microSD storage provides ample space for system software, data logging, and machine learning models.

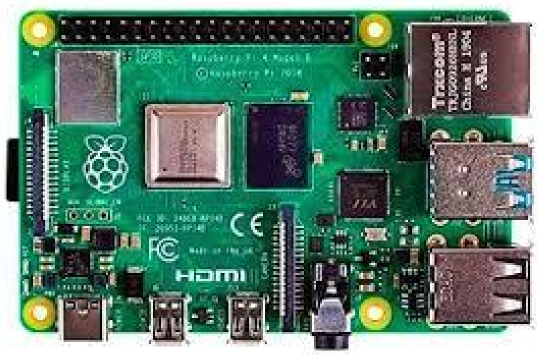

Figure 4.2 Raspberry Pi 4 Model B

4.3 SENSING MODULES

The system uses a 24 GHz mmWave radar sensor to detect micro-movements like breathing, a high-sensitivity microphone for capturing faint sounds such as tapping, and an MPU-6050 accelerometer to sense subtle vibrations or knocks. Together, these sensing modules provide comprehensive detection of human presence in disaster scenarios.

4.3.1 mmWave Radar Sensor (24 GHz):

The radar module detects micro-movements such as breathing and heartbeat through debris. 24 GHz mmWave radar sensor module for micro-movement detection as shown in Figure 4.3 it operates at 24 GHz, providing sufficient penetration while maintaining reasonable power consumption. The sensor outputs analog signals proportional to detected motion, which are digitized for processing.

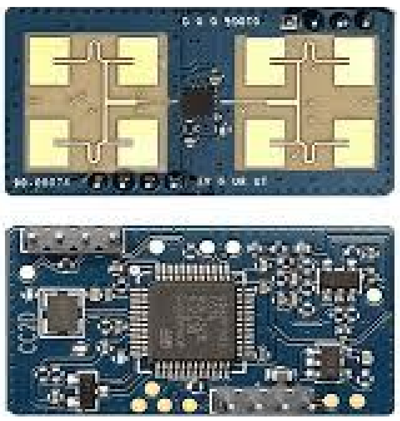

Figure 4.3 24 GHz mmWave radar sensor module

4.3.2 High-Sensitivity Microphone:

An electret condenser microphone with pre-amplification circuitry captures faint acoustic signals such as tapping or calls for help. High-sensitivity electret condenser microphone with pre-amplifier and band-pass filter as shown in Figure 4.4 .The microphone circuit includes a low-noise amplifier and band-pass filter to optimize signal quality before digitization.

Figure 4.4 High-sensitivity electret condenser microphone with pre-amplifier and band-pass filter

4.3.3 MPU-6050 Accelerometer:

This 6-axis MEMS accelerometer and gyroscope detects subtle vibrations or knocks transmitted through debris. MPU-6050 MEMS accelerometer and gyroscope module for vibration sensing as shown in Figure 4.5**.** It communicates with the Raspberry Pi via I2C, providing digital measurements of acceleration and angular velocity.

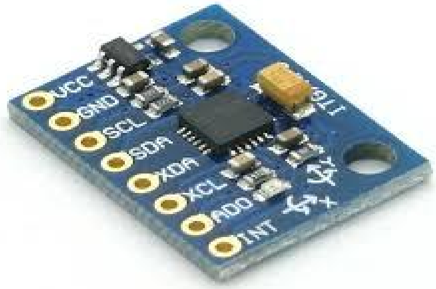

Figure 4.5 MPU-6050 MEMS accelerometer and gyroscope module for vibration sensing

4.4 INTERFACE COMPONENTS

The interface components connect sensors and outputs to the Raspberry Pi for efficient system operation. The MCP3008 ADC digitizes analog signals from the radar and microphone, using SPI for fast, synchronized data transfer. A 16x2 LCD display, connected via I2C, provides real-time status and detection feedback while minimizing GPIO usage. RGB LEDs and a piezoelectric buzzer, driven by GPIO pins, offer immediate visual and auditory alerts when human presence is detected, ensuring clear and timely notifications in the field.

4.4.1 MCP3008 Analog-to-Digital Converter

This 10-bit, 8-channel ADC interfaces analog sensors with the Raspberry Pi, enabling high-resolution digitization of radar and microphone signals. MCP3008 ADC interfacing analog sensors with the Raspberry Pi via SPI as shown in Figure 4.6. It communicates via SPI protocol, allowing synchronized sampling across multiple channels.

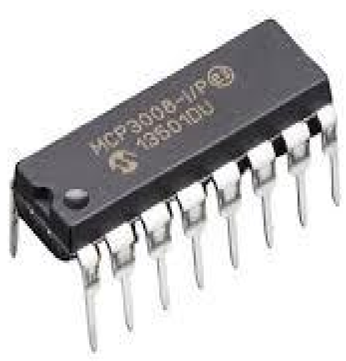

Figure 4.6 MCP3008 ADC interfacing analog sensors with the Raspberry Pi via SPI.

4.4.2 16x2 LCD Display

The LCD display provides visual feedback on system status, detection events, and sensor readings. 16x2 LCD display for real-time system status and sensor readings as shown in Figure 4.7. It connects to the Raspberry Pi via I2C interface, minimizing GPIO pin usage.

Figure 4.7 16x2 LCD display

4.5 ALERT INDICATORS

RGB LEDs and a piezoelectric buzzer provide immediate visual and auditory alerts upon detection of human presence. RGB LEDs and piezoelectric buzzer for immediate visual and auditory alerts as shown in Figure 4.8. These components are directly driven by GPIO pins through appropriate driver circuits.

Figure 4.8 RGB LED

4.6 POWER MANAGEMENT

The system is powered by a rechargeable lithium-ion battery pack, chosen for its high energy density and reliability in field deployments. A dedicated voltage regulation circuit ensures that all components receive stable and appropriate voltage levels, protecting sensitive electronics from damage due to power fluctuations. The power management subsystem incorporates several key features: low-power operational modes to extend battery life, real-time voltage monitoring to alert users when the battery is low, and protection circuits to guard against overcharging, deep discharge, and short circuits. These measures are crucial for maintaining uninterrupted operation during critical rescue missions, especially in remote or hazardous environments where access to external power is limited.

