Name: Xingchen Chen
Matriculation number: S2269664
MEng Honours Project Phase 1 Report
<IoT for Health: An Integrated System
for Monitoring Physiological Parameters
and Activity Recognition>
31 July 2023

Mission Statement

Project Title: Smartphone interface for peripheral sensor

Project Definition

The mission of this project is to develop an advanced IoT system that integrates IMU sensors for accurate human activity recognition, along with blood pressure and SpO2 sensors. The primary objective is to create a comprehensive solution that provides a holistic view of an individual's physical state, enabling the early detection of potential health issues. By combining multiple sensors, this IoT system aims to deliver real-time monitoring and analysis, facilitating proactive healthcare management and timely intervention.

Preparatory Tasks

1. Collect information about sensors, Arduino, Android, Bluetooth, and machine learning concepts.
2. Install the necessary software and tools to set up the development environment.

Main Tasks

1. Develop the IoT system using the MAX30102 sensor, Arduino, and the Nordic Thingy 52, with a focus on the hardware component.
2. Collect and analyze data from the IMU sensors, blood pressure, and SpO2 sensors to obtain accurate measurements and insights.
3. Build the Android application that will serve as the interface for the IoT system. Integrate the functionalities of the sensors and ensure communication between the smartphone and external devices.
4. Test and debug the system to ensure its functionality and reliability.
5. Create comprehensive documentation and user manuals that provide clear instructions and guidelines for users to understand and effectively utilize the IoT system.

Scope for Extension

This project has potential for future expansion and enhancement, which may include:
1. Addition of supplementary sensors to gather additional health-related data.
2. Integration with other devices or platforms to enhance interoperability and data-sharing capabilities.

Background Knowledge

Successful completion of this project requires proficiency and experience in: IoT and embedded systems, sensor utilization, human activity recognition, machine learning concepts and models, data analysis methods, programming languages(C++, Java and Kotlin), bluetooth protocols and communication.

Acceptance Criteria

1. The IoT system integrates and facilitates seamless communication between the sensors, Arduino, and Android smartphones.
2. The system accurately collects data from IMU, blood pressure, and SpO2 sensors, ensuring precise measurements of physical activity and health parameters.
3. The Android application offers an intuitive and user-friendly interface.
4. Rigorous testing and debugging are executed to ensure the functionality and reliability of the IoT system under diverse scenarios and conditions.
5. The IoT system demonstrates optimized performance by efficiently processing data, ensuring prompt and accurate real-time monitoring and analysis.

Resources

Sensors, android phone, computer, etc.

Location

Edinburgh

Abstract

Addressing the escalating demand for accessible, real-time, and reliable health monitoring systems, this study introduces an integrated health evaluation tool's implementation. Harnessing Internet of Things (IoT) technologies, the system leverages a MAX30102 Pulse Oximeter and Heart-Rate sensor for non-invasive monitoring of key physiological parameters, heart rate and blood oxygen saturation (SpO2). Concurrently, it employs a Convolutional Neural Network (CNN) model for effective Human Activity Recognition (HAR), utilizing movement data gathered by 2 IMU sensors (Thingy:52) attached to the human body. The information is made readily accessible through an Android application, bridging the gap between users and their health data. The integrated tool effectively marries physiological monitoring with activity recognition, culminating in a comprehensive health assessment system. The system demonstrated an accuracy of 97% in human recognition, underscoring its reliability and efficacy. This innovative health monitoring system exemplifies the promising future of personalized healthcare in the digital era, signalling the potential for advancements in remote health monitoring and activity recognition technology.

Declaration of Originality

I declare that this thesis is my original work except where stated.

Chen Xingchen

Statement of Achievement

The primary accomplishment of this project is the successful development of an effective, reliable, and user-friendly health-tracking system. The system can measure heart rate and SpO2 level and detect user activities by employing IMU sensors and machine learning techniques. It provides users with immediate access to their physiological data and activity recognition results via an Android app; it also provides useful health advice based on the collected data.

In terms of hardware, I used the Arduino development board, integration using the MAX30102 sensor and HC-06 Bluetooth module, and the associated debugging processes. I also completed the program with the help of existing libraries in the Arduino IDE to achieve successful data acquisition, which is key for measuring heart rate and SpO2 level. Users can connect the Arduino subsystem to their Android phones via the Android application, and the relevant physiological data is presented on the app's interface when the user places their finger on the sensor.

On the software side, the study of machine learning algorithms allowed me to understand each model's distinct characteristics, application scenarios, and optimisation methods. It is vital in implementing a CNN-based ML model for classifying the activities, leading to accurate classification of human activity recognition. Incoming movement data from 2 inertial measurement units is processed through the machine learning model, with detected activities identified and displayed on the app's interface.

Moreover, I used Android Studio and gained proficiency in Java and Kotlin languages, understood various Android components, and learned to use Android Bluetooth and BLE functions. Combined with knowledge of data structures and algorithms, the Android app's frontend and backend were successfully developed. The app provides users access to their real-time physiological data, current activity, and a history feature that uses a histogram to represent the recorded activities and their duration.

Last but not least, the system can take into account both the heart rate and SpO2 readings, along with the HAR outcome, to provide personalised and relevant health advice.

This comprehensive approach towards health monitoring underlines the project's achievements in advancing personal health management.

Contents

List of Symbols

A	Unit of electric current in the International System of Units.
V	Unit of electric potential difference in the International System of Units.
Ω	Unit of electrical resistance in the International System of Units.

Glossary

CNN Convolutional neural networks.

HAR Human Activity Recognition.

IMU Inertial Measurement Unit.

IoT Internet of Things.

ML Machine learning.

RL Reinforcement learning.

Chapter 1

Introduction and Background

1.1 Introduction

The transformative potential of the Internet of Things (**IoT**) has been recognized across various sectors, and healthcare is no exception. Fueled by an ageing global population and escalating healthcare demands, the necessity for advanced, remote, and continuous health monitoring systems is profound. This project addresses these needs by developing an IoT-based system that harnesses the convenience of smartphone interfaces and peripheral sensors to facilitate real-time health monitoring.

This project merges IoT and smartphone technologies to enable continuous, real-time monitoring of critical health parameters. With a smartphone interface for easy accessibility and advanced sensors such as the MAX30102 for non-invasive tracking of heart rate and blood oxygen levels, the system promotes seamless health monitoring. Additionally, implementing Human Activity Recognition (**HAR**) through machine learning techniques and the Nordic Thingy:52 IoT Sensor Kit adds value by identifying a range of human activities, thus providing data that enhances the understanding of collected health metrics.

The structure of this dissertation is organized to provide a comprehensive understanding of the proposed system and its impact. Following this introduction, which outlines the background and significance of the problem, a literature review will offer detailed insights into the field's current state. The subsequent chapter will present the system's design and development, unfolding the intricacies of the hardware and software components. This will be followed by a chapter discussing the testing methodologies deployed, setting the stage for a detailed results and analysis chapter. The dissertation's final sections will form the conclusion and recommendations, thus providing a holistic view of the project's impact and potential future directions.

1.2 Background

The global population is undergoing a significant demographic shift, with the ageing sector projected to rise from 12% to 22% between 2015 and 2050[1]. This increase poses numerous healthcare challenges, particularly in managing chronic diseases like cardiovascular disorders. These conditions, leading to approximately 17.9 million deaths annually or 31% of all global mortality, affect the elderly and an increasingly younger demographic due to lifestyle factors[1]. This rising trend underscores the urgency for innovative, continuous, and efficient health monitoring solutions to address the mounting healthcare demands.

In response to this health crisis, healthcare professionals and technologists seek effective ways to monitor, prevent, and manage diseases across all age groups. One key strategy is the early detection of health problems, which requires monitoring key health parameters, including heart rate and blood oxygen saturation levels. This strategy forms the premise of this thesis project and guides the development of the IoT-based system described herein.

1.3 Problem Statement

The necessity for real-time, non-invasive monitoring of vital health parameters spans various health conditions, including cardiovascular and respiratory disorders, neurological conditions, and general health and wellness. A system that facilitates this continuous tracking is pivotal for proactive healthcare, paving the way for preventive measures and efficient disease management.

Existing monitoring solutions, including fingertip pulse oximeters and wearable devices, can provide data for heart rate and SpO2 levels. However, they often lack data analysis and cannot track long-term health trends. Consequently, they provide more of an instantaneous health snapshot than an ongoing, in-depth analysis crucial for identifying significant fluctuations or patterns.[2]

Most existing monitoring systems also fail to factor in the user's physical activity level, which can impact heart rate and SpO2 readings. The person's current activity (such as sleeping, exercising, or performing everyday tasks) may cause these health parameters to vary, and without taking this into account, the data may potentially lead to false alarms.

Additionally, while wearable devices like smartwatches offer continuous health monitoring, their accuracy can be compromised due to their positioning on the wrist, which limits their ability to recognize the user's posture and movement accurately. The high cost of these devices can also be a barrier for many users.[3]

Moreover, many IoT systems that measure SpO2 are based on control boards like ESP8266 and connect to a server, which requires a stable Wi-Fi connection, limiting their usefulness in areas with poor or no network connectivity. A system based on a mobile platform using Bluetooth connectivity could provide more consistent online and offline performance[4].

In summary, there is an imperative need for a cost-effective, accurate, and user-friendly monitoring system that offers continuous, real-time tracking of heart rate and SpO2 levels while factoring in the user's activity level. The system should offer real-time data analysis and integrate seamlessly into a user's daily life, promoting proactive healthcare management and timely interventions. This project aims to develop such a system.

1.4 Significance of the Study

The importance of this study lies in its potential to contribute to the health monitoring field, particularly regarding wearable devices and human activity recognition. This research aims to explore and elucidate the synergies between wearable health monitoring devices, HAR, the Internet of Things, and machine learning.

Firstly, the study introduces a unique health monitoring method, blending continuous heart rate and blood oxygen levels tracking with Human Activity Recognition. Using the MAX30102 photoelectric sensor, the system generates constant streams of essential health metrics. What elevates the system is the incorporation of HAR, using machine learning to identify users' activities like sitting, walking, or running. This offers a nuanced interpretation of heart rate and SpO2 data with activities.

Moreover, in the realm of patient care, the inclusion of IoT technology for wireless communication could significantly transform the practice. The system's ability to remotely monitor patients' vital signs can reduce the need for regular hospital visits, thereby enhancing patient care, particularly for individuals with mobility difficulties, such as elderly or injured patients.

The findings from this study could not only help direct future research but also influence the design of next-generation health monitoring devices. By identifying the gaps in current knowledge, this research can potentially stimulate innovation and advancement in healthcare technology.

1.5 Objectives

This study is guided by the following main objectives:

1. Development of the IoT System: This research aims to design and implement a comprehensive IoT system for real-time health monitoring. The system will include a user-friendly smartphone interface

that connects with peripheral sensors to gather and process data.

2. Integration of Heart Rate and SpO2 Measurement: Another goal of this study is to implement a continuous heart rate and blood oxygen level (SpO2) monitoring system. Using the MAX30102 photoelectric sensor, the system will capture and process these vital health parameters.

3. Incorporation of Human Activity Data: Realizing the Human Activity Recognition functionality is essential. This ensures the physiological data - heart rate and SpO2 levels - are interpreted in context, considering the user's current physical activity. It can also track and record users' daily activities, understanding their routine and physical exertion levels. Relative health advice can be provided based on the captured data.

4. Testing the System: Once the system is developed, the performance of the system will be rigorously tested. The testing process will involve lab-based experiments and real-world testing to ensure that the system provides reliable and accurate outcomes.

Chapter 2

Literature Review

2.1 Introduction

This literature review aims to provide a comprehensive understanding of the current state of wearable health monitoring devices, their evolution, and how they intersect with emerging technologies like the Internet of Things (IoT) and machine learning. This review explores the application of these devices in monitoring physiological parameters like SpO2 level and heart rate and the role of Human Activity Recognition in healthcare. By examining existing research and identifying gaps in the current body of knowledge, potential areas for further investigation are highlighted.

2.2 IoT and Wearable Devices in Healthcare

The amalgamation of the Internet of Things and healthcare has brought a significant transformation in the healthcare industry, particularly through the proliferation of wearable health monitoring systems [5], [6]. These devices, equipped with a variety of sensors, offer real-time monitoring of health indicators and have fundamentally shifted the healthcare delivery model towards a patient-centric approach [5]–[7].

2.2.1 The Convergence of IoT and Healthcare

The increasing integration of IoT technologies into healthcare has revolutionized health monitoring and disease prevention. Connected devices are now widely used for patient monitoring, prevention of diseases, and rehabilitation, providing unprecedented health data collection and analysis [6]. This growth is fostered by the rapid development of wearable devices that collect a variety of health-related data, facilitating the early detection of health anomalies and diseases such as hypertension, cardiovascular disease, and diabetes [8].

2.2.2 Ubiquitous and Pervasive Health Paradigms

Emerging from this fusion are new healthcare paradigms such as ubiquitous health (u-health) and pervasive health (p-health). These concepts leverage wearable technologies to offer continuous health monitoring, facilitating proactive personal health management and timely healthcare intervention [9]. Such advancements provide personalized health insights, enabling preventive healthcare and early diagnosis of potential health problems [10].

2.2.3 Evolution of Wearable Health Devices

The design and functionality of wearable devices have significantly evolved with advances in material science, microelectronics, and wireless communications. The focus now is on developing "lab-on-skin" devices that emulate skin's physical properties for non-invasive, long-term, and continuous health monitoring [11], [12]. Raw data from wearable devices often possess high dimensionality and may contain noise, necessitating the application of machine learning algorithms for effective health analytics [13].

2.2.4 Challenges in Deployment

Despite the potential of IoT and wearable devices in healthcare, several challenges persist. These include ensuring device reliability, managing power, guaranteeing data accuracy, and safeguarding privacy and data security [12], [14]. Moreover, while wearable devices can encourage healthier behaviours, they may inadvertently increase stress and anxiety levels due to constant monitoring, indicating the need to carefully manage the devices, especially in vulnerable populations [15].

2.2.5 Future Trends and Developments

In the foreseeable future, further innovation and advancement in this field are expected to focus on overcoming the challenges above. Research in sensor technology, power systems, device materials, and design strategies is set to increase. The integration of these devices into telemedicine and telehealth systems could lead to a more structured medical Internet of Things (MIoT) ecosystem, ultimately allowing for real-time, continuous monitoring of patients [16], [17].

2.3 Heart Rate and Blood Oxygen Saturation Technologies

2.3.1 Overview of Heart Rate and SpO2 Monitoring

Heart rate and SpO2 are vital physiological parameters, often measured to assess an individual's cardiovascular and respiratory health status. Heart rate reflects the frequency of the heartbeat, typically measured in beats per minute, while SpO2 is an estimation of the amount of oxygen in the blood, expressed as a percentage. Continually monitoring these parameters provides clinicians and individuals insights into their health and wellness, as well as early warning signs of potential health complications.

2.3.2 Technologies for Heart Rate and SpO2 Monitoring

Currently, heart rate is most commonly monitored using photoplethysmography (PPG), a technology that measures blood volume changes in the microvascular bed of tissue using a light source and a photodetector[18]. PPG is incorporated into many wearable devices, such as smartwatches and fitness bands. SpO2, traditionally measured in clinical settings using a pulse oximeter placed on a patient's fingertip, is now also being monitored through wearable devices. These wearables use a similar method to pulse oximetry, which compares the absorption of two different wavelengths of light (one red and one infrared) to determine the oxygen saturation level[19].

2.3.3 Innovations and Developments in Heart Rate and SpO2 Technologies

As the field of wearable health monitoring evolves, numerous innovations and developments have occurred. For example, companies have been developing algorithms to improve the accuracy of PPG sensors during physical activity or under other conditions that might affect readings[20]. Additionally, innovations have been made to miniaturize and reduce the power consumption of these devices while improving their sensitivity and robustness[21]. For SpO2 monitoring, there has been a trend of integrating this feature into smartwatches and fitness trackers, making it accessible for continuous, non-invasive, real-time monitoring. Advanced algorithms and signal processing techniques have also been developed to enhance the accuracy and reliability of SpO2 readings from these devices[22].

2.4 Challenges and Limitations

Despite the advancements, challenges and limitations persist. Accurate and reliable heart rate and SpO2 monitoring remain challenging, especially during physical activity or when the wearable device is not correctly positioned[23]. Additionally, factors such as skin colour, ambient light, and motion artefacts can affect the accuracy of PPG and pulse oximetry[24]. The security and privacy of health data collected by these devices is another significant concern. Protecting this sensitive information from unauthorized access and ensuring it is used ethically is paramount[25]. Furthermore, the regulatory environment for wearable health devices is complex and varies by region, posing challenges for global deployment[16].

2.5 Human Activity Recognition (HAR)

2.5.1 Overview of Human Activity Recognition

Human Activity Recognition (HAR) is a rapidly evolving field of research that revolves around identifying and naming human activities. The recognition of activities ranges from simple actions such as walking, sitting, and running to more complex tasks like cooking or driving. This task is achieved through artificial intelligence and data collected from various types of sensors, which can be deployed in the environment or carried by the person.

These sensors include wearable sensors, smartphone inertial sensors, optical cameras, and frequency-emission recognition systems, each contributing unique data to identify different activities. Due to its capacity to capture and analyse rich data, HAR has found applicability in diverse fields. These include healthcare, human-machine interaction, security systems, industrial mechanisation, athlete training monitoring, robot monitoring, and rehabilitation systems.[26]

The operation of a typical HAR system involves several stages: perception, where the system acquires raw sensor data; segmentation, where the continuous data stream is divided into meaningful segments; feature extraction, where characteristic attributes are derived from raw data; and classification, where the derived features are used to identify the type of activity. Pre-processing steps may also prepare the data for subsequent stages.[27]

The proliferation of sensors and advancements in AI has propelled HAR into the forefront of human-centric technologies. As we continue to make strides in this field, HAR systems hold the potential to revolutionise the way we interact with our environments and with each other.

2.5.2 Different Approaches in HAR

There are several distinct approaches in HAR, each leveraging different sources of data and methodologies.

