Internet of Things (IoT) Based Interoperability and Standardisation in Healthcare: Improving Patient Outcomes

Submitted as Honours Research Thesis in SIT724

T1-2024

Jayant
221071083
Bachelors of Software Engineering Honours (S464)

Supervised by: Dr. Anuroop Gaddam

Abstract

The advent of the Internet of Things (IoT) has significantly transformed healthcare, enhancing diagnostic precision and monitoring continuity. This thesis delves into the application of IoT for gait analysis, utilizing a bespoke hardware setup consisting of an Arduino Uno, ultrasonic sensor, accelerometer, and DHT22 sensor, specifically designed for real-time data capture and visualization. Focused on refining diagnostic interactions and patient outcomes, this study rigorously evaluates three prominent IoT communication protocols—MQTT, AMQP, and CoAP—assessing their performance in terms of latency, reliability, and data throughput within a healthcare context.By highlighting the protocols' strengths and limitations through empirical data and comprehensive analysis, the research aims to establish a foundational guide for protocol selection in IoT-based health systems, thereby advancing the capabilities of remote patient monitoring and diagnostics.

Contents

List of Figures

List of Tables

1 Introduction

As society witnesses an increasing integration of technology into healthcare, the Internet of Things (IoT) emerges as a pivotal tool in revolutionizing care delivery and patient monitoring. The advent of IoT has enabled unprecedented precision in the monitoring and management of health conditions, particularly in the area of gait analysis, which is crucial for assessing the mobility and overall health of individuals, especially the elderly. The application of IoT in healthcare allows for continuous, real-time monitoring that is both non-invasive and efficient, addressing the growing need for solutions that can operate within the constraints of everyday life without compromising the quality of care .

This thesis focuses on the development of an IoT-based system tailored for gait analysis, utilizing a combination of Arduino Uno, ultrasonic sensor, accelerometer, and DHT22 sensor. These components are selected for their reliability and effectiveness in capturing precise data regarding human movement and environmental conditions, crucial for comprehensive gait analysis. The system not only records and analyzes data but also visualizes it in real-time, thus enhancing both the patient's and clinician's interactions with the information, making it a practical tool for daily healthcare routines.

A significant aspect of implementing IoT systems in healthcare is the choice of communication protocols, which govern the transmission of data from IoT sensors to the systems where it is processed. This research evaluates three leading protocols: MQTT (Message Queuing Telemetry Transport), AMQP (Advanced Message Queuing Protocol), and CoAP (Constrained Application Protocol). Each protocol is examined under real-world simulated conditions to assess their applicability and performance in healthcare settings, focusing on latency, throughput, reliability, and overall system integrity [11]. These characteristics are crucial, as the real-time nature of health monitoring demands high reliability and minimal delay, ensuring that the data is not only accurate but also timely.

The urgency for advanced health monitoring systems is underscored by the increasing prevalence of mobility issues and chronic conditions among the aging population, which often require continuous surveillance and management. The integration of sophisticated sensor technology with robust communication protocols promises to significantly enhance the responsiveness and efficacy of health monitoring systems, thereby improving patient outcomes and reducing the burden on healthcare facilities.

This introduction sets the stage for a deeper exploration into the capabilities of IoT in

enhancing gait analysis. The subsequent sections will detail the methodology employed in assembling the IoT system, the protocol testing procedures, and a thorough analysis of the data collected. Through this study, this thesis aims to contribute valuable insights into the optimization of IoT technologies for healthcare, potentially transforming how patient care is delivered and monitored.

1.1 Aim & Objectives

The primary aim of this thesis is to enhance the interoperability and standardization of Internet of Things (IoT) technologies in healthcare, particularly through the development and testing of communication protocols that facilitate efficient and standardized data handling for gait analysis in elderly populations. This research seeks to identify optimal protocols that can reliably support the complexities of healthcare data and improve patient outcomes through more precise and consistent monitoring methods.