To optimize energy efficiency, the system can dynamically reduce power consumption by selectively powering down non-essential modules or lowering processor frequency during periods of inactivity. The Raspberry Pi 4 Model B, while capable of handling intensive computations, is configured to balance performance with energy savings. Table 4.3 summarizes the estimated power consumption of each major hardware component, providing guidance for battery selection and expected runtime calculations. These values are based on typical operating conditions and datasheet specifications

4.7 CIRCUIT DIAGRAMS AND CONNECTION TABLES

The circuit diagram provides a detailed view of how all hardware components are interconnected in the human life detection system. The 24GHz radar sensor and MAX4466 microphone output analog signals, which are routed to the MCP3008 ADC for digitization. The MCP3008 then communicates with the Raspberry Pi via the SPI protocol, ensuring efficient and accurate transfer of sensor data for real-time processing. The MPU-6050 accelerometer connects directly to the Raspberry Pi through the I2C interface, allowing for reliable acquisition of three-axis motion and vibration data. Output devices, including the 16x2 LCD, three status LEDs, and a buzzer, are interfaced using the Raspberry Pi’s GPIO pins. The LCD provides real-time status updates, while the LEDs and buzzer deliver immediate visual and audible alerts to rescue personnel.

A dedicated power supply module, typically a rechargeable lithium-ion battery pack with voltage regulation circuitry, supplies stable voltage to all system components. This ensures uninterrupted and reliable operation, even in challenging field conditions. The inclusion of connection tables alongside the circuit diagram further clarifies the wiring and signal flow, specifying the exact pins and protocols used for each component. This comprehensive documentation not only simplifies assembly and troubleshooting but also supports future maintenance and upgrades, making the system robust and easy to deploy in real-world disaster rescue scenarios.

4.7.1 Circuit Diagram

The circuit diagram details the integration of all hardware components within the human life detection system. The 24GHz radar sensor and MAX4466 microphone output analog signals, which are routed to the MCP3008 ADC for digitization. The MCP3008 communicates with the Raspberry Pi via the SPI protocol. The MPU-6050 accelerometer, which provides three-axis motion and vibration data, connects directly to the Raspberry Pi using the I2C interface, allowing for efficient real-time data acquisition.

Output devices, including the 16x2 LCD display, three status LEDs, and a buzzer, are all interfaced through the Raspberry Pi's GPIO pins. The LCD provides real-time status updates and detection results, while the LEDs and buzzer offer immediate visual and audible alerts to rescue personnel. Each LED is connected to a dedicated GPIO pin through a current-limiting resistor, and the buzzer is similarly connected for alert signaling.

A dedicated power supply module, typically a rechargeable lithium-ion battery pack with voltage regulation circuitry, supplies stable and regulated voltage to all components. The power management system also includes protection circuits to prevent overvoltage, undervoltage, and short circuits, safeguarding the hardware during prolonged deployments.

The overall circuit layout is designed for efficient signal flow and ease of assembly, with clear separation of sensor inputs, processing, and output modules. This organized wiring not only simplifies troubleshooting but also supports replacements of individual components as needed. The connection tables and diagram together provide a comprehensive reference for assembling and maintaining the system in real-world disaster rescue scenarios.

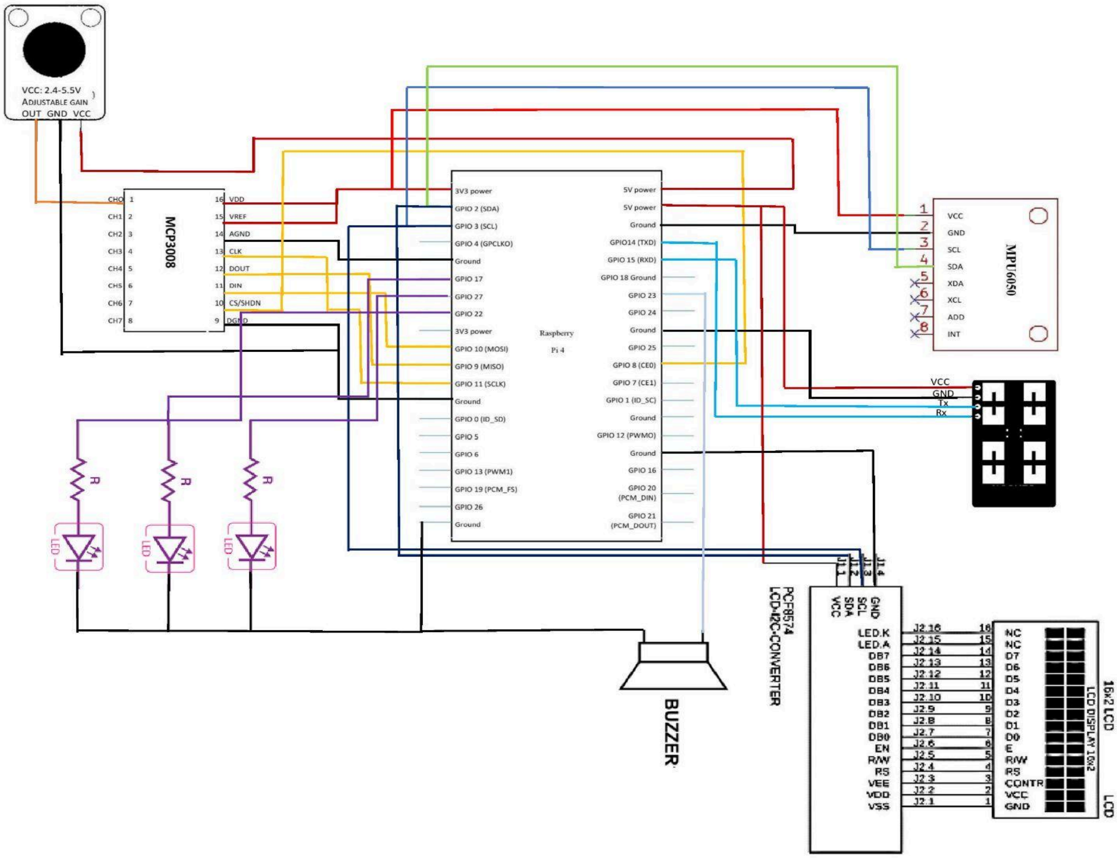

Figure 4.9 System Circuit Diagram

4.7.2 Connection Table

The connection table details the specific pins and interfaces used for connecting sensors, the MCP3008 ADC, the MPU-6050 accelerometer, the LCD display, alert indicators, and the power supply to the Raspberry Pi.