Device-Free Solutions in HAR

Device-free solutions in HAR generally utilise technologies that do not require the user to wear or carry any devices. This approach can further be divided into vision-based HAR and 3D data-based HAR.

Vision-Based HAR Vision-based HAR employs visual data, often from video feeds, to recognise human activities. With computer vision technology, this method has seen significant development and promises more robust and versatile activity recognition[26], [28]. Techniques like pose estimation, object detection, and scene understanding are commonly used in this approach[29].

3D Data-Based HAR 3D data-based HAR utilises depth maps or point clouds generated from 3D imaging technologies. It has gained increasing attention as it is particularly effective in environments where depth information can provide additional context and detail for the activities[30], [31]. This includes applications in elder care and surveillance, where understanding the spatial context is essential[32].

Wearable Device-Based Solutions in HAR

Wearable device-based HAR is an approach that uses data from sensors embedded in wearable devices like smartwatches or fitness bands for activity recognition[33]. This approach offers a higher accuracy as wearable sensors can directly measure bodily movements and physiological signals[34], [35].

Single Modality HAR This approach uses a single type of sensor data for activity recognition. For example, accelerometers in a smartwatch might be used to detect activities such as walking or running[36].

Multi-Modality-Based HAR Multi-modality-based HAR is an extension of wearable device-based HAR that combines multiple types of sensor data, such as accelerometer readings, heart rate, and possibly even video feed, to enhance the accuracy of activity recognition[37], [38]. By integrating diverse sensor data, this approach can capture a more holistic view of human activities[39].

These diverse approaches showcase the broad spectrum of technologies and methodologies applied in HAR. Each approach has its strengths and limitations, and the selection often depends on the specific requirements and constraints of the application context.

2.5.3 Technologies for Human Activity Recognition

Sensor Technologies in HAR

Sensor technologies provide the essential data that powers HAR systems. These technologies vary depending on the approach used for HAR. They can span from wearable devices to environmental sensors, each capturing a unique type of raw data that is then processed and analysed for activity recognition.

Accelerometers and Gyroscopes Often forming the bedrock of HAR systems, accelerometers and gyroscopes are especially prevalent in wearable and smartphone-based solutions[33]. These sensors measure acceleration and rotational motion, respectively, recording the user's movements with high precision. The data generated can then be used to distinguish between activities, such as walking, running, or sitting, based on unique movement patterns associated with each activity[40].

RF Signals, Wi-Fi, and Radar Technologies like RF signals, Wi-Fi, and radar have proven instrumental in device-free HAR solutions. They detect and identify human activities by measuring signal strength changes induced by human movement[41], [42]. A classic example of this can be seen in Wi-Fi-based device-free activity recognition systems, which leverage variations in Wi-Fi signals caused by human movement to distinguish between different activities.

Camera and 3D Imaging Technologies Camera and 3D imaging technologies are vital for vision-based and 3D data-based HAR. Traditional cameras capture 2D visual data, providing more information about the subject's activities. In contrast, 3D imaging technologies provide an added layer of depth to the data captured, generating depth maps or point clouds. This depth of information gives additional context and detail for activity recognition, enabling a more nuanced understanding of the activities being performed[43].

Data Processing and Feature Extraction in HAR

Data processing and feature extraction are essential components of HAR. These processes transform raw sensor data into meaningful features used for activity recognition. The methods used can vary based on the sensor technologies and specific requirements of the HAR system.

Time and Frequency Domain Features Time and frequency domain features are critical in activity recognition algorithms. Position-independent models utilising frequency domain features have been proposed to tackle the issue of variable smartphone positions in HAR. For example, one method analyses FFT curves of different mobile positions and activities, improving recognition accuracy by 5%[44]. Moreover, time-series sensor data processed through convolutional neural networks (**CNN**) have shown efficient recognition performance, achieving an overall accuracy of 88.23% on a benchmark dataset[45].

Semantic Features Semantic feature extraction is another approach emphasising extracting semantically meaningful features to improve the interpretability and robustness of the HAR systems. This method allows the system to identify activity and understand the underlying intent, enhancing overall recognition performance[46].

R Transform The R transform is a novel feature representation method for HAR. It is particularly effective when dealing with time series data where the temporal order of events is crucial for recognition tasks. The unique aspect of the R transform lies in its ability to cater to time-sensitive activities, offering an edge in HAR[47].

2.5.4 Machine Learning Models in HAR

Machine Learning (**ML**) models form the crux of Human Activity Recognition (HAR), deciphering the meaningful patterns in sensor data to predict and identify human activities. These models range from traditional supervised and unsupervised learning methods to advanced deep learning algorithms, each with strengths and challenges when applied to HAR[48].

Supervised Learning Models

Supervised learning models, trained with labelled datasets, are commonly used in HAR. They include algorithms such as k-Nearest Neighbors (KNN), Decision Trees, Random Forest, and Support Vector Machines (SVM). K-NN is a simple yet effective model for HAR. It classifies activities based on the 'majority vote' of its nearest neighbours in the feature space[33]. Decision Trees and their ensemble variant, Random Forest, offer interpretable models and can handle non-linear relationships, proving useful in HAR[49]. SVMs, known for their effectiveness in high-dimensional spaces, are often employed in HAR, leveraging their ability to maximise the margin between classes[50].

Unsupervised Learning Models in HAR

Unsupervised learning models, which discover inherent structure in unlabeled data, also find use in HAR. Models such as K-Means and Hierarchical Clustering are typically used for preliminary exploration of data or when labelled data are scarce. K-Means clusters similar activities together based on their feature vectors, providing a simple and intuitive way to categorise activities[51]. Hierarchical Clustering offers a more nuanced view, creating a tree-like structure of clusters that can reveal relationships between different activities[52].

Deep Learning Models in HAR

Deep Learning models have gained significant attention in HAR due to their ability to process high-dimensional data and learn complex patterns[53]. Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Long Short-Term Memory (LSTM) networks are among the most commonly used models in this context. CNNs have been employed in HAR to process both temporal and spatial data from sensors, such as 2D images or time-series sensor data[54]. CNNs are particularly adept at extracting complex features from this type of data, which can significantly enhance recognition performance. RNNs and their variant LSTM networks are well-suited for HAR tasks that involve temporal sequences, such as continuous monitoring of activities[55]. They can process time-series data by considering temporal dependencies, making them powerful tools for recognising activities that unfold over time.

Reinforcement Learning Models in HAR

Reinforcement Learning (**RL**) offers a different approach to HAR. In RL, an agent learns to make decisions based on cumulative reward. Deep Reinforcement Learning (DRL), which combines RL with deep learning techniques, has also been applied in HAR[56]. RL models can continually adjust and improve performance over time, making them particularly valuable in dynamic or unpredictable environments. However, their application in HAR is relatively new, and further research is needed to exploit their potential fully. DRL, on the other hand, allows for complex state representation learning through deep learning, which can enhance the ability of the RL agent to understand and react to its environment. These models can be especially beneficial for HAR tasks that involve complex environments or require continuous adaptation to changes[56].

2.6 Future Directions and Open Issues

Human Activity Recognition (HAR) is a rapidly evolving domain with the potential to transform various sectors such as healthcare, security, entertainment, and more. Nevertheless, there are significant challenges and open issues that require extensive research and innovative solutions[33].

2.6.1 Future Research Directions in HAR

As the realm of HAR expands, there are numerous avenues for future research. One crucial aspect is the development of more robust and versatile activity recognition algorithms that can effectively handle noise, cope with different sensor configurations, and adapt to diverse users and contexts. Further, the exploration of new sensor technologies or integration of multiple sensor data can potentially enhance the accuracy and versatility of HAR systems. The expansion of HAR applications, such as in geriatric care, immersive gaming, advanced security systems, or personalised fitness, is another promising direction[57].

2.6.2 Open Issues in HAR

Several open issues in HAR need to be addressed for the successful mainstream adoption of this technology. The availability of standardised and diverse datasets for training and evaluating HAR systems is critical. Most datasets are collected in controlled lab settings and lack the diversity and variability encountered in real-life situations.

Moreover, recognising activities in uncontrolled or unpredictable environments poses a significant challenge due to the variations in how different individuals perform the same activity and the influence of environmental factors. Therefore, developing HAR systems that adapt to individual differences and environmental changes is a pressing need.

Data privacy and security is another concern in HAR, given the sensitive nature of the information collected by the systems. Robust encryption methods, privacy-preserving algorithms, and secure data handling procedures are needed to ensure the users' privacy and gain their trust.

Lastly, the design of energy-efficient HAR systems, particularly for wearable and IoT devices, is an open issue. The limited battery life of these devices necessitates the development of energy-efficient algorithms and hardware to extend their operational time[33].

2.7 Summary

This literature review first introduced the overall topic. Following the introduction, the literature review explored the ongoing integration of advanced digital technologies into healthcare, specifically the Internet of Things (IoT). Emphasis was given to the rising significance of wearable devices, which are rapidly becoming integral to health monitoring and management. In this context, the review sheds light on evolving health paradigms, tackling the challenges in implementing such pervasive technologies and forecasting future trends and advancements.

The review then transitioned into a detailed examination of heart rate and blood oxygen saturation (SpO2) monitoring technologies. An overview of these monitoring techniques was provided, followed by an exploration of the technologies enabling such monitoring. Emphasis was placed on recent innovations and developments within this field.

Subsequently, the focus shifted towards Human Activity Recognition (HAR), starting with an overview of the concept and the different approaches involved in its implementation. Various technologies enabling HAR were discussed, and their applications in different scenarios. The review also explored the role of machine learning models in HAR, covering supervised, unsupervised, and deep learning methods. Potential future directions and open issues in HAR were addressed. The discussion highlighted areas for future research, including the need for more versatile activity recognition algorithms and the exploration of new sensor technologies. Open issues encompassed the need to standardise datasets, recognise activities in uncontrolled environments, ensure user privacy and data security, and design energy-efficient systems.

In conclusion, the literature review offered a comprehensive understanding of the current state of IoT and wearable devices in healthcare, heart rate and SpO2 monitoring technologies, and the evolving field of HAR. The challenges, developments, and open issues identified through the review set a clear direction for future research in these domains.

Chapter 3

System Requirement

3.1 System Function Overview

This project is about developing a user-friendly health tracking system for real-time monitoring.

The system can measure heart rate and SpO2 level and detect user activities by employing **IMU** sensors and machine learning techniques. It provides users with immediate access to their physiological data and activity recognition results via an Android app; it also provides useful health advice based on the collected data.

For heart rate and SpO2 level monitoring, users connect the Arduino to their Android phones via the Android application. After placing their finger on the sensor, the relevant physiological data is presented on the app's interface. In case of Bluetooth connection failure or improper sensor placement, an automated message is sent to the user, prompting them to rectify the issue.

For activity recognition, users need to pair the Thingy sensors with their MAC addresses within the app and wear them appropriately. After processing the incoming IMU data through a pre-defined ML model, the detected activities are identified and displayed on the app's interface. Two sensors are used to enhance the accuracy of the recognition. The weight of the two sensor's results is adjusted based on the sensors' performance using linear combination.
The system also includes a history feature, using a histogram to visually represent the recorded activities, including the activity type, date and duration. To promote a healthier lifestyle, the app has an embedded feature that sends a notification to the user if prolonged inactivity, such as sitting or desk work, is detected.

In terms of physiological norms, heart rates ranging from 60 to 100 bpm and SpO2 levels between 95% to 100% are considered healthy. However, these parameters may vary based on the user's activity. Therefore, the system intelligently takes into account both the heart rate and SpO2 readings, along with the HAR outcome, to provide personalised and relevant health advice.

Figure3.1 illustrates the function block diagram for the project.

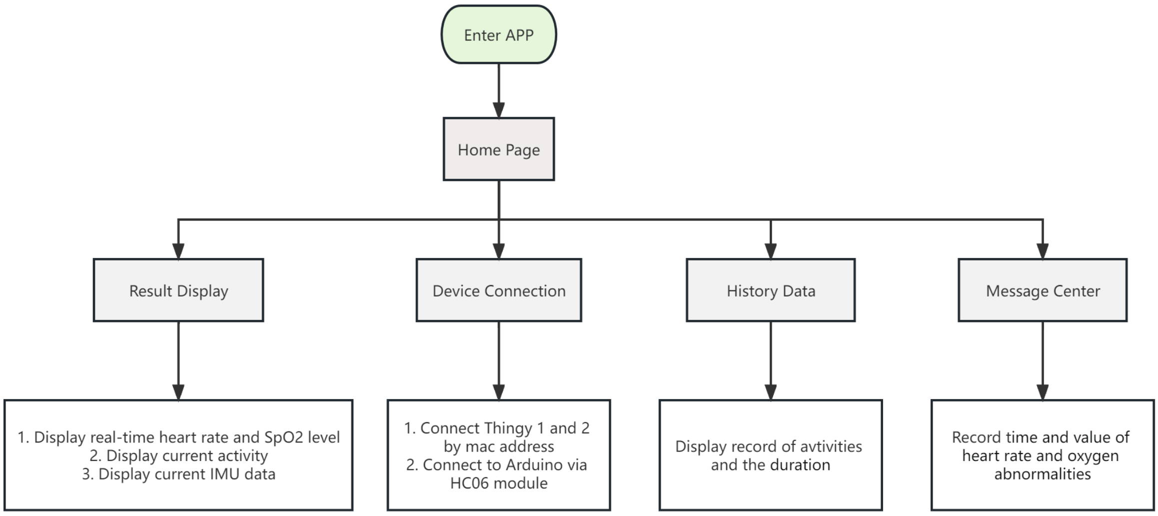

Figure 3.1: Function block diagram

3.2 Development Tools

In the process of developing the system, a set of specific hardware and software tools are utilized to ensure efficient development and effective outcomes. The following are the most important ones.

3.2.1 Hardware

Arduino Development Board

Arduino Uno is an open-source microcontroller board featuring the ATmega328P. The module is as shown in Figure 3.2.

In this project, an Arduino development board served as the central controller for the MAX30102 sensor and the HC-06 Bluetooth module, enabling the reading of the sensor data and sending it to the smartphone application.

MAX30102 Pulse Oximeter and Heart Rate Sensor

The MAX30102 is an integrated pulse oximeter and heart rate sensor module that combines two LEDs (RED and IR), a photodetector, optimized optics, and low-noise analogue signal processing. It measures SpO2 and heart rate signals by shining the LEDs and detecting the reflected light's amount. The module

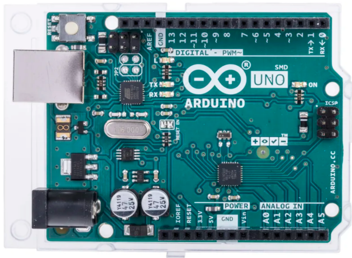

Figure 3.2: Arduino UNO development board

operates efficiently with a power consumption of less than 600µA during measurement and 0.7µA in standby mode, making it ideal for battery-powered devices. The MAX30102 module operates within a power supply range of 3.3V to 5.5V, with a temperature range of -40 to +85 degree celsius.The module is as shown in Figure 3.3.

In the project, the MAX30102 sensor was used to monitor the user's heart rate and SpO2 levels.

Figure 3.3: MAX30102 Sensor

HC-06 Bluetooth Module

The HC-06 is a Bluetooth module designed for efficient bidirectional serial communication. Operating on the Bluetooth 2.0 communication protocol, it functions exclusively as a slave device. The HC-06 module provides a cost-effective and flexible solution for wireless data transmission, supporting speeds up to 2.1Mb/s. Its operating frequency aligns with the widely-used 2.4GHz ISM band, adhering to Bluetooth 2.0+EDR standards. These standards allow for a 0.5-second signal transmit time interval between devices, significantly reducing the Bluetooth chip's workload and conserving power. The module is as shown in Figure 3.4.

The HC-06 module was used to establish the connection between the Arduino board and the Android smartphone.

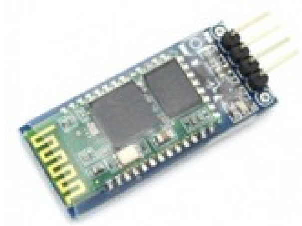

Figure 3.4: HC-06 Bluetooth Module

Android Smartphone

Android, an open-source and Linux-based operating system developed by the Open Handset Alliance, led by Google, powers the Android Smartphone used in this project. With its rich ecosystem of development tools, libraries, and APIs, it allows the creation of robust, responsive, and user-friendly applications.

This project uses the smartphone's Bluetooth capabilities to receive data from the HC-06 module, thus enabling real-time monitoring. The smartphone also facilitates the user to visualize and understand the data through a well-designed application interface. Furthermore, smartphones can be used to set up alerts and store historical data.

Nordic Thingy:52 IoT sensor kit

The Nordic Thingy:52 IoT sensor kit is a flexible prototyping platform. It is built around the nRF52832 Bluetooth 5 System on Chip (SoC) and has multiple sensors integrated within. It can remotely configure sensors and actuators through Bluetooth Low Energy, enhancing the kit's convenience and adaptability. The module is as shown in Figure **??**. In this health monitoring project, two Nordic Thingy:52 IoT

Figure 3.5: Nordic Thingy:52 IoT sensor

sensor kits are utilized, primarily leveraging their Inertial Measurement Unit (IMU) sensor, including an accelerometer and gyroscope.

The first Thingy:52 sensor is positioned under the left rib cage, as shown in Figure3.5. This location is close to the human body's centre of gravity, where most physical activities generate discernable movements. The accelerometer measures acceleration while the gyroscope monitors angular movement. The combined use of them provides richer information about the body's activities, enhancing accuracy. Furthermore, this placement is advantageous for detecting respiratory states, which further helps classify types of human activities. Data from these sensors is collected and transmitted to the Android app via Bluetooth Low Energy (BLE) at a frequency of 25Hz. The second Thingy:52 sensor is worn in the right

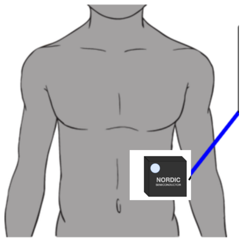

Figure 3.6: Thingy on the upper body

front pocket of the trousers. This placement enable the sensor to take leg movement data. Its data is collected and transmitted at the same frequency as the first sensor. It improves the overall accuracy of action recognition when combined with data from the first Thingy:52. The wearing position of the second Thingy:52 is illustrated in the following Figure3.7.