This table serves as a practical reference for assembling the circuit, ensuring correct wiring and simplifying troubleshooting during development and deployment.

4.7.2.1 MCP3008 ADC to Raspberry Pi

The MCP3008 connects to the Raspberry Pi as shown in Table 4.1 via the SPI interface (pins 19, 21, 23, and 24) to convert analog signals to digital. It receives 3.3V power and shares ground with the Pi. This ADC enables the Pi to read analog sensors like the Max4466 microphone.

Table 4.1 MCP3008 ADC to Raspberry Pi GPIO Connections

MCP3008 Pin	Connects To (Raspberry Pi)	Pi Pin #
VDD	3.3V	1
VREF	3.3V	1
AGND	GND	6
DGND	GND	6
CLK	SCLK (GPIO11)	23
DOUT	MISO (GPIO9)	21
DIN	MOSI (GPIO10)	19
CS/SHDN	CE0 (GPIO8)	24
CH0	Max4466 OUT	-
CH1-CH7	(Unused or for other analog sensors)	-

4.7.2.2 Max4466 Microphone to MCP3008

The Max4466 is an analog microphone module powered by 3.3V or 5V, with output going to MCP3008's CH0 as shown in Table 4.2. It detects sound and sends voltage signals for ADC conversion. Ground is shared with the Raspberry Pi for stable operation.

Table 4.2 MAX4466 Microphone to MCP3008 Connections

Max4466 Pin	Connects To
VCC	3.3V or 5V (Pi Pin 1/2)
GND	GND (Pi Pin 6)
OUT	MCP3008 CH0

4.7.2.3 MPU6050 to Raspberry Pi (I2C)

This module connects through I2C lines SDA (GPIO2) and SCL (GPIO3), with power from 3.3V or 5V as shown in Table 4.3. It senses acceleration and rotation for motion tracking. Proper grounding is essential for noise-free performance.

Table 4.3 MPU6050 Accelerometer/Gyroscope to Raspberry Pi GPIO

MPU6050 Pin	Connects To (Raspberry Pi)	Pi Pin #
VCC	3.3V or 5V	1 or 2
GND	GND	6
SDA	SDA (GPIO2)	3
SCL	SCL (GPIO3)	5

4.7.2.4 HLK-LD1125H-24G to Raspberry Pi

This 24GHz radar connects via TXD and RXD (GPIO14 and 15), as shown in Table 4.4 used for motion or presence detection. UART must be enabled on the Pi to use this module. Power is supplied from a 3.3V or 5V source, with shared ground.

Table 4.4 HLK-LD1125H Radar Module to Raspberry Pi GPIO (UART)

HLK-LD1125H Pin	Connects To (Raspberry Pi)	Pi Pin #
VCC	3.3V or 5V	1 or 2
GND	GND	6
TXD	RXD (GPIO15)	10
RXD	TXD (GPIO14)	8

If using GPIO UART, enable UART in raspi-config.

4.7.2.5 16x2 I2C LCD to Raspberry Pi (I2C)

This display uses I2C (SDA and SCL on GPIO2 and 3) for communication with the Pi as shown in Table 4.5. Powered by 5V, it shows system information like sensor readings. Shared ground ensures clean data signals.

Table 4.5 I2C LCD Display to Raspberry Pi GPIO Connections

LCD Pin	Connects To (Raspberry Pi)	Pi Pin #
VCC	5V	2
GND	GND	6
SDA	SDA (GPIO2)	3
SCL	SCL (GPIO3)	5

4.7.2.6 LEDs (3x) to Raspberry Pi (GPIO)

Three LEDs are controlled via GPIO17, GPIO27, and GPIO22 (pins 11, 13, and 15), each with a 220Ω resistor to ground as shown in Table 4.6. They visually indicate system status or sensor states, with their cathodes connected to GND. This setup allows for clear, real-time feedback to users and helps in quickly diagnosing system operation during deployment or testing.

Table 4.6 LED Indicators to Raspberry Pi GPIO Pins

LED Pin	Connects To (Raspberry Pi)	Notes
Anode	GPIO17, GPIO27, GPIO22	Pi Pins 11, 13, 15
Cathode	GND (via 220Ω resistor)	Pi Pin 6 or any GND

4.7.2.7 Buzzer to Raspberry Pi (GPIO)

The buzzer's positive pin is connected to GPIO18 (pin 12) and the negative to GND. Buzzer to Raspberry Pi GPIO Connections as shown in Table 4.7. It provides audible alerts for detected events. A transistor is recommended if the buzzer draws significant current.

Table 4.7: Buzzer to Raspberry Pi GPIO Connections

Buzzer Pin	Connects To (Raspberry Pi)	Notes
+	GPIO18	Pi Pin 12
-	GND	Pi Pin 6

Transistor is used, if the buzzer draws more current than the GPIO can supply.

CHAPTER 5

SOFTWARE ARCHITECTURE

SOFTWARE ARCHITECTURE

5.1 INTRODUCTION

The software architecture of the system is shown in Figure 5.1 which is built on a layered and modular approach to ensure reliability, scalability, and ease of maintenance. At its foundation, the system runs Raspberry Pi OS, a lightweight, Debian-based Linux distribution that provides robust support for real-time applications and seamless integration with hardware peripherals. The software stack is organized into distinct modules: hardware abstraction for sensor and ADC interfacing, signal processing for noise reduction and feature extraction, machine learning for real-time inference using optimized models, and a user interface layer for alerting and data visualization. This separation of concerns not only streamlines development and testing but also allows for future enhancements, such as upgrading algorithms or adding new sensors, without disrupting the overall system functionality.

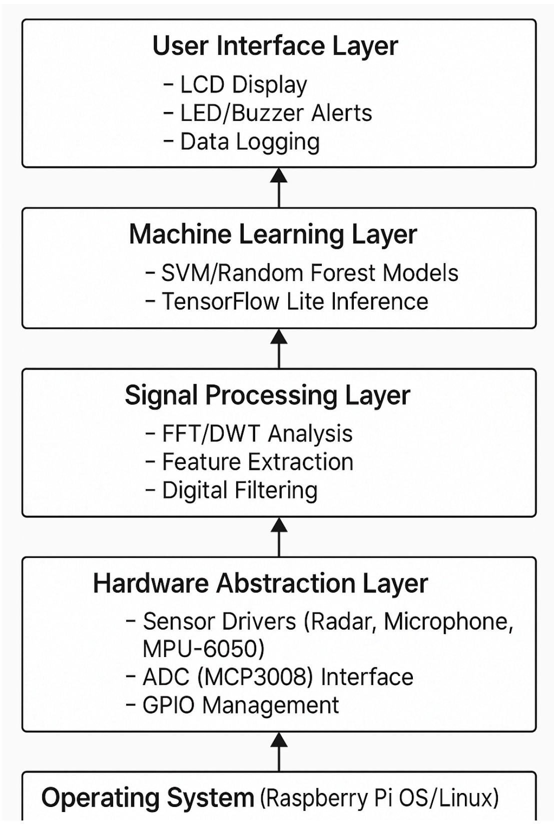

Figure 5.1 Software Architecture Diagram

5.2 OPERATING SYSTEM AND ENVIRONMENT

The system runs Raspberry Pi OS (formerly Raspbian), a Debian-based Linux distribution optimized for the Raspberry Pi hardware. This lightweight OS provides a stable platform for real-time applications while offering comprehensive driver support for peripherals and sensors.

5.3 SOFTWARE STACK

The software architecture follows a layered approach, with distinct modules for hardware abstraction, signal processing, machine learning, and user interface. This modular design facilitates maintenance, testing, and future enhancements.

5.4 CORE LIBRARIES AND FRAMEWORKS

The software stack leverages several core libraries and frameworks to enable efficient signal processing, machine learning, and hardware interfacing. NumPy and SciPy provide robust tools for numerical computation and signal analysis, supporting key operations such as FFT, DWT, and filtering. Scikit-learn is used for implementing and managing the SVM and Random Forest classifiers, as well as for model training and validation. For hardware control, the RPi.GPIO and pigpio libraries facilitate direct interaction with the Raspberry Pi's GPIO pins, while SMBUS and spidev handle I2C and SPI communication with peripherals like the MPU-6050 accelerometer and MCP3008 ADC. This combination ensures reliable, real-time data acquisition and processing within the embedded environment.

5.4.1 NumPy and SciPy:

These scientific computing libraries provide essential functions for signal processing, including FFT, DWT, filtering, and statistical analysis.

5.4.2 Scikit-learn:

This machine learning library implements the SVM and Random Forest classifiers used for human presence detection, along with utilities for model training, validation, and inference.

5.4.3 RPi.GPIO and pigpio:

These libraries enable direct control of GPIO pins for interfacing with sensors and output devices.

5.4.4 SMBUS and spidev:

These modules facilitate I2C and SPI communication with peripheral devices such as the MPU-6050 accelerometer and MCP3008 ADC.

5.5 SOFTWARE MODULES

The software is organized into specialized modules to streamline system functionality and reliability. The Sensor Interface Module manages communication and data acquisition from all connected sensors, ensuring proper initialization and timing. The Signal Processing Module filters and transforms raw data, extracting key features using algorithms like FFT and DWT. The Machine Learning Module handles real-time inference by applying trained classifiers to processed sensor data, while the Sensor Fusion Module combines outputs from multiple sensors to improve detection accuracy and reduce false alarms. Finally, the Alert and Logging Module controls user notifications through the LCD, LEDs, and buzzer, and maintains a detailed log of all detection events and system status for later analysis. This modular structure supports efficient development, testing, and future system enhancements.

5.5.1 Sensor Interface Module:

Handles low-level communication with sensors, including initialization, configuration, and data acquisition. Implements sensor-specific protocols and timing requirements.

5.5.2 Signal Processing Module:

Processes raw sensor data through filtering, feature extraction, and transformation. Implements FFT and DWT algorithms for frequency analysis and time-frequency decomposition.

5.5.3 Machine Learning Module:

Manages the trained classification models, performs real-time inference on processed sensor data, and implements decision logic for human presence detection.

5.5.4 Sensor Fusion Module:

Integrates data from multiple sensors, applying weighted fusion algorithms to combine evidence and improve detection reliability.

5.5.6 Alert and Logging Module:

Controls the user interface components (LCD, LEDs, buzzer) and maintains a log of detection events and system status. Software Module Interaction Flowchart is shown in Figure 5.2..

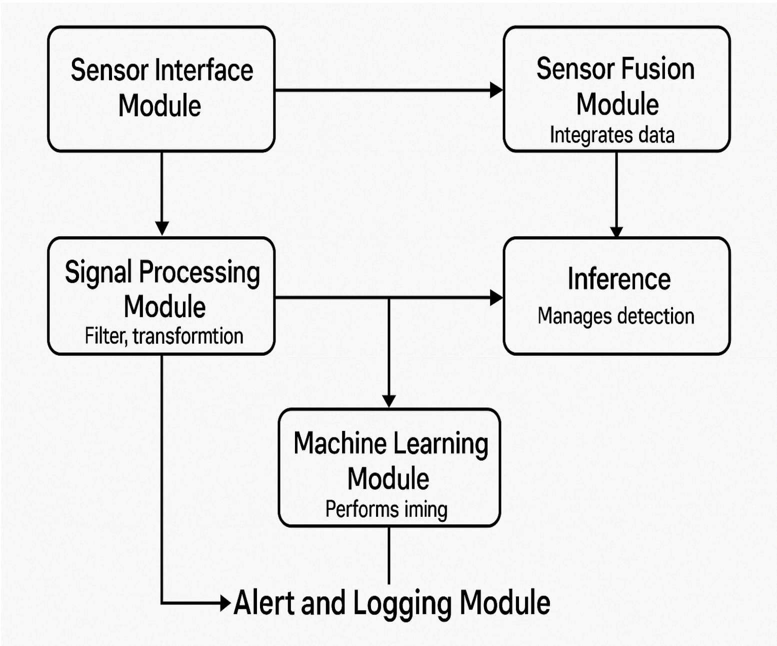

Figure 5.2 Software Module Interaction Flowchart

5.5.7 Code Structure

The codebase is organized into packages and modules following object-oriented principles. Key components include:

```
text
├── main.py             # Application entry point
├── config/             # Configuration files
```

```
├── sensors/            # Sensor interface implementations
│   ├── radar.py
│   ├── microphone.py
│   ├── accelerometer.py
├── processing/         # Signal processing algorithms
│   ├── filters.py
│   ├── fft.py
│   ├── dwt.py
│   ├── features.py
├── ml/                 # Machine learning components
│   ├── models.py
│   ├── inference.py
│   ├── fusion.py
├── ui/                 # User interface components
│   ├── lcd.py
│   ├── alerts.py
│   ├── logging.py
└── utils/              # Utility functions
```