Figure 3.7: Thingy on the lower body

3.2.2 Software

Arduino IDE

The Arduino IDE is an open-source environment for writing and uploading code to development boards. With its user-friendly interface and support for C and C++ languages, it is ideal for controlling the MAX30102 pulse oximeter and HC-06 Bluetooth module.

Android Studio

Android Studio is the official IDE for Android app development. It is used to develop the application that collects, processes, and displays health data for the user.

Jupyter Notebook

Jupyter Notebook is an open-source web application that enables users to create and share documents with live code, equations, and visualizations. It was mainly used to develop and evaluate the machine learning model for HAR.

Chapter 4

Design and Implementation

4.1 System Architecture

The architecture of the health monitoring system is divided into two primary parts. The first part is devoted to configuring and debugging sensor-related components, a vital phase for ensuring the accurate collection of physiological metrics. The second part entails the construction of the Android application, which acts as the conduit between the user and the system, delivering real-time health insights and recommendations.

4.2 Sensor Deployment

4.2.1 Heart Rate and SpO2 Measurement System

The hardware constituents of the system encompass an Arduino UNO development board, a MAX30102 sensor, an HC06 Bluetooth module, a breadboard, jumper wires, and resistors. The system employs the Arduino platform for sensor interfacing and data acquisition.

Interfacing the MAX 30102 Sensor

The process of interfacing the MAX30102 sensor with the Arduino Uno involves a number of steps. First, the power supply connection is established by connecting the VCC pin on the sensor to the power supply (5V) of the Arduino, ensuring alignment with the microcontroller's logic voltage. Subsequently, the GND is connected to the common ground. The I2C protocol is used to connect the sensor and Arduino, with the SCL pin of the MAX30102 connected to the I2C clock pin on the Arduino, and the SDA pin to the I2C data pin.

The SparkFun MAX3010x Pulse and Proximity Sensor Library and the MAX3010x Sensor Library are utilized for debugging and developing due to the simple and user-friendly functions, which facilitate calculating the pulse rate and SpO2. The library is installed from within the Arduino IDE Library Manager.

Once the library is installed, example sketches provided can be adapted for custom development. Subsequently, code leveraging the library functions to read sensor data and calculate heart rate and SpO2 levels is written and uploaded to the Arduino. Figure4.1 shows the wiring.

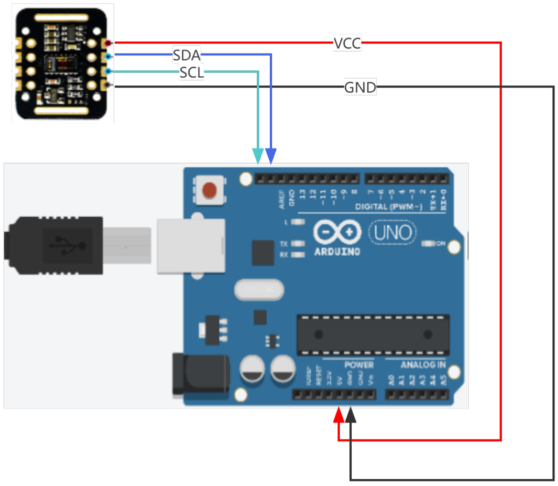

Figure 4.1: MAX30102 and Arduino wiring

Bluetooth Module Configuration

The HC-06 Bluetooth module is used for establishing short-range wireless data communication. It is highly compatible and can be interfaced with virtually all microcontrollers through its UART interface. Notably, this module operates on the Bluetooth 2.0 communication protocol and exclusively acts as a slave device.AT commands can be used to configure the module.

The HC-06 Bluetooth module operates within a voltage spectrum of $3.3V$ to $6V$, and a frequency bandwidth of $2.402GHz$ to $2.480GHz$. It features four key pins:

- RXD: The serial data receive pin, handling serial input at $3.3V$.
- TXD: The serial data transmit pin, managing serial output at $3.3V$.
- GND: Ground pin.
- VCC: A $+5V$ power pin.

The module's VCC connects to the Arduino's $+5V$, and its GND aligns with the Arduino's GND. Theoretically, the module's RXD would connect to the Arduino's TX, and its TXD would link to the Arduino's RX. However, this configuration would potentially overload the $3.3V$-tolerant RXD and TXD pins with the Arduino's $5V$ output. Short-term use may not result in damage, but prolonged exposure might burn the module. Thus, a voltage divider circuit is integrated with the HC-06's RXD pin to downscale the Arduino's higher voltage output to a level the HC-06 can safely manage. A $10k\Omega$and a $20k\Omega$ resistor in the voltage divider circuit achieve this reduction. The voltage (V_{RX}/V) and current (I/A) are calculated using the following formulas:

$$I = \frac{5V}{10k\Omega + 20k\Omega} = 0.16mA \tag{4.1}$$

$$V_{RX} = I \times 20k\Omega = 3.2V \tag{4.2}$$

Figure4.2 shows the wiring.

It is vital to remember that, once the Arduino's RX (pin 0) and TX (pin 1) pins are linked to external devices, program uploading becomes inaccessible since these pins double as the programming port. To circumvent this, just disconnect wires from pins 0 and 1 for uploading. When the Bluetooth module and Arduino board utilized independent power sources, their ground connections were linked to preserve a shared voltage reference.

Serial library functions enabled communication in the Arduino code. The module's operational functionality was verified by establishing successful bi-directional communication with an Android device, using a Bluetooth terminal app. Messages were successfully sent from the app to the Arduino IDE's Serial Monitor, and vice versa, confirming effective Bluetooth communication.

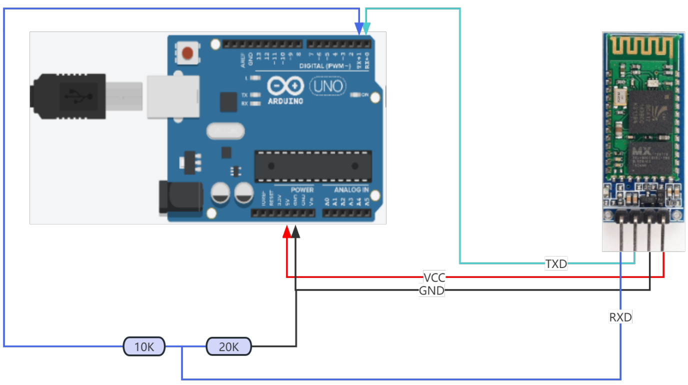

Figure 4.2: HC06 and Arduino wiring

Integration

The integration of the MAX30102 sensor and the HC06 Bluetooth module is easy, primarily because there is no pin conflict between the two devices. To enhance the user experience, I have incorporated a LED connected to pin 11, which serves as a visual indicator for successful Bluetooth transmissions.

A breadboard is used to organise and interconnect the components to simplify the setup. Figure4.3 shows the simulation diagram and Figure4.4 shows the physical diagram for this part. Apart from

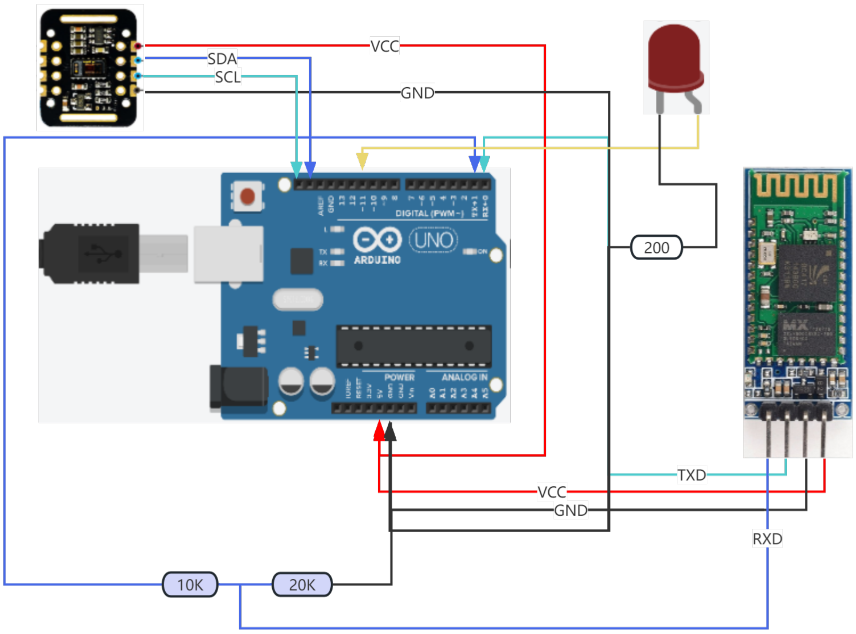

Figure 4.3: Wiring for the Arduino system

the hardware, proper Arduino code is used. The main parts of the code include sensor and serial port initialization, finger detection, data filtering, heartbeat detection, heart rate and SpO2 calculation, and data output.

Initialization The code begins with the inclusion of necessary libraries and setting up essential parameters. The *MAX30105* library is utilised to interface with the MAX30102 sensor. This library provides an abstraction layer for the MAX30102, simplifying the sensor initialisation and data reading. Other parameters define the sensor's sampling rate, finger detection threshold, and edge detection threshold. The 'setup()' function initialises the sensor, sets up an LED on pin 11, and begins serial communication. If the sensor is not found, the Arduino enters an infinite loop, halting the program.

Reading Sensor Data The 'loop()' function is the main body of the code, executed repeatedly as long as the Arduino is powered. Data is read from the MAX30102 sensor at the start of each loop. The sensor.readSample(1000) function is called to read a sample from the sensor.

Finger Detection The presence of a finger is checked by comparing the raw sensor red value to a predefined threshold. A finger is considered present if the reading is above the threshold and the time elapsed since the last finger detection is greater than the predefined cooldown time.

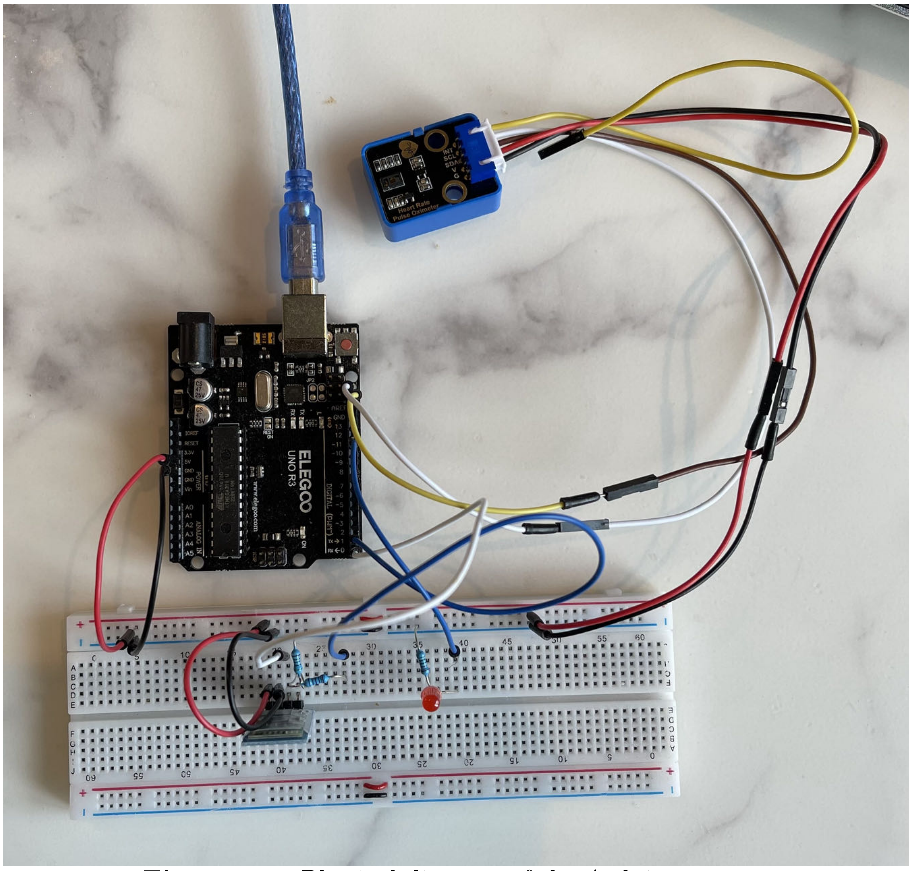

Figure 4.4: Physical diagram of the Arduino system

Data Filtering Upon finger detection, data goes through two filtering processes: a low-pass filter to remove high-frequency noise and a high-pass filter to eliminate dc components and drifts in the data. These filtered values are then utilised for further processing.

Statistics Updating Next, the code updates statistics for pulse oximetry. A MinMaxAvgStatistic object is utilised to track the readings' minimum, maximum, and average values from the red and infrared light sensors.

Heartbeat Detection The code uses the differentiator function to detect changes in the signal's slope, which helps identify the heartbeats. If the slope of the signal crosses zero from positive to negative (indicating a peak), a potential heartbeat is detected. A subsequent steep drop in the signal's slope confirms the heartbeat.

Heart Rate and SpO2 Calculation Upon heartbeat confirmation, the code calculates the heart rate and SpO2 levels. The heart rate is derived from the time difference between consecutive heartbeats. The

SpO2 level is calculated using the ratio of red and infrared light absorption per the empirical equation from the MAXIM application note.

Data Output The code outputs the computed heart rate and SpO2 values to the serial monitor and the Bluetooth terminal. A LED light is also toggled on and off each time a heartbeat is detected, and these values are calculated.The data is transmitted in JSON format.

JSON (JavaScript Object Notation) is a lightweight data exchange format that is simple to read and write for humans while also being simple for machines to understand and produce. The JSON data in this context contains two properties, namely `Spo2` and `HeartRate`. A sample JSON string would look like this:
{"Spo2": 98.6,"HeartRate": 72}
Figure4.5 shows the block diagram for the Arduino code.

4.2.2 Configuring the Nordic Thingy:52

Setting up the Thingy:52 requires the official app to adjust the sampling rate to 25Hz. The app is available for download from the Google Play Store. After turning it on and connecting it to the app, the next step is to navigate to the "Configuration" section to set the frequency of the motion processing unit to 25Hz, preparing the Thingy:52 for operation.

4.3 Android Application Development

The Android application acts as the interface of the system. The user interacts with this layer for real-time health monitoring and advice. It is responsible for getting and processing data from Bluetooth and hold the machine learning model.

4.3.1 Machine Learning Model

The machine learning model is a key part in the HAR subsystem. This model is trained to identify and classify a range of human activities based on sensor data. After designing and training the model, the model is exported as a TFLite file for further integration.

Data Collection

The data utilized in this research is gathered from an open-source repository, which can be accessed from the following URL: `https://github.com/specknet/pdiot-data.git`. This data was collected from IMU sensors during fourteen different activities. These activities encompass various postures, such as sitting in different positions (straight, bent forwards, bent backwards), standing, and lying down in different orientations (on the left, right, back, and front). Dynamic actions like walking, running or jogging, ascending and descending stairs, desk work, and general movement (such as sudden turns, bending down,

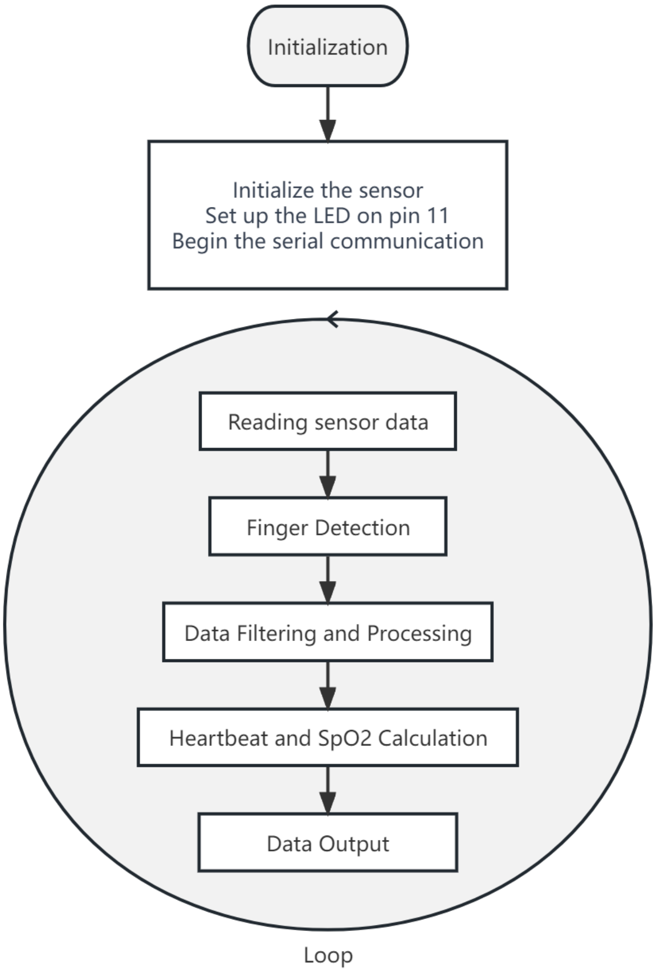

Figure 4.5: Block diagram for the Arduino code

getting up from chairs, and other movements that do not qualify as a specific activity) were also included.

Data was procured from two sensors placed on the upper and lower body. The upper body sensor was positioned on the lower left ribcage. The lower body sensor was placed in the front right trouser pocket. Both sensors sampled accelerometer data measuring acceleration and gyroscope measuring angular velocity data at a frequency of 25Hz.

The collected data is stored in a CSV format with the following headers: timestamp, accel_x, accel_y, accel_z, gyro_x, gyro_y, gyro_z, sensor_type, and activity_type. The 'timestamp' column represents the time at which the data was collected, 'accel_x', 'accel_y', and 'accel_z' are the readings from the accelerometer for the x, y, and z axes, 'gyro_x', 'gyro_y', and 'gyro_z' are the readings from the gyroscope for the x, y, and z axes respectively. The 'sensor_type' column indicates whether the data originated from the upper or lower body sensor, and 'activity_type' denotes the activity performed when the data was collected.