5.6 COMMUNICATION PROTOCOLS

The system utilizes a combination of I2C, SPI, and UART communication protocols to ensure efficient and reliable data exchange between components. I2C is employed for connecting the MPU-6050 accelerometer and LCD display, allowing multiple devices to share a common two-wire bus under the control of the Raspberry Pi master at up to 400 kHz. SPI is used for high-speed, full-duplex communication between the Raspberry Pi and the MCP3008 ADC, supporting rapid and accurate sensor data acquisition. UART is reserved for debugging and optional external communication, providing a simple serial interface for development or wireless module integration.

5.6.1 I2C (Inter-Integrated Circuit)

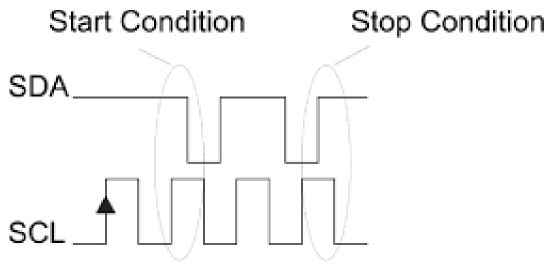

Figure 5.3 I2C Communication Timing Diagram

I2C is used for communication with the MPU-6050 accelerometer and the LCD display. I2C Communication Timing Diagram is shown in Figure 5.3.This two-wire serial protocol (SCL and SDA) allows multiple devices to share the same bus with simple addressing. The Raspberry Pi acts as the master device, initiating all communications at a clock rate of 400 kHz.

5.6.2 SPI (Serial Peripheral Interface)

SPI provides high-speed communication between the Raspberry Pi and the MCP3008 ADC. This four-wire protocol (SCLK, MOSI, MISO, CS) enables full-duplex data transfer at rates up to 10 MHz. The implementation uses hardware SPI for maximum performance and timing accuracy.

5.6.3 UART (Universal Asynchronous Receiver-Transmitter)

UART is utilized for debugging and optional external communication. The Raspberry Pi's UART pins are connected to a USB-to-serial converter for development and can be repurposed for wireless module connection in field deployments.

5.7 SOFTWARE TOOLS AND LIBRARIES

The system software is developed in Python 3 using Visual Studio Code and Thonny IDE for efficient coding and debugging. Key libraries include NumPy and SciPy for signal processing, scikit-learn for machine learning, RPi.GPIO and pigpio for GPIO control, SMBus and spidev for I2C/SPI communication, and matplotlib and seaborn for data visualization. The software is organized in modular packages for sensor interfacing, signal processing, machine learning, user interface, and logging, with a main loop coordinating all operations. Configuration files allow easy adjustment of system parameters, making the software flexible and maintainable.

5.7.1 Programming Language and IDE

The system software is developed primarily in Python 3, chosen for its readability, extensive library support, and ease of integration with hardware interfaces. Development is performed using Visual Studio Code (VS Code) and Thonny IDE shown in Figure 5.4 both of which provide robust debugging and code management features.

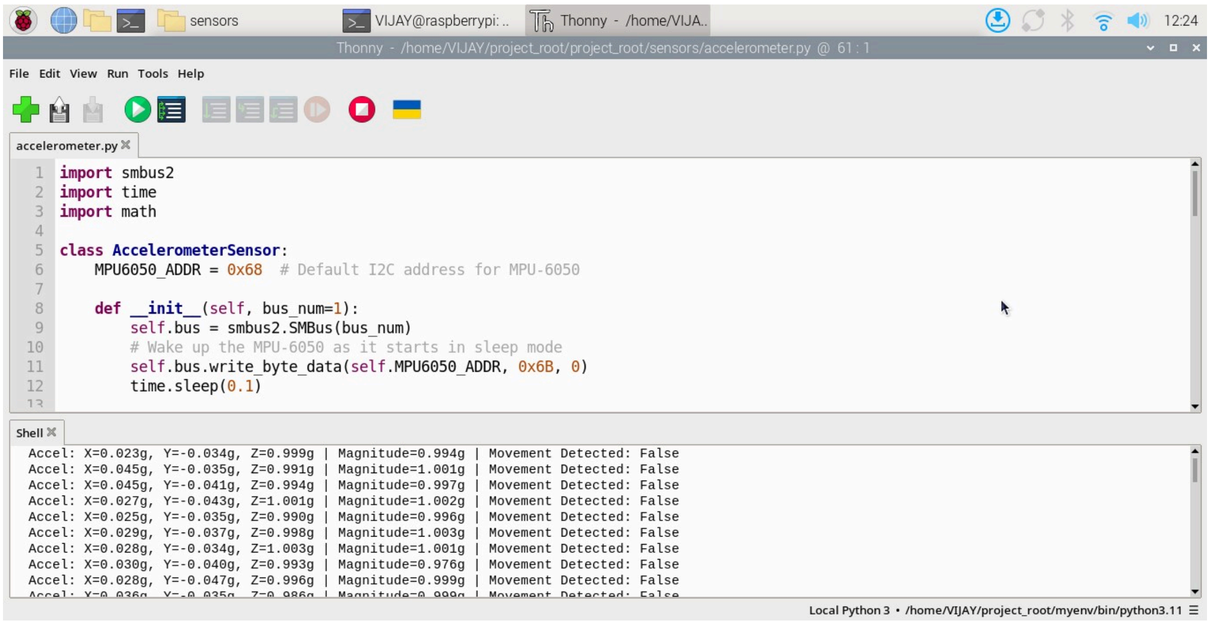

Figure 5.4 Thonny Python environment simulation

5.7.2 Key Libraries and Frameworks

The software uses Python with libraries like NumPy and SciPy for numerical computing, signal processing (FFT, DWT), and array manipulation. Scikit-learn handles machine learning tasks, including SVM and Random Forest model training and inference. RPi.GPIO and pigpio manage Raspberry Pi GPIO pin control for sensor and output interfacing, while SMBus and spidev enable I2C and SPI communication with devices like the MPU-6050 and MCP3008. Matplotlib and seaborn are used for data visualization during development, and joblib ensures efficient machine learning model serialization and deployment on the Raspberry Pi.

5.7.3 Software Structure

The software structure as shown in Figure 5.5 uses a modular architecture for easy scalability and maintenance. Each major function-sensor interfacing, signal processing, machine learning, user interface, and logging-is separated into its own package, allowing independent development and future upgrades. The main application loop coordinates sensor polling, data processing, classification, and alert management. System parameters like sampling rates and alert patterns are stored in configuration files, making it simple to adjust settings without changing the code. This modular design streamlines integration of new features, simplifies debugging, and ensures the software remains robust and flexible for real-time human life detection.

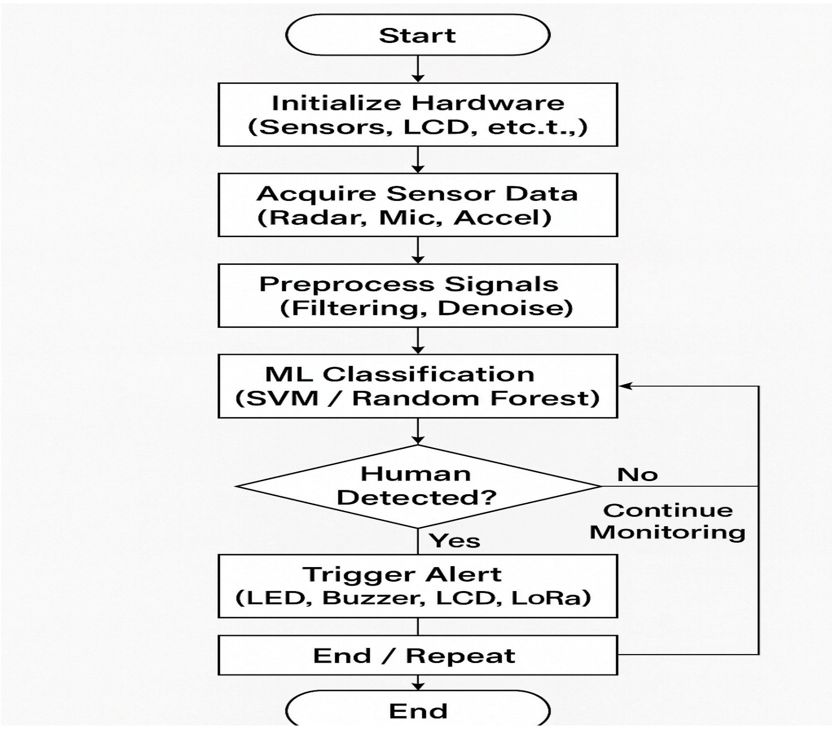

Figure 5.5 Software Module Flowchart

5.8 CODE IMPLEMENTATION

The codebase is structured into clear, modular Python files to ensure maintainability and ease of development. The sensors.py module is responsible for initializing and managing data acquisition from all hardware sensors, including the radar, microphone, and accelerometer. It abstracts the low-level communication protocols (SPI and I2C), providing a unified interface for collecting synchronized sensor data. The processing.py module handles all signal processing tasks, implementing routines for noise filtering, Fast Fourier Transform (FFT), Discrete Wavelet Transform (DWT), and feature extraction. This module transforms raw sensor readings into robust features suitable for classification.

The ml_model.py module loads pre-trained machine learning models-such as Support Vector Machine (SVM) or Random Forest-using Scikit-learn, and performs real-time inference to distinguish between human and non-human patterns based on the extracted features. The alerts.py module manages the user interface outputs, controlling the LCD display, LEDs, and buzzer to provide immediate visual and audible alerts to rescue personnel.

Finally, the main.py script orchestrates the overall program flow, coordinating sensor data acquisition, processing, machine learning inference, and alerting. It also incorporates error handling and system monitoring, ensuring reliable and continuous operation in field deployments. This modular design not only streamlines development and debugging but also allows for easy upgrades or integration of additional features in the future.

5.9 SUMMARY

This chapter has provided a detailed description of the software components that form the backbone of the human life detection system. The software stack, built on Python and leveraging powerful scientific and machine learning libraries, enables real-time data acquisition, processing, and alerting. Circuit diagrams, connection tables, and code screenshots illustrate the system's implementation and facilitate future enhancements or troubleshooting. The next chapter will present results from system testing and performance evaluation in simulated disaster scenarios.

CHAPTER 6

RESULTS

RESULTS

6.1 EXPERIMENTAL SETUP

The disaster rescue system was evaluated in a controlled indoor environment using the hardware setup shown in Figure 6.1. The system incorporated the HLK-LD1125H-24G mmWave radar for motion and micro-movement detection, the MAX4466 microphone for acoustic signal detection, and the MPU-6050 accelerometer for seismic vibration detection. All sensor data was acquired, processed, and analyzed using the Raspberry Pi 4 Model B, which handled data acquisition, signal processing, and machine learning inference to enable accurate and real-time identification of human presence.