Data Preprocessing

Data preprocessing is a crucial step in the machine-learning pipeline. During this phase, the raw data is transformed into a format that the machine learning model can readily ingest. Several preprocessing steps are undertaken to ensure the data is in a suitable format.

Loading Multiple Files into a DataFrame The data in the repository is stored in multiple files. Each file represents a separate recording and contains data from the accelerometer and gyroscope sensors. The first step involves loading these multiple files into one large data frame.

Sliding Windows Once the data is loaded into a single DataFrame, the next step is to split it into sliding windows. In the context of time series data, such as that from the accelerometer and gyroscope sensors, a sliding window approach allows the model to consider a subset of consecutive data points as one instance. This is essential because the classification of the current activity is often dependent on the immediate past and future data points.

In this script, a window size of 50 is chosen, which equates to 2 seconds of data at a sampling rate of 25Hz. The windows have an overlap of 50%, as specified by the step size of 25. Using overlapping windows ensures that no important transition between activities is missed.

While creating the sliding windows, care is taken to ensure that separate recordings are not included in the same window. This is because each recording is an independent data stream, and combining different recordings in the same window may introduce erroneous jumps in the signal.

The data is first grouped by recording ID, and then each group is split into windows. Finally, all the resulting windows are aggregated into a single dataset for model training.

Model Structure

The machine learning model employed in this project is a sequential Convolutional Neural Networks (CNN) model. The structure of the ML model is depicted in Figure4.6. The Convolutional Neural Network (CNN) model used in this project is intricately designed with several critical parameters. The number of filters is set to 128. Filters help detect distinct patterns within the input data, enhancing the model's predictive accuracy. The kernel size, which indicates the window size for scanning through the input data, is set to 3. This means that the model considers three consecutive units of data to detect patterns effectively. The Rectified Linear Unit (ReLU) is chosen as the activation function. It introduces non-linearity into the model, equipping it to learn complex relationships within the data. Lastly, the

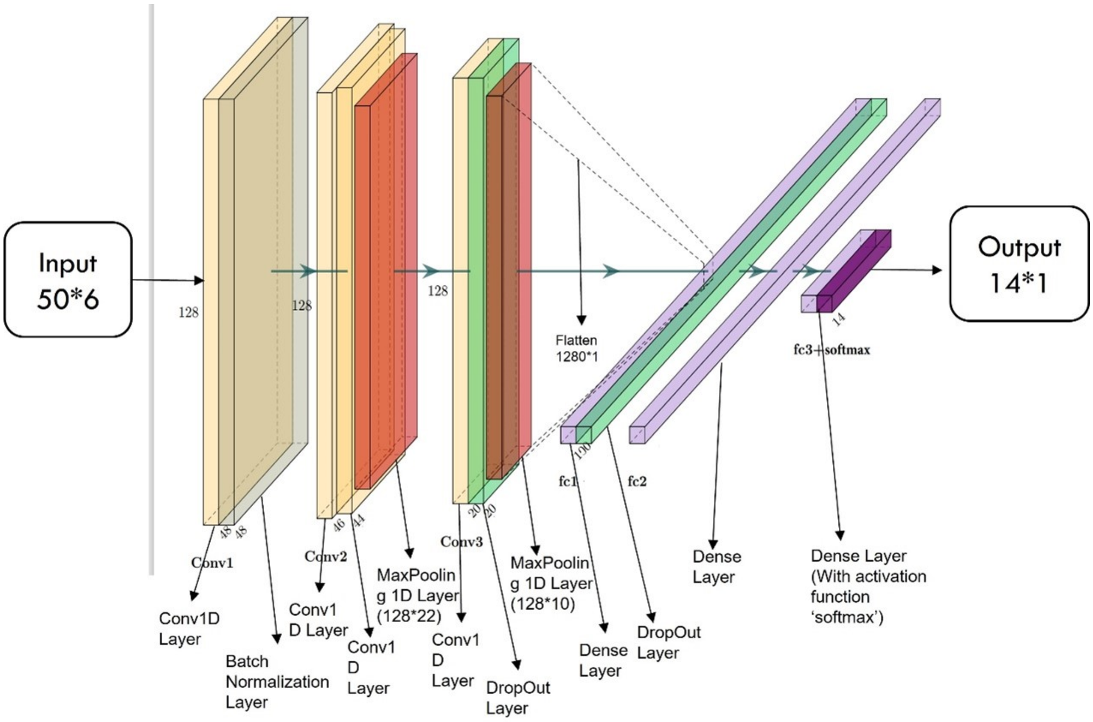

Figure 4.6: Architecture of the CNN model

model is trained to recognise 14 classes of physical activity. The softmax activation function in the output layer provides a probability distribution over these classes, thereby predicting the likelihood of each activity.

The model consists of various layers as follows:

- **Input Layer:** The input data has a dimension of 50*6, where 50 is the model window size and 6 corresponds to the number of data features, encompassing three accelerometer data and 3 gyroscope data.
- **Convolution 1D Layer:** Four Conv1D layers follow the Input Layer. These layers perform convolution operations along the width of the input, extracting features at varying levels of complexity.
- **Batch Normalization Layer:** A Batch Normalization layer is added after the first Conv1D layer to reduce overfitting. This layer works by normalizing the input or output of the Conv1D layers, speeding up training, and providing some regularisation effect, preventing overfitting.
- **MaxPooling Layer:** Two MaxPooling layers are introduced to further prevent overfitting. These layers downsample the input's dimensionality by taking the maximum value over a patch.
- **Dropout Layer:** Two Dropout layers are used to enhance the model's generalisation ability. They achieve this by randomly disabling a proportion of hidden units in the model during training.

- **Flatten Layer:** A Flatten layer is used to transform the multi-dimensional tensor into a single dimension. This transformation is necessary to ensure compatibility with subsequent Dense layers.

- **Dense Layer:** Three Dense (fully-connected) layers are utilised to perform classification based on the combination of features. The activation functions for these layers are ReLu for the first two layers and softmax for the final layer, allowing the model to output a probability for each of the 14 activity classes. The dimension of the output 14*1

Model Training

The model was trained using the clean data provided for 2022 and 2021. During training, an 80%:20% split was applied where 80% of the data was used for training and the remaining 20% for testing. This partitioning validated the model's performance on unseen data and ensured that the model could generalise well beyond the training data.
The training phase ran for 100 epochs with a batch size of 128. The epoch parameter specifies the number of times the entire training dataset is used to update the model parameters, and the batch size denotes the number of samples propagated through the network before updating the model parameters. The choice of 100 epochs and a batch size of 128 was intended to balance computational efficiency with the accuracy of the model's predictions.
A critical component of the training process was the selection of the loss function and the optimizer. For this multi-classification problem, the categorical crossentropy loss function was chosen. Categorical crossentropy quantifies the difference between the predicted and actual distributions and it demonstrated superior performance compared to other loss functions in preliminary tests.
The Adam optimizer, an effective algorithm that combines the advantages of Adagrad and momentum gradient descent, was deployed to update the model parameters. Its selection was backed by its capabilities in managing sparse gradients and reducing gradient oscillation. Compared to other optimizers, such as SGD, Adam provided the best results.
The model's performance and robustness were further evaluated using a 5-fold leave-one-subject-out cross-validation (LOSOXV) strategy. This validation technique offers an unbiased estimate of model performance on new data, making it less susceptible to variance in the dataset. In this method, the means of the folds are pooled together, providing an overall measure of the model's true validation accuracy.
Through this comprehensive training and validation process, the sequential CNN model was fine-tuned to classify physical activities based on sensor data accurately. The choices made during the training, from data splitting to loss function and optimiser selection, were all crucial steps in achieving a highly accurate and efficient predictive model.

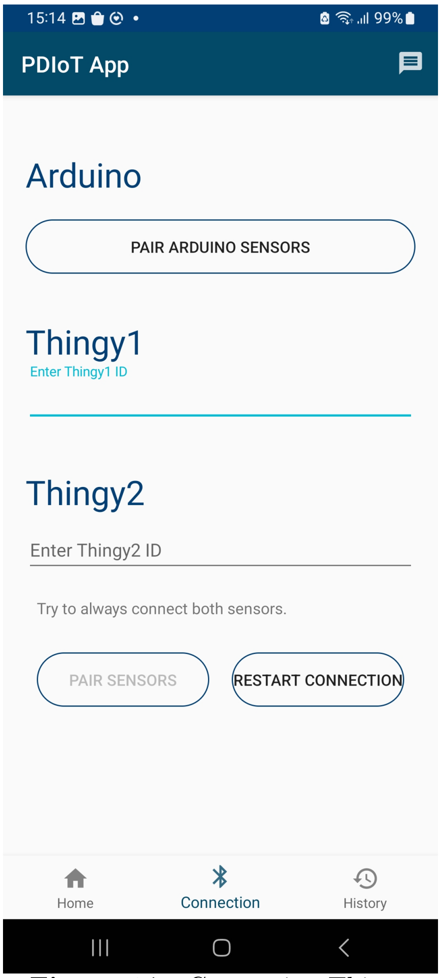

Figure 4.7: Connecting Thingy

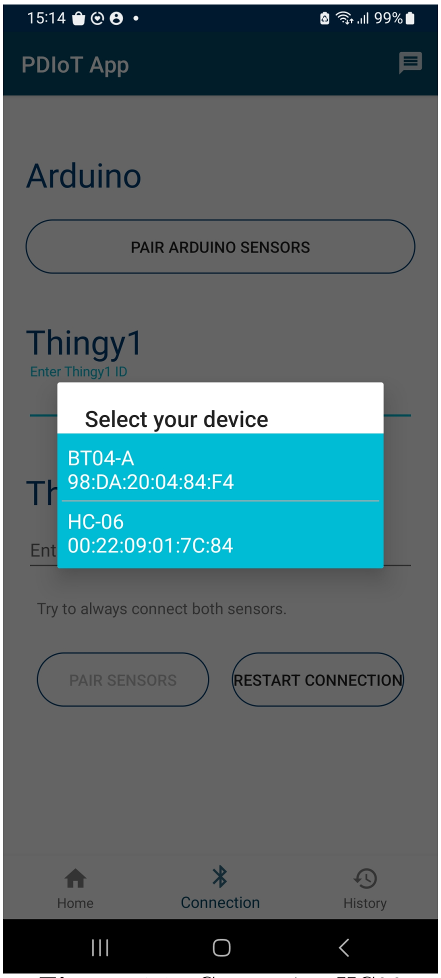

Figure 4.8: Connecting HC06

4.3.2 Application Interface Design

The application's user interface (UI) employs fragments to structure its functionalities. A navigation bar at the bottom of the screen offers access to three pivotal functions: 'Home', 'Sensor Connection', and 'History' using the ViewPager framework.

Bluetooth Connection

Users initially need to use the 'Sensor Connection' feature to pair the application with the Arduino and Thingy52 sensors. Users should ensure that their phone's Bluetooth is turned on and the sensors are activated before proceeding with the connection process. To connect with the MAX30102 sensor, the user needs to select 'Pair Arduino Sensor' and choose 'HC06' from the device list. For the Thingy sensors, the user can manually enter the MAC address on the device and tap the 'Pair Sensor' button.Figure4.7and Figure4.8shows the UI of the connecting function.

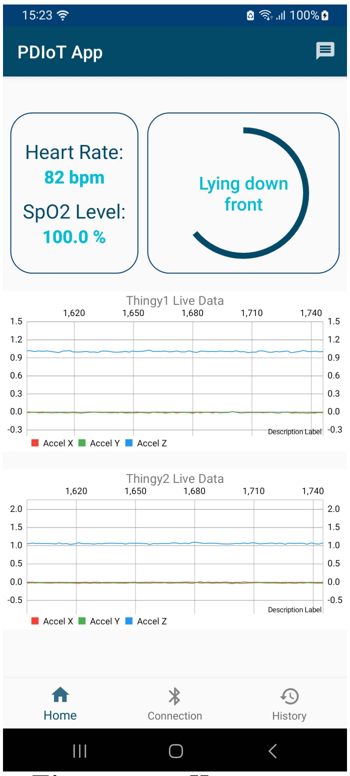

Figure 4.9: Home page

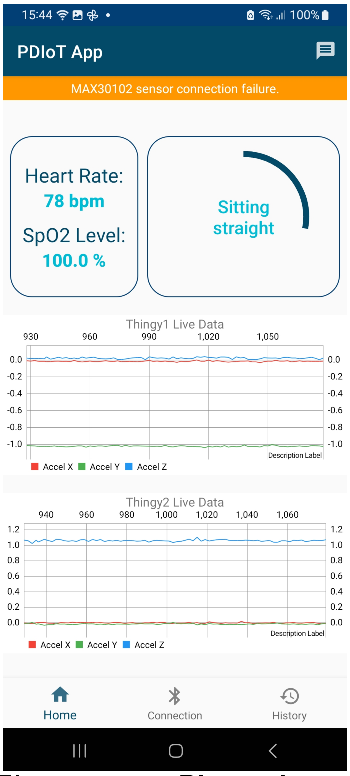

Figure 4.10: Bluetooth connection alert

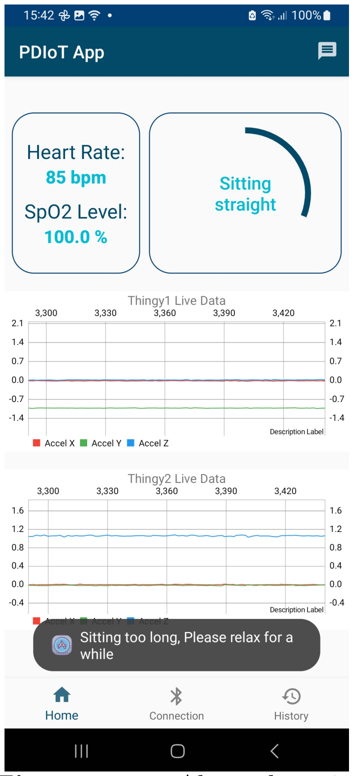

Figure 4.11: Alert when sitting too long

Home

Once connected, the 'Home' page becomes the primary interface for the user. The top left corner displays real-time heart rate and SpO2 levels. If the Arduino sensor data isn't received for a period, a prompt will appear to remind the user to verify the sensor connection. Regarding activity prediction, a dedicated card in the top right corner conveys the result textually. An accompanying circular component visually illustrates the prediction's probability—for instance, three-quarters of the circle turns blue when the probability is 75%. These predictions typically have a two-second delay due to the data processing time. The system also monitors sedentary behaviour—if the user remains seated too long, the app prompts them to take a break. Two live graphs at the bottom of the page showcase real-time accelerometer data from both Thingy sensors.Figure4.9, Figure4.10and Figure4.11shows the UI of the related function.

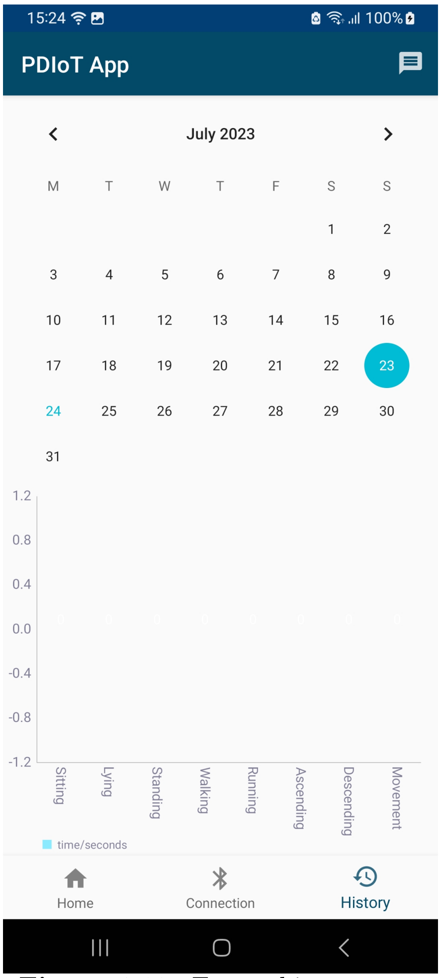

Figure 4.12: Empty history page

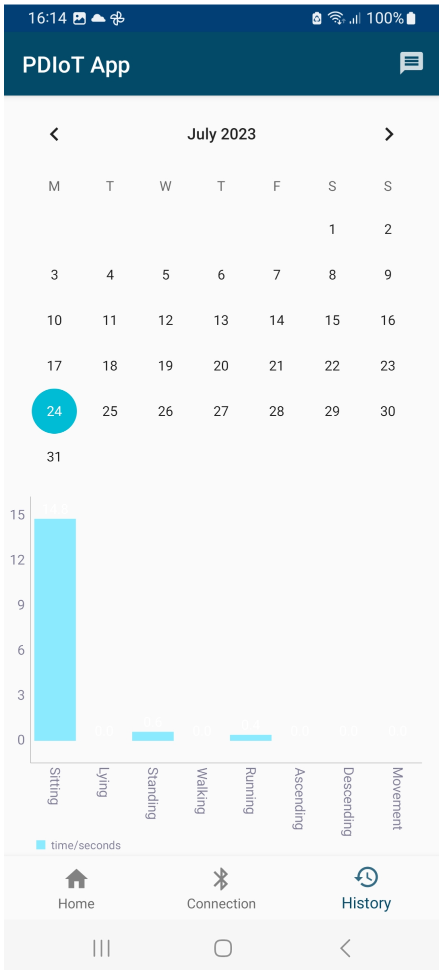

Figure 4.13: History page with data

History

The 'History' function, accessible from the navigation bar, allows users to review past activity data. By selecting a date, the app displays a bar chart enumerating each activity conducted that day, with the activity name on the horizontal axis and the duration on the vertical axis. Figure4.12and Figure4.13shows the UI of the history function.

Message Center

Finally, the application features a 'Message' button at the top right corner. It notifies users of any detected abnormalities in heart rate and SpO2 levels, providing the specific time and value of the abnormal readings. Figure4.14shows the design of the alert message.

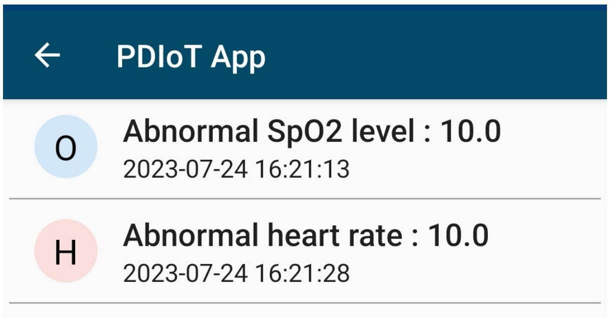

Figure 4.14: Alert message

4.3.3 Application Implementation Details

Figure 4.15illustrates the system structure.

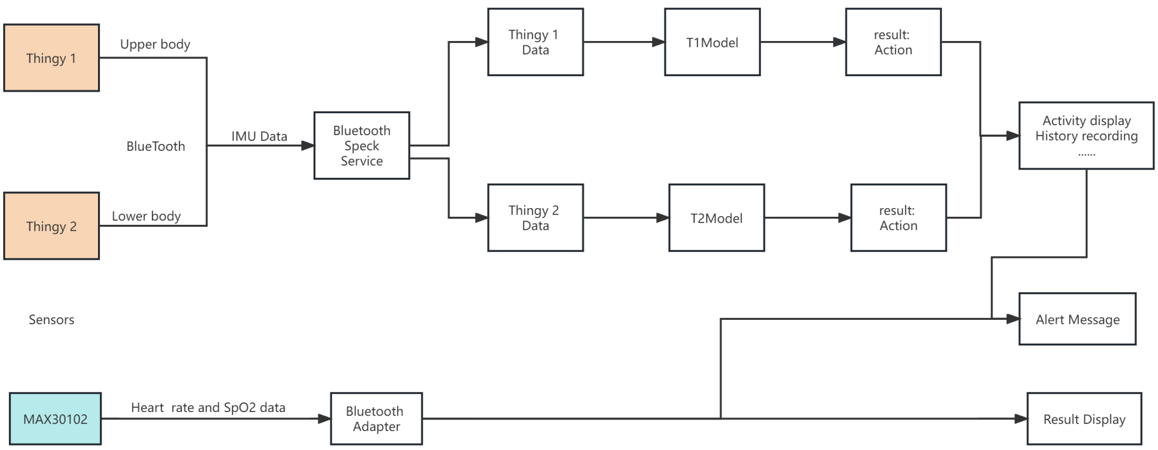

Figure 4.15: System structure

Bluetooth Connection

Figure4.16shows the block diagram of the Bluetooth connection. The connection the Arduino with android Bluetooth function is on the left hand side and the connection with thingywith BLE is on the right hand side.

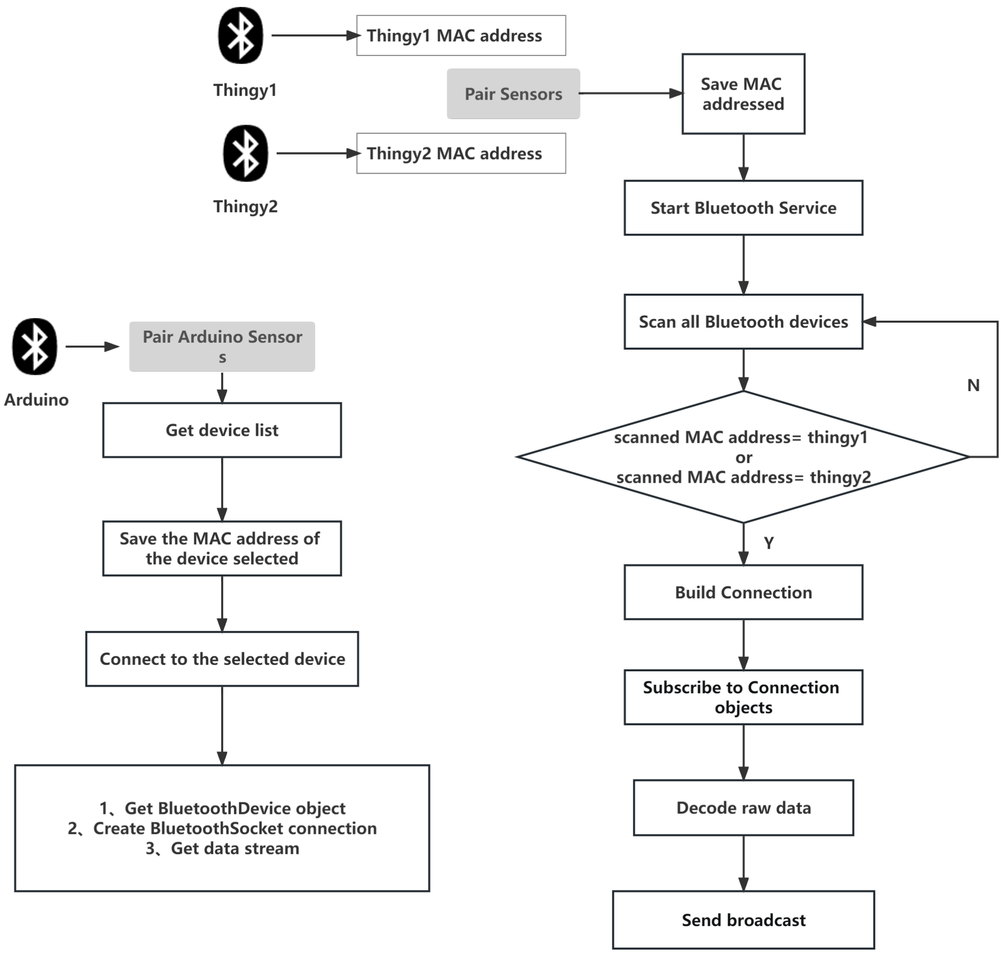

Figure 4.16: Structure for Bluetooth connection

Bluetooth for Arduino The connection between an Android device and the HC-06 Bluetooth module is managed by the `DeviceListDialog` class. This process can be summarized into the following key steps:

1. **Display Paired Bluetooth Devices:** The `DeviceListDialog` class generates a dialogue to present all paired Bluetooth devices to the user in a ListView format.

2. **Fetch and Display Paired Devices:** The `BluetoothAdapter#getBondedDevices` method retrieves a set of `BluetoothDevice` objects representing all paired Bluetooth devices. Each `BluetoothDevice` object contains device-specific information such as MAC address and device name. This information is added to an `ArrayAdapter` and displayed on the ListView.

3. **Device Selection and Connection:** A click listener is set for the ListView. When a device is selected, the `onDeviceSelected` method of the `DeviceSelectedListener` interface is invoked. The `BluetoothConnectTask` saves the MAC address of the selected device and calls the `BluetoothManager#connectTo` function to initiate the connection process.

4. **Data Stream Establishment:** The `BluetoothDevice` object is retrieved using the saved MAC address. A `BluetoothSocket` connection is created using the `createRfcommSocketToServiceRecord` method, and the input data stream from the Bluetooth device is obtained through the `BluetoothSocket#getInputSt` method.

Bluetooth for Thingy

1. **Data Persistence:** SharedPreferences interface is used to store primitive data in key-value pairs, ensuring persistent data across the application's lifecycle.

2. **Bluetooth Service:** BluetoothSpeckService is started. This service handles tasks related to Bluetooth connectivity and data transfer in the background.

3. **Bluetooth Device Scanning:** The application scans for all nearby Bluetooth devices using the RxBleClient#scanBleDevices method.

4. **Device Connection:** Connections are established with devices matching the MAC address of Thingy1 or Thingy2 using the device.establishConnection method.

5. **Subscription and Data Processing:** The application subscribes to the Thingy1 device using the setupThingy1Subscription method, allowing it to receive and process accelerometer and gyroscope data.

6. **Data Packaging and Broadcasting:** The processed data is packaged into a ThingyLiveData object and broadcasted, allowing other components to handle this data.

Home

Heart and SpO2 display The `HomeFragment` class is a Fragment in an Android application that displays SpO2 and heart rate data received from a connected Bluetooth device. It parses incoming data from a JSON format, and updates the UI to display this data, with the following key components:

1. **Initialization of Variables:** This includes TextViews for displaying the heart rate and SpO2 data, an error layout for sensor anomaly prompts, a Handler object for running code on the main thread, and lists to store recent heart rate and SpO2 data.

2. **onCreateView Method:** Called when the fragment's user interface is drawn. It inflates the layout for this fragment from an XML file.

3. **onViewCreated Method:** Called immediately after onCreateView. It locates and initializes the required UI elements in the layout.

4. **addIncomingDataListener Method:** This method adds the fragment as a listener for incoming data from the Bluetooth device.

5. **onDataReceived Method:** This method handles incoming data. It processes the JSON string to extract the heart rate and SpO2 values, and updates the UI to display these values. If the incoming data cause the lists to exceed a certain size, the oldest entry is removed. An average is calculated and if it falls outside the normal range, an alert is triggered and the lists are cleared.

The block diagram of this function is shown in Figure 4.17.

Human Activity Recognition The `HomeFragment` class is also responsible for a real-time activity recognition feature. This activity utilizes the users' motion data and a machine-learning model. Here is a high-level overview of the important components and functionalities:

1. **Initialization of Variables**: At the start of the activity, several variables associated with the User Interface (UI), graph datasets, broadcast receiver for sensor data, and data tracking are initialized.

2. **Activity Creation (onCreate Method)**: This method is triggered when the activity is initiated. It sets up the UI elements and graph datasets. Additionally, it registers a broadcast receiver that listens for incoming sensor data and is programmed to operate on a separate thread to prevent UI disruptions.

3. **Data Receiver and Processing**: The broadcast receiver listens for sensor data broadcasts. Upon receipt, it checks the data, extracts the necessary sensor data, and updates the graph and data collection. Once a specific amount of data (50 data points) is accumulated, it invokes a TensorFlow Lite model for activity recognition, processes the data, identifies the activity with the highest probability, and updates the UI with this information. If sedentary activities are detected for an extended period, a reminder to take a break is presented to the user.

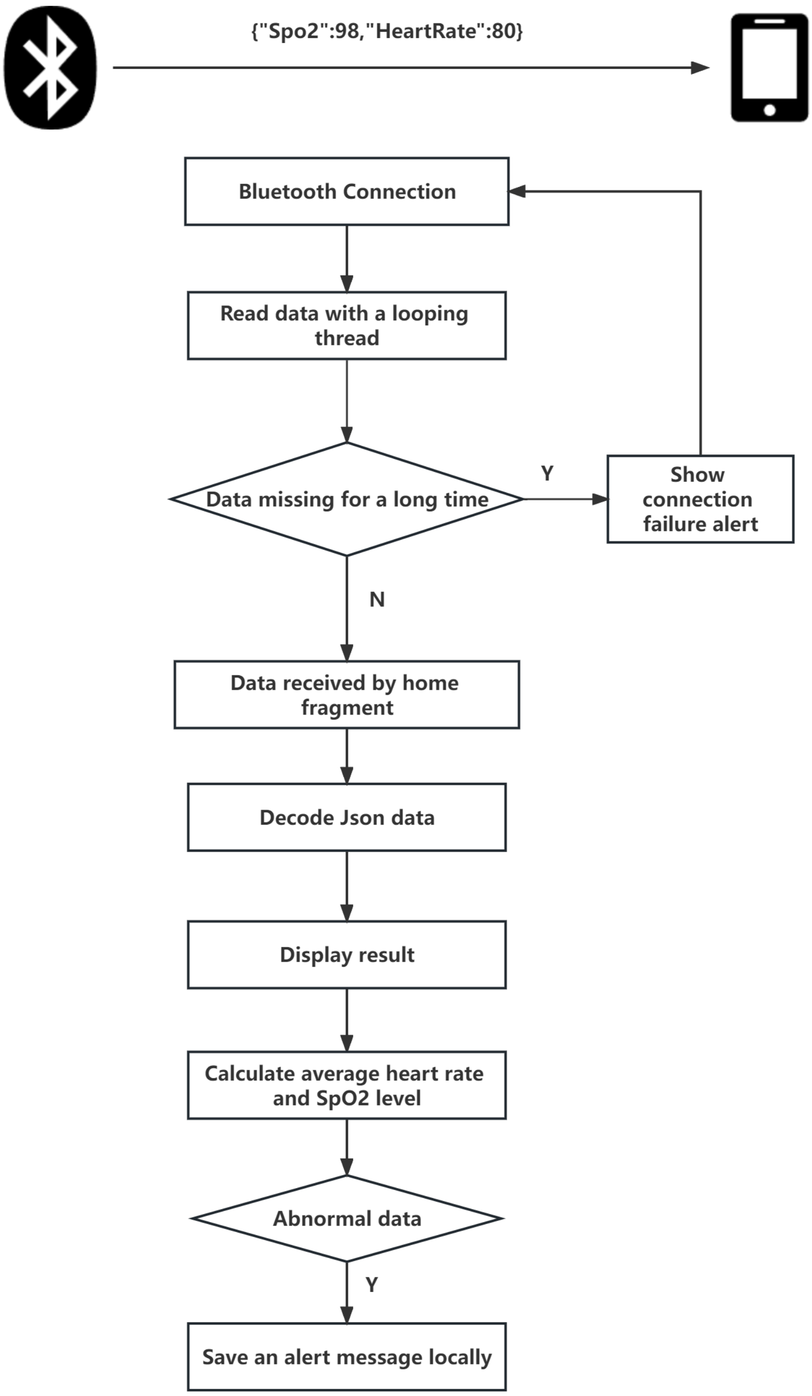

Figure 4.17: Block diagram for heart rate and SpO2 data

4. **Graph Management**: The system initializes an empty graph for data visualization. As new data arrives, it is added to the graph, which then redraws to include new information.

5. **Activity Prediction**: The system determines the user's activity by mapping the highest probability from the machine learning model's output to a corresponding label of a particular activity.

6. **Activity Destruction**: When the activity is terminated, the system stops the receiver and background thread and saves the recognized activities and their counts to the local storage.

In a nutshell, `ThingyRecognitionActivity` provides an interface for real-time monitoring and recognition of user's activities by harnessing motion data from the Thingy:52 sensor and machine learning techniques. It also includes features to visualize this data and provide feedback to the user. The block diagram of the HAR function is shown in Figure 4.18.

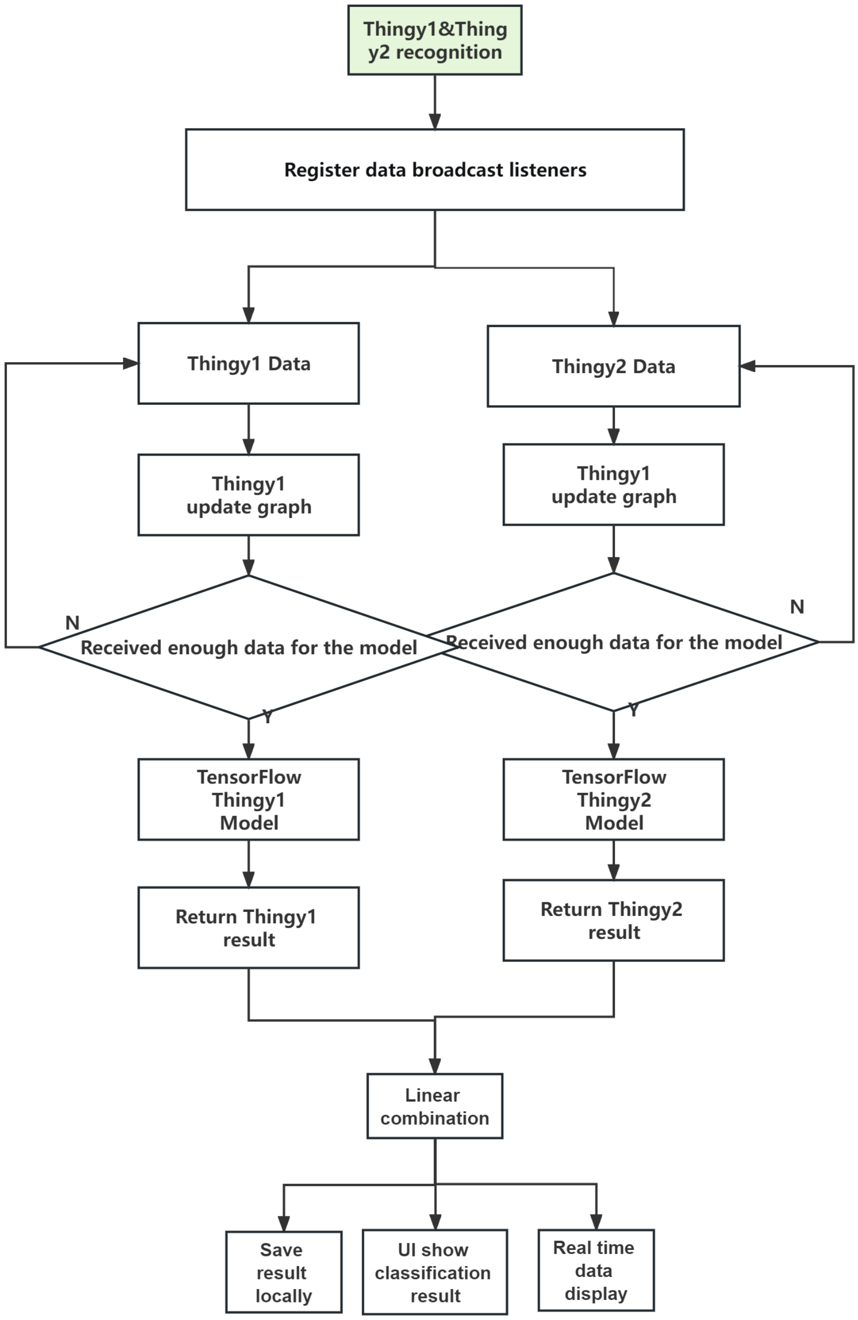

Figure 4.18: Block diagram for HAR

History

The `DataHistoryActivity` class is an Android `AppCompatActivity` that manages a screen for viewing historical activity data. This class provides a bar chart view of activity data for a specific date, which

includes categories such as sitting, lying down, standing, walking, running, ascending, and descending. This date can be changed using a `CalendarView`.
The important methods of the class are as follows:

1. **onCreate**: It initializes the activity's views and data.
2. **initBarChartStyle**: This method configures the style of the bar chart, including the settings for the X and Y axes, and provides the labeling function for the X-axis.
3. **initBarChartData**: This method populates the bar chart with the activity data for the selected day. It groups the data by activity types and calculates the sum for each group.
4. **initCalenderView**: This method initializes the `CalendarView` and sets up an `OnDateChangeListener` to refresh the bar chart when the selected date changes.