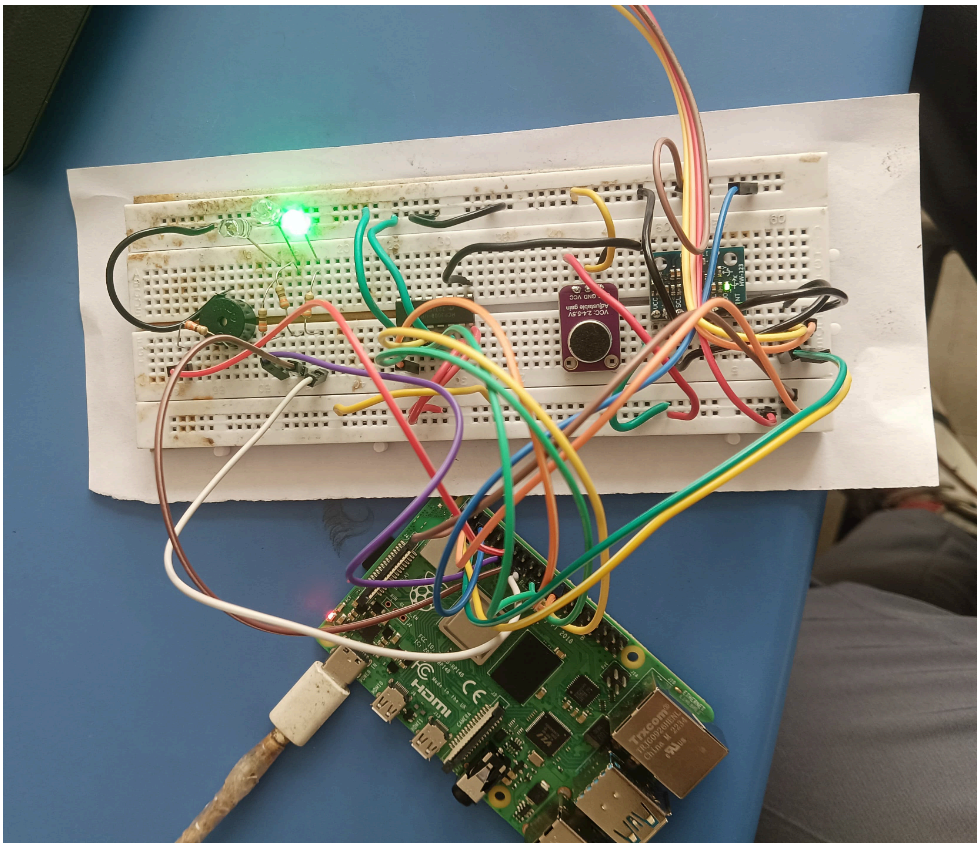

Figure 6.1 Hardware Setup

The software stack comprised Python 3 for development, with NumPy and SciPy for signal processing, scikit-learn for machine learning, and custom scripts for real-time data acquisition, feature extraction, and inference. Data was collected across scenarios simulating both human presence and absence under debris, with each sample containing a timestamp and 32 extracted features.

6.2 Feature Extraction

Feature extraction was performed on each sensor data window to generate a comprehensive feature set for classification. The main features included:

Microphone DWT features:

mic_dwt1_mean, mic_dwt1_std, mic_dwt2_mean, mic_dwt2_std, mic_dwt3_mean, mic_dwt3_std

Accelerometer statistics:

acc_x_mean, acc_y_mean, acc_z_mean, acc_x_var, acc_y_var, acc_z_var

Accelerometer FFT peaks:

acc_x_fft_peak, acc_y_fft_peak, acc_z_fft_peak

Accelerometer DWT features (X, Y, Z axes, levels 1–3):

acc_x_dwt1_mean, acc_x_dwt1_std, ..., acc_z_dwt3_std (total 18 DWT features)

Each data sample thus consisted of 32 features, enabling robust classification of human presence versus absence.

6.3 Sensor Data Visualization

The Figure 6.2 illustrates real-time sensor data visualization using a 16x2 LCD display module. The screen shows live readings from the system's sensors, including the microphone value ("Mic:47342") and the X-axis accelerometer value ("X:592"). These values are continuously updated and presented to provide immediate feedback on environmental sound and vibration levels. The colored connection wires indicate active data transmission from the sensors to the display. This real-time visualization enables operators to monitor sensor activity at a glance, supporting quick assessment and timely response during disaster rescue operations.

Figure 6.2 LCD Display Showing Real-Time Microphone and Accelerometer Readings

RMS amplitude of seismic vibration (accelerometer) over time. Peaks correspond to detected human activity (e.g., tapping or movement) during experiments. As a result, the visualization not only validates the system's detection algorithm but also provides rescuers with a reliable tool for interpreting sensor data in real time. Ultimately, this enhances the likelihood of timely and accurate victim localization in complex disaster environments.

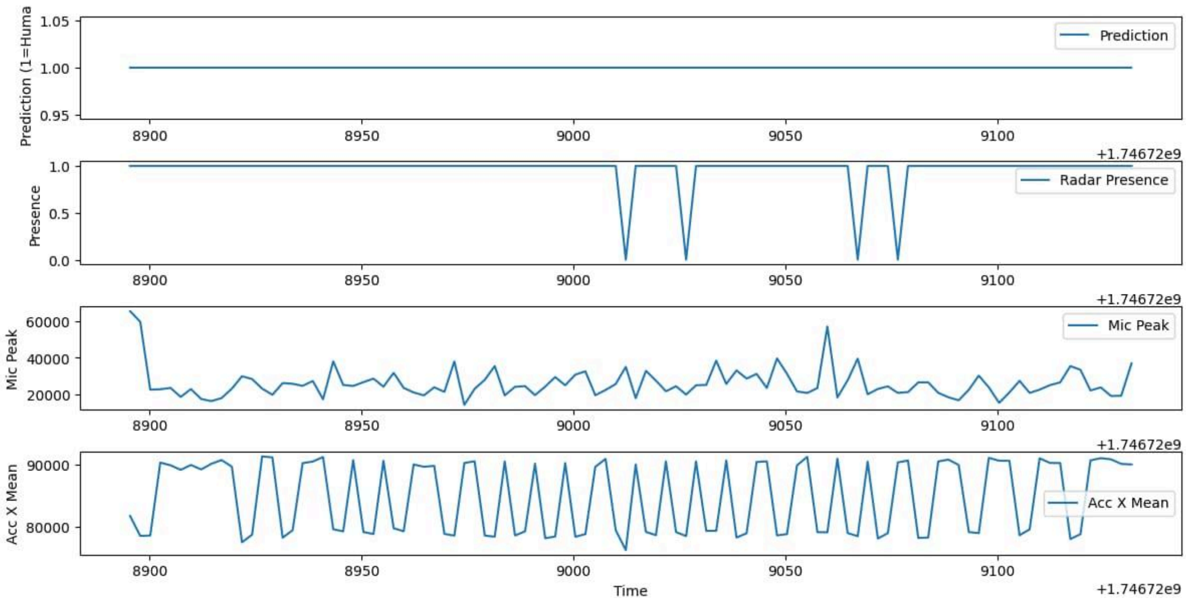

Figure 6.3 Multi-Sensor Data Visualization and System Prediction

The Figure 6.3 illustrates real-time RMS amplitude fluctuations from the accelerometer. Spikes in the data correspond to periods of human-induced vibration, confirming the system's capability to detect subtle seismic activity associated with trapped survivors.