The historical data is stored in a local data structure, which includes variables such as activity type and timestamp. When reading historical data, the app first locates the timestamp based on the selected date and then displays the historical data in the form of a bar chart. The block diagram of history function is shown in Figure 4.19.

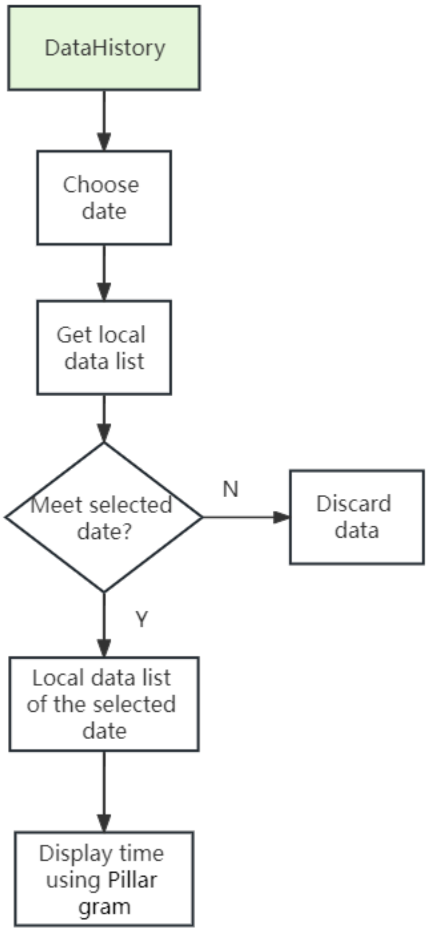

Figure 4.19: Block diagram for history function

Message Center

There are 2 key components in this part: the `MessageActivity` class which manages the display of alert messages, and the `AlertMsg` class, a custom data model used to store and manage the alert messages.
The `MessageActivity` class extends the `AppCompatActivity` class and is responsible for handling the display of health alert messages to the user. It retrieves the list of alert messages stored locally, uses an adapter to prepare the data for display, and binds the data to the user interface.
The `MessageActivity` class overrides the `onCreate()` method, where it retrieves a list of alert messages from local storage via the `MsgUtils.getMsgFromLocal()` method. It then initializes a `ListView` with this data. Each list item, represented by the `AlertMsg` data model, is inflated to a view layout and populated with the appropriate data. The item view includes visual indications for the type of alert and displays the time of the alert as well as the abnormal value (either SpO2 or heart rate) that triggered the alert.

The `AlertMsg` class is a custom data model that encapsulates health alert messages. Its key attributes are:

1. **timestamp**: a long type variable that stores the timestamp associated with the alert.
2. **msg**: a String that holds the content of the alert message.
3. **heartRateAverage**: a double representing the average heart rate.
4. **spo2Average**: a double representing the average SpO2 level.

This class provides getter and setter methods for all its attributes. The class includes a redefinition of the `toString()` method, which generates a string interpretation of the JSON `AlertMsg` object. In addition, it furnishes a `toJsonObject()` method, which produces a `JSONObject` instance representing the string `AlertMsg` object.

By integrating the `MessageActivity` class and the `AlertMsg` data model, the application effectively manages health alert messages, delivering essential information to the user in a user-friendly manner.

Chapter 5

Testing and Discussion

5.1 Testing

The testing phase is conducted to ensure the functionalities of all parts of the system.

5.1.1 Testing the MAX30102 reading

A comparative test was conducted to evaluate the accuracy of the Heart Rate and SpO2 level readings. This involved using two MAX30102 sensors from different manufacturers and deploying two separate libraries: the SparkFun MAX3010x Pulse and Proximity Sensor Library and the MAX3010x Sensor Library.

These sensors simultaneously recorded data from the same individual under identical conditions. Each library processed the data independently from each sensor, negating potential bias in data interpretation. The findings demonstrated a high level of correlation between the readings obtained from the two sensors. The consistency suggests that the MAX30102 sensor readings and the software libraries used to process these data were reliable.

This comparative test confirms the successful implementation and accuracy of the heart rate and SpO2 level monitoring feature.

Table 5.1: Comparison of Heart Rate and SpO2 Level from Different Sensors

	Company A		Company B	
	Heart Rate	SpO2 Level	Heart Rate	SpO2 Level
Sensor using SparkFun Library	75	97	76	98
Sensor using MAX3010x Library	76	98	75	97

	precision	recall	f1-score	support
0	0.92	0.95	0.94	259
1	1.00	0.98	0.99	261
2	0.99	0.98	0.98	281
3	0.96	0.94	0.95	273
4	1.00	1.00	1.00	269
5	1.00	0.99	1.00	259
6	0.99	1.00	1.00	287
7	1.00	0.99	0.99	278
8	0.97	0.93	0.95	272
9	1.00	1.00	1.00	258
10	0.90	0.96	0.93	216
11	0.92	0.97	0.94	234
12	0.94	0.93	0.94	273
13	0.91	0.87	0.89	256
accuracy			0.96	3676
macro avg	0.96	0.96	0.96	3676
weighted avg	0.97	0.96	0.96	3676

Figure 5.1: Accuracy for upper body model

	precision	recall	f1-score	support
0	0.81	0.94	0.87	268
1	0.87	0.90	0.89	269
2	0.91	0.78	0.84	259
3	1.00	0.99	1.00	276
4	0.96	0.99	0.97	250
5	0.96	0.99	0.97	278
6	0.99	0.92	0.95	265
7	0.95	0.95	0.95	274
8	0.97	0.97	0.97	257
9	1.00	0.98	0.99	258
10	0.95	0.97	0.96	229
11	0.85	0.93	0.89	217
12	0.91	0.88	0.90	257
13	0.91	0.85	0.88	288
accuracy			0.93	3645
macro avg	0.93	0.93	0.93	3645
weighted avg	0.93	0.93	0.93	3645

Figure 5.2: Accuracy for lower body model

5.1.2 Testing the ML model

A detailed approach to training and validation was employed to evaluate the accuracy of the machine-learning models. This was achieved by splitting the dataset into 80% and 20% subsets. The larger subset was utilised for training the models, while the smaller subset was used to validate the effectiveness of the training. The model's structure and parameters were iteratively optimised based on the validation results, resulting in optimal performance. To evaluate the accuracy of the machine learning models in a robust manner, a 5-fold leave-one-subject-out cross-validation (LOSOXV) strategy was adopted. This method provides an unbiased estimation of the model's performance on unseen data. It works by partitioning the data into five distinct subsets. The model is trained on four subsets, and the remaining subset is used for validation. This process is repeated five times, each time with a different validation subset. By applying LOSOXV, the model's ability to generalise to new, unseen data, a key consideration in machine learning, is measured more accurately.

This project mainly utilises two models: one for the upper body and another for the lower body. The results are represented in tabular forms and confusion matrixes. Figure5.1 and5.3 show the accuracy and confusion matrix of the upper model; Figure5.2 and5.4show the accuracy and confusion matrix of the lower model.

The upper body model achieved a commendable overall accuracy rate of 97%. Despite its overall high performance, some activities such as sitting straight, standing straight, walking, desk work, and general movement were only moderately accurate, with accuracy rates under 0.95. This can be attributed to the IMU sensor placed on the upper body, and the differences in these activities primarily involve the lower body's movements. Therefore, the recognition using the ML model from the upper body faced difficulties distinguishing these activities, leading to observed inaccuracies. The lower body model, on the other hand, had an overall accuracy of approximately 93%. Though the overall accuracy rate was lower compared to the upper body model, the lower body model excelled in recognising activities such as standing and walking, surpassing the upper body model's accuracy for these specific activities. This enhanced performance is due to the placement of this Thingy sensor in the trousers pocket, making it more sensitive to lower body movements and thus better at distinguishing activities involving the lower body.

These results demonstrated complementarity between the two sensors, suggesting their combined use could improve the system's overall accuracy. Consequently, a third hybrid model was developed, utilising both sensors. However, both models have their strengths in different classification tasks. To address this, the proportion of each activity was adjusted according to its observed accuracy rate. After several rounds of adjustments, the hybrid model's final accuracy and confusion matrix results are depicted in Figure. This iterative testing process and critical analysis of the model's implementation maximised its accuracy.

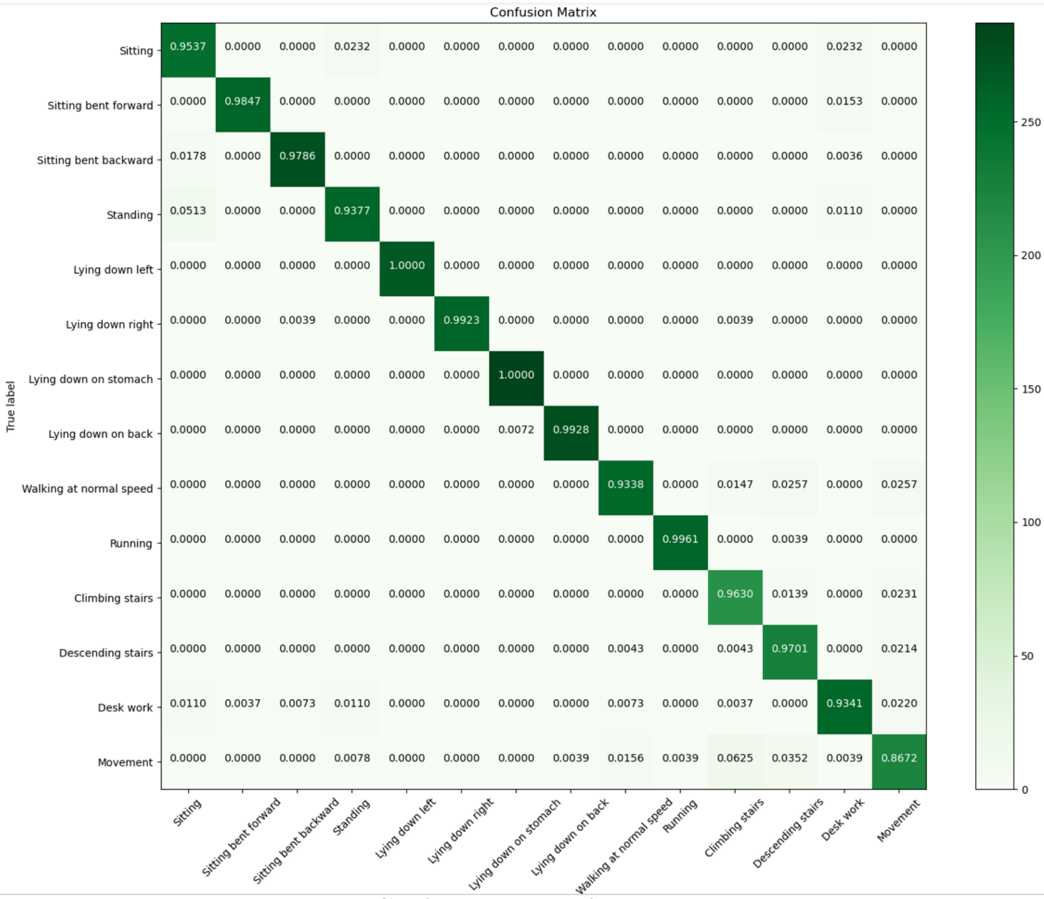

Figure 5.3: Confusion matrix for upper body model

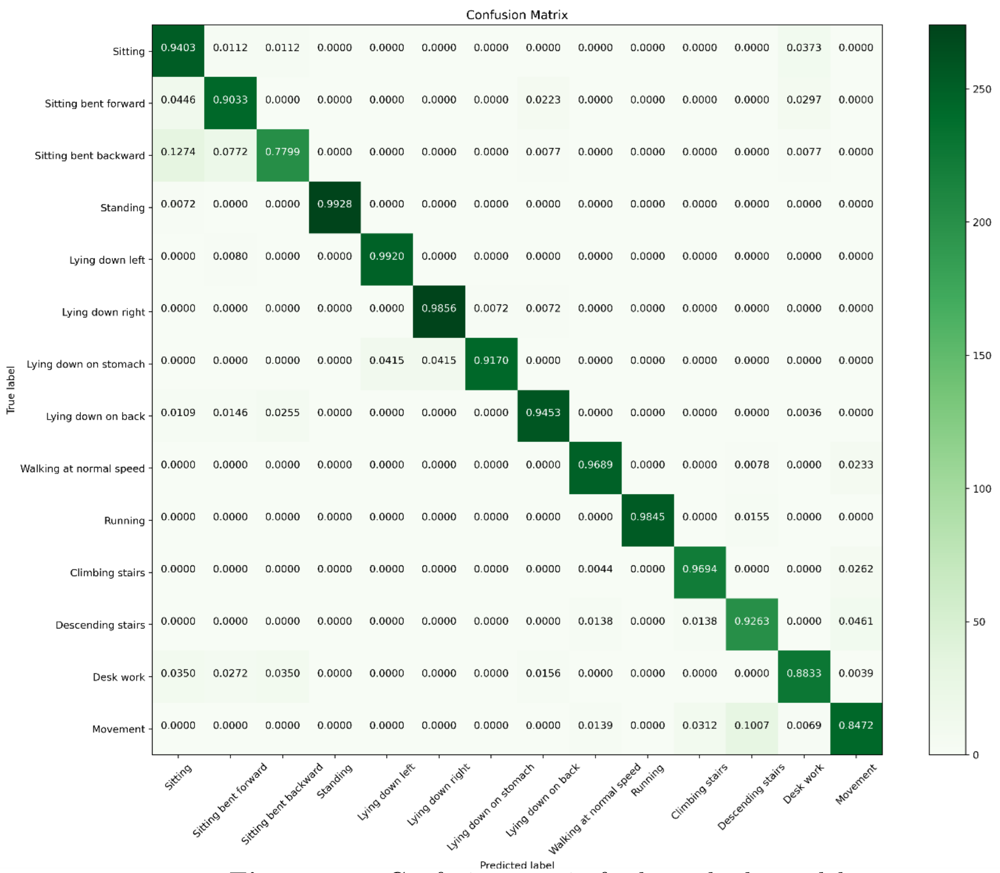

Figure 5.4: Confusion matrix for lower body model

5.1.3 Testing the recognition

The evaluation of real-time accuracy was a critical part of the testing process. This involved people engaging in various activities wearing the sensors, with the results tracked by the application.

For the upper body model, the results indicated a high level of accuracy for most activities. Certain activities, such as desk work, sitting bent forward, sitting bent backward, and ascending stairs, were correctly identified in most instances, with occasional miscategorisations. Standing and walking, however, were frequently misidentified. These results were consistent with expectations based on offline testing.

For the lower body model, it can successfully recognise activities such as standing and walking. It also works well recognising desk work and ascending stairs. However, the lower body model underperformed in recognising activities such as sitting bent forward and backwards, suggesting that these activities should primarily be identified by the upper body model.

These tests provided valuable data for refining the hybrid model. The relative contributions from each activity's upper and lower body results were adjusted based on these tests. The testing of the hybrid model demonstrated a significant improvement in recognition accuracy. With these results, the app was deemed to have met its design goals.

5.2 Discussion

5.2.1 Understanding Heart Rate, SpO2, and Their Interplay

Heart Rate: An Essential Life Sign

Heart rate refers to the number of heart contractions, or beats per minute (bpm). This measure varies considerably among individuals, influenced by age, fitness level, stress levels, medication use, and environmental conditions.

Tachycardia: High Heart Rate Tachycardia is defined by an abnormally high resting heart rate, typically exceeding 100 bpm in adults. While this can be normal during physical exertion or emotional stress, persistent high heart rate at rest may indicate conditions like anaemia, hyperthyroidism, or cardiovascular diseases.

Bradycardia: Low Heart Rate Conversely, Bradycardia is characterized by a resting heart rate below 60 bpm. This may be normal in highly fit individuals or athletes but can also signify potential health issues like heart blockages or hypothyroidism.

Blood Oxygen Saturation (SpO2): A Key Respiratory Indicator

SpO2 measures the proportion of oxygen-carrying haemoglobin in the blood relative to haemoglobin not carrying oxygen. It is an important gauge of how effectively the body is receiving oxygen.

Optimal and Low SpO2 Levels Normal SpO2 levels range between 95% and 100%, indicating efficient oxygen delivery. However, levels below 90% are considered low (hypoxemia), suggesting insufficient oxygen delivery to the body, potentially causing tissue damage and associated with conditions such as COPD and asthma.

The Interplay of Heart Rate and SpO2

Atypical Combinations and Health Implications The simultaneous occurrence of a high heart rate and high SpO2 is unusual and can suggest serious underlying issues. A high heart rate combined with low SpO2 might suggest severe health complications such as hypoxemia, prompting the body to compensate by increasing the heart rate. A low heart rate accompanied by high SpO2 might reflect a high level of physical fitness. Still, if the heart rate is abnormally low (bradycardia) with normal or high SpO2, it might indicate heart conduction issues. A low heart rate and low SpO2 can suggest serious health concerns like severe heart or respiratory conditions.[58]

In conclusion, while heart rate and SpO2 provide valuable insights into a person's health, they should be considered alongside other health indicators. They are components of a comprehensive health assessment and should not be used as standalone diagnostic tools.

5.2.2 The Combination of Physiological Data and Activity

The heart rate and SpO2 levels of an individual fluctuate dynamically in response to different activities. Examining these variations during states like sleep, routine daily activities, and vigorous exercise can offer insights into the body's potential health concerns.

During Sleep: A State of Rest and Restoration

In sleep, the body lowers its metabolic demands, resulting in a decreased heart rate typically ranging around 40-60 bpm for an average adult. However, significant deviations, like an abnormally low heart rate or irregular rhythm, could hint at sleep conditions like sleep apnea.
Regarding SpO2 levels during sleep, they ideally remain within the 95-100% normal range. Decreased SpO2 during sleep may also due to conditions like sleep apnea or other respiratory issues.

During Daily Activities: Metabolic Demand Increases

Regular daily activities like walking, cleaning, or working, the heart rate and SpO2 level generally stay within the normal range, which is 60-100 bpm and 95-100%, respectively.

During Running: High Intensity and Demand

Running, a high-intensity exercise, causes a significant increase in heart rate to cater to heightened oxygen and nutrient needs. The heart rate during running can escalate anywhere from 100 bpm to more than 200 bpm, based on the individual's fitness level, age, and exercise intensity.
During physical exercise, it is natural for a healthy individual to exhibit a slight decrease in SpO2 levels. This drop is attributable to the body's heightened demand for oxygen, driven by the increased energy output of the muscles engaged in the activity. Therefore, an observed SpO2 level ranging from 88% to 92% amid a workout routine does not generally warrant alarm as long as it is a transient phenomenon and recovers when the exercise ends.[59]

In summary, heart rate and SpO2 levels exhibit variations in tune with different activities. Significant deviations from these expected values might indicate potential underlying health issues.

5.2.3 Window Size in HAR

The process of selecting the appropriate window size for activity recognition was conducted through testing with three different window sizes: 4s, 2s, and 1.2s. The tests yielded similar validation accuracies across the board, suggesting that the model is robust across these window sizes. Specifically, a window size of 4s produced a validation accuracy of 96%, while reducing the window size to 2s slightly improved the accuracy to 97%. However, further reducing the window size to 1.2s led to a slight decrease in validation accuracy back to 96%.

Even though the differences in accuracies were marginal, the choice of window size can be a critical factor in real-time applications where response time is a concern. Larger windows might lead to a delayed activity recognition as more data is needed for prediction. On the other hand, smaller windows can potentially yield quicker responses but may come at the cost of lower accuracy.

The model trained with a 2s window was selected for the final implementation. This choice was determined to offer the optimal accuracy while maintaining a reasonable response time for the recognition tasks.

The following diagrams are the sample data and accuracy history of the 3 cases.

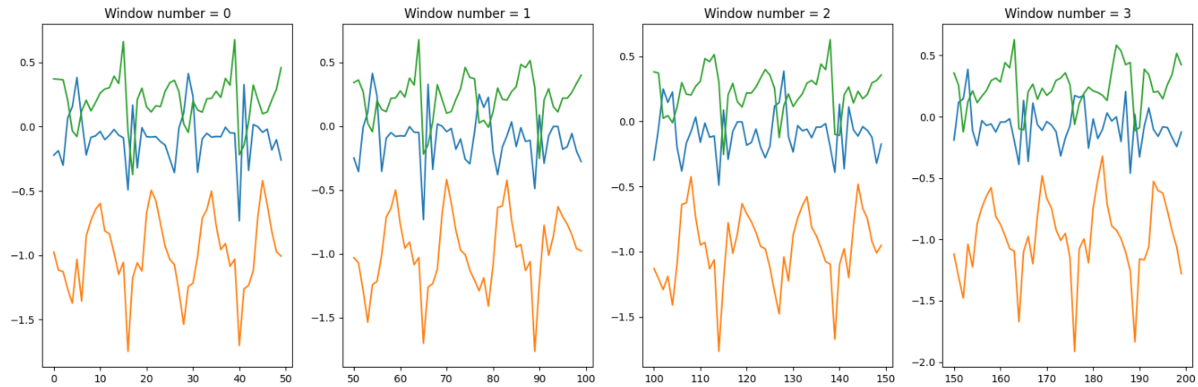

Figure 5.5: Sample data for 1.2s window

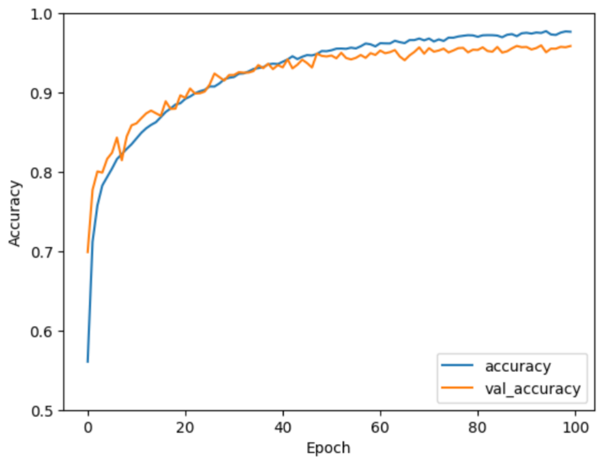

Figure 5.6: Accuracy history for 1.2s window

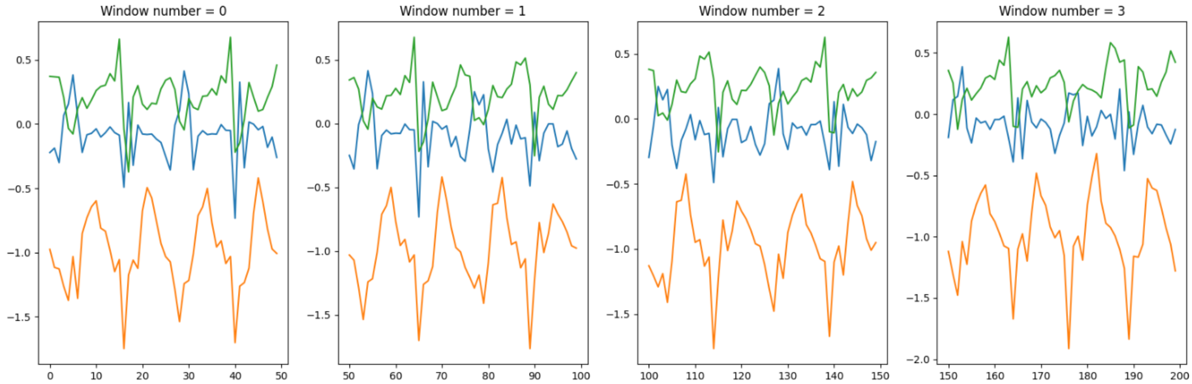

Figure 5.7: Sample data for 2s window

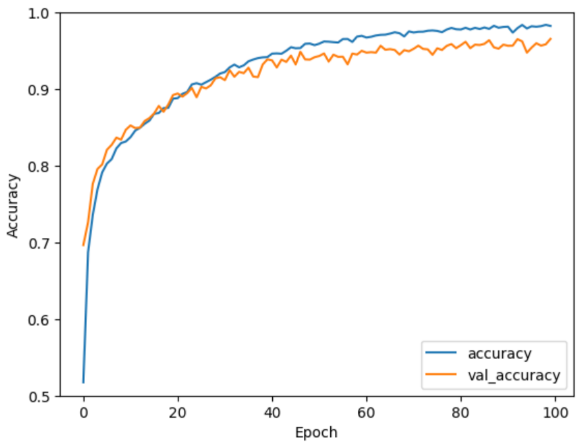

Figure 5.8: Accuracy history for 2s window

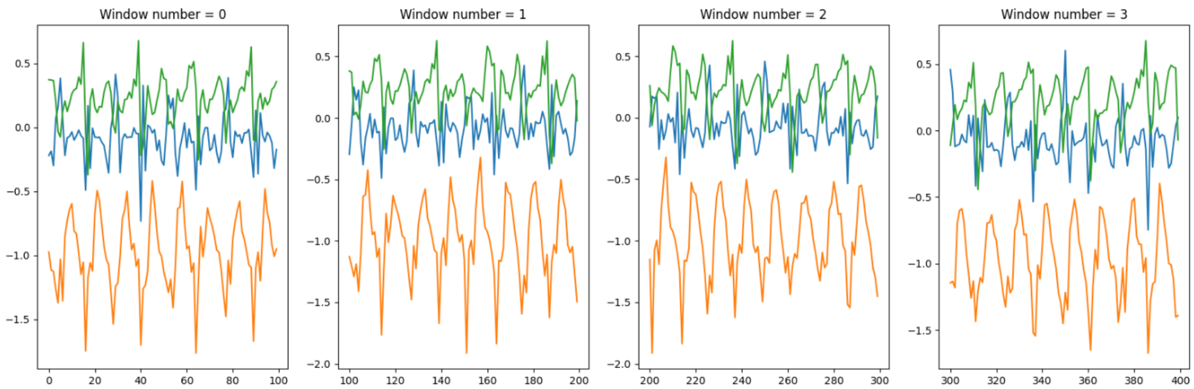

Figure 5.9: Sample data for 4s window

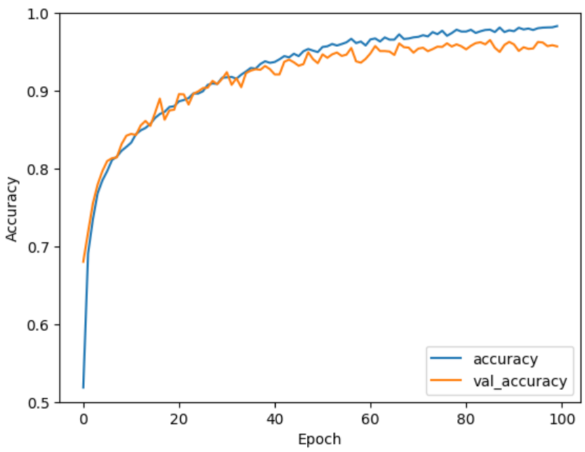

Figure 5.10: Accuracy history for 4s window

Chapter 6

Conclusion

This project completed the implementation of a health-monitoring system. The system serves as an accessible tool to track physiological parameters such as heart rate and SpO2 levels and to recognise user activities in real-time, providing individuals with enhanced control over their health management.

Firstly the system employs the Arduino development platform, the MAX30102 sensor, and the HC-06 Bluetooth module, with a meticulous configuration that guarantees reliable data acquisition. The data, including heart rate and SpO2 levels, is available to users via a user-friendly Android application. Simultaneously, the project deploys a Convolutional Neural Network (CNN)-based machine learning model that processes data from two Inertial Measurement Units (IMU) on the Nordic Thingy:52 kits to accurately classify user activities, providing users with activity records and enrich the user's understanding of their physical condition. This was achieved through an in-depth exploration of machine learning techniques. The Android application, developed with proficiency in Java and Kotlin, provides a platform for users to view their physiological parameters, activity data and receive personalised health advice. Communication between devices is realised by using Android Bluetooth and BLE functionalities. This project goes beyond mere data presentation by providing tailored health advice based on heart rate, SpO2 readings, and recognised activities, marking a stride towards innovative and personalised health management.

Despite its various strengths, the project also bears some limitations: While the MAX30102 sensor provides reliable measurements for heart rate and SpO2 levels, its accuracy can be affected by external factors such as user movement or incorrect finger placement, which might lead to erroneous readings, impacting the quality of the health advice provided. Then, although efficient in activity recognition, the machine learning model is currently limited to a predefined set of activities. Its performance may not be as accurate when encountering activities not included in the training data, reducing its applicability in real-life scenarios. Furthermore, the Android application might pose usability issues for individuals unfamiliar with smartphones or those with accessibility needs. The application currently only supports Android devices, limiting the user base to a specific subset of smartphone users. Lastly, the whole system runs in a local environment; if the user change to another smartphone, data will be lost.

There are several potential avenues for future research and development building on the existing system. More advanced sensor technology could improve the robustness of physiological data collection, and the integration of more diverse health data like body temperature or ECG could offer more comprehensive health monitoring. The machine learning model used for activity recognition could be trained on a broader range of activities, with the integration of user-specific models to bolster the personalisation and accuracy of the results. In terms of application development, expanding the application to other operating systems like iOS and Windows would increase its accessibility. Professional tools like Git and Docker may be applied during development for better version control. Cloud platforms like AWS from Amazon or AZURE from Microsoft can be applied, and databases like MongoDB or MySQL for scalability. One promising future application would be the development of predictive models, leveraging machine learning and AI to transition from a reactive to a proactive health management system. This highlights the system's potential as a powerful tool for personal health management and signifies exciting future advancements in the field of digital health.

Chapter 7

Phase 2 Planning

In the second phase of this project, the primary objectives will be to expand the existing system to cloud-based platforms and enhance its computational efficiency using GPU programming. This phase will allow the system to benefit from the scalability and flexibility of cloud computing and attempt to accelerate ML algorithms with GPUs' high-speed parallel computational power.

First, I will use Git and Docker to create a standardized development cycle and mitigate inconsistencies from varying local configurations. Git will serve as a version control system, while Docker will containerize the application for smooth deployment and testing without worrying about the local environment.

Next, I will move our system to a cloud platform like AWS, AZURE or Google Cloud, enabling remote data storage and computational capacity. This will be followed by integrating a database system like MongoDB or MySQL to store large-scale user data.

In addition, a user authentication system will also be developed to provide login, signup, and password recovery features.

The subsequent significant undertaking will be the acceleration of machine learning algorithms with CUDA (Compute Unified Device Architecture). By translating the activity recognition model into CUDA code, the model can exploit the parallel processing power of GPUs, thereby improving computational speed.

Week-by-week breakdown of the Phase 2 plan:

- **Week 1:** Setup version control using Git and create a Docker container for the application.
- **Week 2:** Develop user authentication system. This should include creating secure login, sign up, and password recovery functionalities.

- **Week 3-5:** Select the suitable cloud platform (such as AWS, Google Cloud, or Azure) and migrate the current system to this platform. Begin integration of a database system (like MongoDB or MySQL) to handle data storage and retrieval.
- **Week 6-9:** Learn CUDA programming and re-write the existing machine learning model for activity recognition using CUDA, allowing it to run on a GPU. Test the model to ensure it is working as expected.
- **Week 10:** Complete any unfinished tasks from previous weeks. Begin a thorough testing phase to identify and fix bugs.
- **Week 11-12:** Finish writing the dissertation.

Acknowledgements

I would like to express my deepest gratitude to my advisor, Dr Jiabin Jia, for his invaluable guidance, constant encouragement, and insightful suggestions throughout this project. His profound knowledge and expertise have immensely contributed to the successful completion of this work.

Huge appreciation goes to Dr Shiwei Wang, the project examiner, for taking the time to read this thesis.

Finally, I acknowledge the resources and facilities provided by the University of Edinburgh, which played a significant role in the realization of this project. I am immensely grateful for the learning experiences and the opportunities this institution has provided me during my study period.

References

[1] Organization, W. H.: *Ageing and health*, Accessed: 2023-07-26, 2023. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/ageing-and-health.

[2] Amazon: *Oxipro finger tip pulse oximeter*, Accessed: 2023-07-26, 2023. [Online]. Available: https://www.amazon.co.uk/OxiPro-Supplied-Saturation-Certified-Pulse-Oximeter/dp/B09NQPMRKQ/ref=sr_1_1_sspa?keywords=pulse+oximeter&qid=1690783392&sr=8-1-spons&sp_csd=d2lkZ2V0TmFtZT1zcF9hdGY&psc=1.

[3] Inc., A.: *Apple watch*, Accessed: 2023-07-26, 2023. [Online]. Available: https://www.apple.com/uk/watch/.

[4] Hassan, A. U., Salihu, Y. O. and Musa, M. J.: 'Real-time monitoring of heart beat rate and spo2 based on internet of things,' in *2021 International Conference on Electrical, Computer and Energy Technologies (ICECET)*, pp. 1–4. DOI: 10.1109/ICECET52533.2021.9698527.

[5] Pantelopoulos, A. and Bourbakis, N. G., 'A survey on wearable sensor-based systems for health monitoring and prognosis,' *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 40, no. 1, pp. 1–12, 2009.

[6] Banaee, H., Ahmed, M. U. and Loutfi, A., 'Data mining for wearable sensors in health monitoring systems: A review of recent trends and challenges,' *Sensors*, vol. 13, no. 12, pp. 17 472–17 500, 2013.

[7] Soh, P. J., Vandenbosch, G. A., Mercuri, M. and Schreurs, D. M.-P., 'Wearable wireless health monitoring: Current developments, challenges, and future trends,' *IEEE microwave magazine*, vol. 16, no. 4, pp. 55–70, 2015.

[8] Patel, S., Park, H., Bonato, P., Chan, L. and Rodgers, M., 'A review of wearable sensors and systems with application in rehabilitation,' *Journal of neuroengineering and rehabilitation*, vol. 9, no. 1, pp. 1–17, 2012.

[9] Custodio, V., Herrera, F. J., López, G. and Moreno, J. I., 'A review on architectures and communications technologies for wearable health-monitoring systems,' *Sensors*, vol. 12, no. 10, pp. 13 907–13 946, 2012.

[10] Piwek, L., Ellis, D. A., Andrews, S. and Joinson, A., 'The rise of consumer health wearables: Promises and barriers,' *PLoS medicine*, vol. 13, no. 2, e1001953, 2016.

[11] Liu, Y., Pharr, M. and Salvatore, G. A., 'Lab-on-skin: A review of flexible and stretchable electronics for wearable health monitoring,' *ACS nano*, vol. 11, no. 10, pp. 9614–9635, 2017.

[12] Lou, Z., Wang, L., Jiang, K., Wei, Z. and Shen, G., 'Reviews of wearable healthcare systems: Materials, devices and system integration,' *Materials Science and Engineering: R: Reports*, vol. 140, p. 100 523, 2020.

[13] Xu, H., Liu, J., Hu, H. and Zhang, Y., 'Wearable sensor-based human activity recognition method with multi-features extracted from hilbert-huang transform,' *Sensors*, vol. 16, no. 12, p. 2048, 2016.