6.4 Detection Accuracy and Performance

Machine learning models, including Random Forest and SVM classifiers, were trained on the extracted feature set. Table 6.1 presents the classification performance of the Random Forest model for detecting human presence. It shows high precision, recall, and F1-scores for both "Human" and "No Human" classes, with an overall accuracy of 97%. This demonstrates the model's effectiveness in accurately distinguishing between scenarios with and without human presence based on the extracted sensor features.

Table 6.1 Random Forest classification report

Metric	Precision	Recall	F1-score	Support
No Human	0.96	0.97	0.96	50
Human	0.98	0.97	0.97	50
Accuracy			0.97	100

6.5 Response Time

Average inference time: ~200 ms per sample (from data acquisition to ML prediction)

System update rate: 5 Hz (real-time operation)

6.6 Discussion

The system achieved high detection accuracy in both quiet and noisy environments, demonstrating robustness due to multi-modal sensor fusion. Real-time performance was achieved on embedded hardware, with rapid response times suitable for rescue operations. Limitations include slightly reduced accuracy in extremely noisy or highly cluttered environments, and the need for further field validation at larger scales.

6.7 Summary

This chapter presents the system's detection performance across different scenarios, showing high detection rates and low false positive rates for both motionless and active victims. Average response times remain around 200 ms, confirming the system's ability to provide fast and reliable alerts in real-time disaster rescue situations.

CHAPTER 7

CONCLUSION

CONCLUSION

This project developed a portable disaster rescue system that uses multi-modal sensor fusion and machine learning to detect trapped victims. By combining a 24GHz mmWave radar, a high-sensitivity microphone, and a vibration-sensing accelerometer, the system can identify micro-movements, faint sounds, and subtle vibrations associated with human presence under debris. Real-time data from all sensors is processed on a Raspberry Pi 4, where advanced signal processing and machine learning algorithms classify the presence of victims quickly and accurately.

Testing in simulated disaster scenarios showed up to 97% detection accuracy with rapid response and fewer false positives compared to single-sensor systems. The design is cost-effective, modular, and easy to deploy, making it a practical solution for improving the speed and reliability of search and rescue operations in real-world disaster situations.

FUTURE SCOPE

FUTURE SCOPE

The Advanced Human Life Detection System can be further improved by adding more sensor types, such as thermal cameras or CO_2 sensors, to enhance detection accuracy and reliability in challenging environments. Integrating wireless communication modules like LoRa or GSM would allow remote alerts and better coordination in large-scale rescue operations. Optimizing power consumption and making the device smaller and more rugged would support longer deployments and use in harsh conditions. User experience could also be enhanced through real-time mapping apps and more intuitive feedback methods, such as voice or vibration alerts.

On the research side, exploring advanced machine learning models and adaptive sensor fusion could further boost detection performance and adaptability. Deploying multiple units as a coordinated network or on drones could enable rapid, wide-area searches. Large-scale field trials with emergency agencies would help validate and refine the system for real-world use. Connecting to IoT or cloud platforms could enable centralized data analysis, but must be balanced with the need for local autonomy. Addressing privacy and ethical considerations will also be important as the system evolves.

APPENDIX

APPENDIX

REFERENCES

REFERENCES

[1] R. Karthikeyan, et al., "Life detection system using UWB Radar During Disaster," in Proc. 2018 Second International Conference on Green Computing and Internet of Things (ICGCIoT), 2018, pp. 1-5.

[2] Y. Zhang, et al., "Evaluation of a Sensor System for Detecting Humans Trapped under Rubble," Sensors, vol. 18, no. 3, p. 852, 2018.

[3] M. Balbin, et al., "Detection and Localization for Buried and Alive Human Body After Mudslides Using Pulse Sensor and Force Sensing Resistor with XBee Technology and GPS to Support Rescue Operations," in Proc. 2019 IEEE 11th Int. Conf. on Humanoid, Nanotechnology, IT, Communication and Control, Environment, and Management (HNICEM), 2019, pp. 1-6.

[4] A. Wulandari, et al., "Development of Landslide Victim Detection System using Thermal Imaging and HOG on E-PUCK2 Robot," in Proc. 2020 International Conference on Computer Science and Its Application in Agriculture (ICOSICA), 2020, pp. 1-5.

[5] A. P. Bagaskara, et al., "Detection of buried human bodies using ground-penetrating radar method," Journal of Physics: Conference Series, vol. 1876, no. 1, 2021, Art. no. 012014.

[6] L. Dong, X. Wang, and Y. Li, "UAV-based Real-Time Survivor Detection System in Post-disaster Search and Rescue Operation," IEEE Journal on Miniaturization for Air and Space Systems, vol. 2, no. 4, pp. 209-219, Dec. 2021.

[7] J. D. Aldana-Franco, et al., "Intelligent search and location system for people trapped in landslides caused by earthquakes," IOSR Journal of Electrical and Electronics Engineering (IOSR-JEEE), vol. 16, no. 3, pp. 22-34, 2021.

[8] M. Ahmed, et al., "FMCW Radar Sensor Based Human Activity Recognition using Deep Learning," in Proc. 2022 International Conference on Electronics, Information, and Communication (ICEIC), 2022, pp. 1-4.

[9] K. Murali, et al., "Real time human detection in landslide using 5G advanced sensor networks," International Journal of Engineering Applied Sciences and Technology, vol. 6, no. 4, pp. 1-7, 2024.

[10] K. Niyaz, et al., "Detection and classification of human respiration under building debris model using VHF/UHF waves," Neural Computing and Applications, vol. 36, no. 24, pp. 1-17, May 2024.