[14] Paul, G. and Irvine, J.: 'Privacy implications of wearable health devices,' in *Proceedings of the 7th International Conference on Security of Information and Networks*, pp. 117–121.

[15] Elkefi, S. and Asan, O.: 'Wearable devices' use in geriatric care between patient-centeredness and psychology of patients,' in *Proceedings of the International Symposium on Human Factors and Ergonomics in Health Care*, SAGE Publications Sage CA: Los Angeles, CA, vol. 11, pp. 125–128.

[16] Haghi, M., Thurow, K. and Stoll, R., 'Wearable devices in medical internet of things: Scientific research and commercially available devices,' *Healthcare informatics research*, vol. 23, no. 1, pp. 4–15, 2017.

[17] Liu, S. and Han, J., 'Toward energy-efficient stochastic circuits using parallel sobol sequences,' *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 26, no. 7, pp. 1326–1339, 2018.

[18] Allen, J., 'Photoplethysmography and its application in clinical physiological measurement,' *Physiological measurement*, vol. 28, no. 3, R1, 2007.

[19] Scully, C. G., Lee, J., Meyer, J. *et al.*, 'Physiological parameter monitoring from optical recordings with a mobile phone,' *IEEE Transactions on Biomedical Engineering*, vol. 59, no. 2, pp. 303–306, 2011.

[20] Castaneda, D., Esparza, A., Ghamari, M., Soltanpur, C. and Nazeran, H., 'A review on wearable photoplethysmography sensors and their potential future applications in health care,' *International Journal of Biosensors & Bioelectronics*, vol. 4, no. 4, pp. 195–202, 2018. [Online]. Available: `https://doi.org/10.15406/ijbsbe.2018.04.00125`.

[21] Kim, K. B. and Baek, H. J., 'Photoplethysmography in wearable devices: A comprehensive review of technological advances, current challenges, and future directions,' *Electronics*, vol. 12, no. 13, p. 2923, 2023.

[22] Yan, Y.-S. and Zhang, Y.-T., 'An efficient motion-resistant method for wearable pulse oximeter,' *IEEE Transactions on information technology in biomedicine*, vol. 12, no. 3, pp. 399–405, 2008.

[23] Jovanov, E.: 'Preliminary analysis of the use of smartwatches for longitudinal health monitoring,' in *2015 37th annual international conference of the IEEE engineering in medicine and biology society (EMBC)*, IEEE, pp. 865–868.

[24] Choi, S.-S., Cha, S.-H., Tappert, C. C. *et al.*, 'A survey of binary similarity and distance measures,' *Journal of systemics, cybernetics and informatics*, vol. 8, no. 1, pp. 43–48, 2010.

[25] Jara, A. J., Zamora, M. A. and Skarmeta, A. F., 'An internet of things–based personal device for diabetes therapy management in ambient assisted living (aal),' *Personal and Ubiquitous Computing*, vol. 15, pp. 431–440, 2011.

[26] Poppe, R., 'A survey on vision-based human action recognition,' *Image and vision computing*, vol. 28, no. 6, pp. 976–990, 2010.

[27] Straczkiewicz, M., James, P. and Onnela, J.-P., 'A systematic review of smartphone-based human activity recognition methods for health research,' *npj Digital Medicine*, vol. 4, no. 1, pp. 1–12, 2021.

[28] Beddiar, D. R., Nini, B., Sabokrou, M. and Hadid, A., 'Vision-based human activity recognition: A survey,' *Multimedia Tools and Applications*, vol. 79, no. 41-42, pp. 30 509–30 555, 2020.

[29] Tran, D., Bourdev, L., Fergus, R., Torresani, L. and Paluri, M.: 'Learning spatiotemporal features with 3d convolutional networks,' in *Proceedings of the IEEE international conference on computer vision*, pp. 4489–4497.

[30] Hasler, B. S., Buecheler, T. and Pfeifer, R.: 'Collaborative work in 3d virtual environments: A research agenda and operational framework,' in *Online Communities and Social Computing: Third International Conference, OCSC 2009, Held as Part of HCI International 2009, San Diego, CA, USA, July 19-24, 2009. Proceedings 3*, Springer, pp. 23–32.

[31] Chen, C., Jafari, R. and Kehtarnavaz, N., 'A survey of depth and inertial sensor fusion for human action recognition,' *Multimedia Tools and Applications*, vol. 76, pp. 4405–4425, 2017.

[32] Aggarwal, J. K. and Xia, L., 'Human activity recognition from 3d data: A review,' *Pattern Recognition Letters*, vol. 48, pp. 70–80, 2014.

[33] Lara, O. D. and Labrador, M. A., 'A survey on human activity recognition using wearable sensors,' *IEEE communications surveys & tutorials*, vol. 15, no. 3, pp. 1192–1209, 2012.

[34] Shoaib, M., Scholten, H., Havinga, P. J. and Incel, O. D.: 'A hierarchical lazy smoking detection algorithm using smartwatch sensors,' in *2016 IEEE 18th International Conference on e-Health Networking, Applications and Services (Healthcom)*, IEEE, pp. 1–6.

[35] Bulling, A., Blanke, U. and Schiele, B., 'A tutorial on human activity recognition using body-worn inertial sensors,' *ACM Computing Surveys (CSUR)*, vol. 46, no. 3, pp. 1–33, 2014.

[36] Amft, O.: 'A wearable earpad sensor for chewing monitoring,' in *SENSORS, 2010 IEEE*, IEEE, pp. 222–227.

[37] Vrigkas, M., Nikou, C. and Kakadiaris, I. A., 'A review of human activity recognition methods,' *Frontiers in Robotics and AI*, vol. 2, p. 28, 2015.

[38] Minnen, D., Westeyn, T., Ashbrook, D., Presti, P. and Starner, T.: 'Recognizing soldier activities in the field,' in *4th International Workshop on Wearable and Implantable Body Sensor Networks (BSN 2007) March 26–28, 2007 RWTH Aachen University, Germany*, Springer, pp. 236–241.

[39] Wang, S., Chen, C. and Ma, J.: 'Accelerometer based transportation mode recognition on mobile phones,' in *2010 Asia-Pacific Conference on Wearable Computing Systems*, IEEE, pp. 44–46.

[40] Zhang, M. and Sawchuk, A. A.: 'Usc-had: A daily activity dataset for ubiquitous activity recognition using wearable sensors,' in *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, (Association for Computing Machinery, 2012), pp. 1036–1043, ISBN: 9781450312240. [Online]. Available: `https://doi.org/10.1145/2370216.2370438`.

[41] Sharma, S., Mohammadmoradi, H., Heydariaan, M. and Gnawali, O.: 'Device-free activity recognition using ultra-wideband radios,' in *2019 International Conference on Computing, Networking and Communications (ICNC)*, pp. 1029–1033. DOI: `10.1109/ICCNC.2019.8685504`.

[42] Wu, X., Chu, Z., Yang, P., Xiang, C., Zheng, X. and Huang, W., 'Tw-see: Human activity recognition through the wall with commodity wi-fi devices,' *IEEE Transactions on Vehicular Technology*, vol. 68, no. 1, pp. 306–319, 2019. DOI: `10.1109/TVT.2018.2878754`.

[43] Jalal, A., Kamal, S. and Kim, D.: 'Depth map-based human activity tracking and recognition using body joints features and self-organized map,' in *Fifth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*, pp. 1–6. DOI: `10.1109/ICCCNT.2014.6963013`.

[44] Wang, C., Zhang, J., Wang, Z. and Wang, J.: 'Position-independent activity recognition model for smartphone based on frequency domain algorithm,' in *Proceedings of 2013 3rd International Conference on Computer Science and Network Technology*, pp. 396–399. DOI: `10.1109/ICCSNT.2013.6967138`.

[45] Sadouk, L. and Gadi, T.: 'Convolutional neural networks for human activity recognition in time and frequency-domain,' in *Lecture Notes in Real-Time Intelligent Systems*, Mizera-Pietraszko, J., Pichappan, P. and Mohamed, L., Eds., (Springer International Publishing, 2019), pp. 485–496, ISBN: 978-3-319-91337-7.

[46] Ziaeefard, M. and Bergevin, R., 'Semantic human activity recognition: A literature review,' *Pattern Recognition*, vol. 48, no. 8, pp. 2329–2345, 2015, ISSN: 0031-3203. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S0031320315000953`.

[47] Wang, Y., Huang, K. and Tan, T.: 'Human activity recognition based on r transform,' in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8. DOI: `10.1109/CVPR.2007.383505`.

[48] Hammerla, N. Y., Halloran, S. and Plötz, T., 'Deep, convolutional, and recurrent models for human activity recognition using wearables,' *arXiv preprint arXiv:1604.08880*, 2016.

[49] Chen, Y. and Xue, Y.: 'A deep learning approach to human activity recognition based on single accelerometer,' in *2015 IEEE international conference on systems, man, and cybernetics*, IEEE, pp. 1488–1492.

[50] Khan, A. M., Tufail, A., Khattak, A. M. and Laine, T. H., 'Activity recognition on smartphones via sensor-fusion and kda-based svms,' *International Journal of Distributed Sensor Networks*, vol. 10, no. 5, p. 503 291, 2014.

[51] Zhao, S., Li, W. and Cao, J., 'A user-adaptive algorithm for activity recognition based on k-means clustering, local outlier factor, and multivariate gaussian distribution,' *Sensors*, vol. 18, no. 6, p. 1850, 2018.

[52] Qi, W., Su, H. and Aliverti, A., 'A smartphone-based adaptive recognition and real-time monitoring system for human activities,' *IEEE Transactions on Human-Machine Systems*, vol. 50, no. 5, pp. 414–423, 2020.

[53] Wang, J., Chen, Y., Hao, S., Peng, X. and Hu, L., 'Deep learning for sensor-based activity recognition: A survey,' *Pattern recognition letters*, vol. 119, pp. 3–11, 2019.

[54] Cho, H. and Yoon, S. M., 'Divide and conquer-based 1d cnn human activity recognition using test data sharpening,' *Sensors*, vol. 18, no. 4, p. 1055, 2018.

[55] Guan, Y. and Plötz, T., 'Ensembles of deep lstm learners for activity recognition using wearables,' *Proceedings of the ACM on interactive, mobile, wearable and ubiquitous technologies*, vol. 1, no. 2, pp. 1–28, 2017.

[56] Nikpour, B., Sinodinos, D. and Armanfard, N., 'Deep reinforcement learning in human activity recognition: A survey,' 2022.

[57] Chen, L. and Nugent, C., 'Ontology-based activity recognition in intelligent pervasive environments,' *International Journal of Web Information Systems*, vol. 5, no. 4, pp. 410–430, 2009.

[58] Nguyen, T.-N., 'Estimation of the relationship between ecg and spo2 signals of human,' Mar. 2020.

[59] Cables and Sensors: *Do spo2 levels fluctuate?* `https://www.cablesandsensors.com/pages/do-spo2-levels-fluctuate`, Accessed: July 31, 2023, 2022.

Appendix A

Evaluating the project using the SMART method

The SMART method was employed in this project to ensure clarity, focus, and feasibility. SMART, an acronym for Specific, Measurable, Achievable, Relevant, and Time-bound, is an effective tool for setting and accomplishing goals. It facilitates setting clear and concise objectives, establishing tangible measures for success, ensuring the feasibility of goals, aligning the project with broader relevance, and setting realistic timelines. Using the SMART framework in this project enabled precise planning, efficient execution, and effective monitoring of progress, thereby ensuring the successful realisation of the project's objectives within the stipulated timeline. The SMART method played an instrumental role in fostering transparency, accountability, and productivity throughout the project lifecycle.

A.1 SMART Objectives

A.1.1 Specific

The objective was to develop a health monitoring system capable of tracking the user's heart rate and SpO2 level and identifying the user's physical activity. The system would enable the user to manage their health effectively and act proactively if necessary.

A.1.2 Measurable

The effectiveness of the system can be gauged by its correctness in measuring heart rate and SpO2 levels, and its ability to classify physical activities correctly. The system is also expected to provide meaningful health advice based on collected data.

A.1.3 Achievable

The hardware used in this project, including the MAX30102 sensor and the Thingy:52, are technologically capable of providing the required measurements. The use of machine learning techniques for activity recognition is a proven method. The project utilised the Android Studio, Jupyter Notebook and Arduino IDE, powerful development tools that made the creation of the application feasible.

A.1.4 Relevant

With the increasing awareness of health and fitness, a portable and user-friendly health monitoring system is a timely and significant tool. It is particularly beneficial for individuals who require regular health monitoring, offering them an efficient and non-intrusive solution.

A.1.5 Time-bound

The project was structured in stages, with individual timelines for each stage, including system design, development, testing, and refinement. This made it possible to complete the project within the academic year, in line with the dissertation schedule.

Appendix B

Requirement Documentation

B.1 Requirement Specification

B.1.1 Purpose

- The Software Requirements Specification (SRS) is a blueprint for the system we aim to build, providing comprehensive details about the functional and non-functional requirements. The primary goal of this SRS is to define the expected system behaviour and performance to ensure that the system developed will meet the needs and expectations of its users while providing a robust, reliable, and user-friendly solution.

B.1.2 Scope

- Developing an Android application that can communicate with the peripheral sensors and the Arduino board via Bluetooth.
- Implementing data collection from the IMU sensors for accurate human activity recognition. This includes activities such as sitting, standing, lying down, walking, and falling.
- Integrating a SpO2 sensor to monitor blood oxygen levels and heart rate in real-time.
- Building data analysis and machine learning algorithms within the app to analyze and interpret the sensor data for various purposes including fitness tracking, health monitoring, and potential health issue detection.
- Ensuring that the system is user-friendly, reliable, and can operate both online and offline.
- Including functionalities such as real-time data display, data history visualization, and automatic reminders for long periods of inactivity.

B.1.3 Functional Requirements

1. **Data Acquisition:**
    - The system shall acquire data from the IMU and MAX30102 sensors in real-time.
    - The system shall accurately interpret sensor data for SpO2, heart rate, and physical activity.
2. **Data Analysis:**
    - The system shall analyze the acquired sensor data in real-time to detect potential health risks.
    - The system shall utilize machine learning models for accurate human activity recognition.
3. **User Interaction:**
    - The system shall provide a user-friendly interface to display sensor data and analysis results.
    - The system shall allow users to set up and configure the connected sensors.
4. **Alerts and Notifications:**
    - The system shall alert the user in case of abnormal sensor readings indicating potential health risks.
    - The system shall send reminders to users based on their physical activity, for instance, reminders to move after prolonged periods of inactivity.

B.1.4 Non-Functional Requirements

1. **Performance:**
    - The system shall ensure real-time processing and analysis of sensor data with minimal latency.
2. **Reliability:**
    - The system shall perform reliably under varying conditions and maintain a high accuracy of sensor data interpretation and human activity recognition.
3. **Usability:**
    - The system shall provide a clear, intuitive, and user-friendly interface that can be easily operated by non-technical users.
4. **Security and Privacy:**
    - The system shall ensure the privacy and security of user data.
5. **Portability:**
    - The system shall be compatible with Android devices, enabling users to monitor their health parameters and activities anywhere, anytime.

Appendix C

The Gantt chart of Phase 1

FigureC.1is the gantt chart that guided the development of phase 1.

MEng Project Phase 1

Smartphone Interface for Peripheral Sensors		**Organisation**	University of Edinburgh
Xingchen Chen	S2269664	**Start Date**	2023.1.16

No.	Task Name	Start Date	End Date	Percent Complete	Phase 1																
					First half of the semester					Second half of the semester						Summer vacation					
					Week1	Week2	Week3	Week4	Week5	FLW	Week6	Week7	Week8	Week9	Week10	Week11	Week12	Week13	Week14	Week15	Week16
1	**Preparatory Tasks**																				
1.1	Project Planning and Preparation	2023.1.16	2023.1.20	100%	■																
1.2	Set up development environment	2023.1.16	2023.1.20	100%	■																
1.3	Purchase hardware	2023.1.16	2023.1.20	100%	■																
2	**Arduino Part**																				
2.1	Basic reading of MAX30102	2023.1.16	2023.1.27	100%	■	■															
2.2	Basic communication of HC-06	2023.1.16	2023.1.27	100%	■	■															
2.3	Analyze the Arduino code and organise collected data	2023.1.23	2023.2.3	100%		■	■														
2.4	Make MAX30102 and HC-06 work together	2023.1.23	2023.2.3	100%		■	■														
2.4	Debug and calibrate the MAX30102 sensor	2023.2.13	2023.2.17	100%					■												
3	**Nordic Thingy part**																				
3.1	Connect the sensor with the phone	2023.2.27	2023.3.3	100%							■										
3.2	Calibrate the sensor and check the readings	2023.2.27	2023.3.3	100%							■										
4	**Android Development Part**																				
4.1	Learn Java basics	2023.1.30	2023.2.11	100%			■	■													
4.2	Learn Android development basics	2023.1.30	2023.2.11	100%			■	■													
4.3	Research into general bluetooth APPs	2023.1.30	2023.2.11	100%			■	■													
4.5	Research into Thingy-52 APP	2023.2.20	2023.3.3	100%						■	■										
4.6	User Interface Design	3023.3.13	2023.3.24	100%									■	■							
4.7	Implement Heart rate and SpO2 functions	2023.6.30	2023.7.10	100%												■	■				
4.8	Implement HAR functions	2023.7.02	2023.7.11	100%													■	■			
4.9	Integrate functions	2023.7.08	2023.7.20	100%														■	■		
4.11	Testing	2023.7.09	2023.7.21	100%														■	■		
5	**Machine Learning Part**																				
4.1	Implement simple ML model	2023.7.00	2023.7.09	100%													■	■			
4.2	Test various models and find the optimal	2023.7.01	2023.7.10	100%													■	■			
6	**Presenttions n reoprts**																				
6.1	Phse 1 presenttion	2023.3.6	2023.3.10	100%								■									
6.2	Phse 1 report	2023.6.29	2023.7.30	100%												■	■	■	■	■	■

Figure C.1: The Gantt chart of Phase 1