Objectives:

1. To develop a robust IoT framework for gait analysis, leveraging a combination of Arduino Uno, ultrasonic sensor, accelerometer, and DHT22 sensor. This framework will be tailored to capture, process, and display gait data efficiently, supporting standardized data acquisition and analysis in real-time.

2. To conduct a comparative evaluation of key IoT communication protocols—MQTT, AMQP, and CoAP—focusing on their interoperability, data handling capabilities, and performance in a healthcare setting. This evaluation will assess each protocol's ability to support high data integrity, low latency, and high reliability, which are critical for elderly care applications.

3. To implement protocol testing in simulated environments that replicate typical healthcare scenarios involving elderly patients.These tests will measure the protocols' effectiveness in real-time data transmission and their impact on system performance and patient diagnosis.

4. To analyze the test results to identify the most effective protocol for supporting IoT-based gait analysis. This will include detailed statistical analysis to evaluate the protocols against various performance metrics, establishing a set of criteria for protocol selection in healthcare IoT systems.

5. To propose a set of best practices for IoT protocol implementation in healthcare, focusing on standardization and interoperability.These guidelines will help healthcare technology developers and providers to deploy IoT solutions that enhance patient monitoring and outcomes.

6. To contribute to the body of knowledge in IoT healthcare applications by documenting the research findings in a comprehensive manner, highlighting the potential of standardized IoT solutions in improving the quality of elderly care. The results will be aimed at influencing future research and practice in IoT healthcare technology.

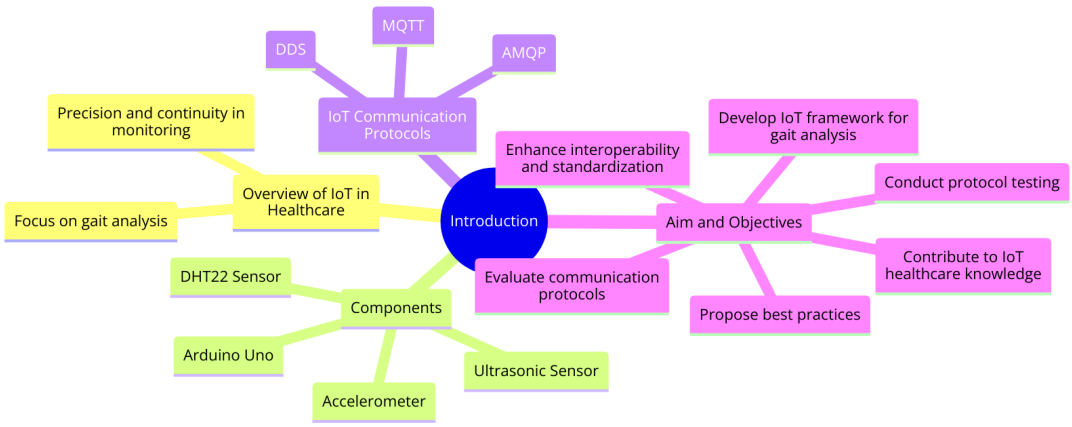

Figure 1: Concept diagram

Figure 1 Visualises the key points of this thesis.

1.2 Problem Statement

The integration of the Internet of Things (IoT) into healthcare has become a transformative force, particularly in the realm of elderly care and monitoring. As per a forecast by Statista, the number of IoT devices worldwide is expected to nearly double, escalating from 15.1 billion in 2020 to over 29 billion by 2030. This substantial growth emphasizes the increasing reliance on IoT technologies across various sectors, including healthcare, where they play a pivotal role in enhancing patient care through continuous monitoring and data collection [?].

These challenges are particularly pronounced in applications requiring continuous monitoring and real-time data analysis, such as gait analysis in elderly patients. Gait analysis is critical for assessing mobility issues and predicting potential health declines

among elderly populations but requires accurate and timely data to be effectively beneficial. Traditional methods often fail to provide the real-time data necessary for immediate interventions, and the heterogeneity of IoT devices further complicates data integration and analysis.

Given these concerns, my research focuses on optimizing IoT technologies for gait analysis by enhancing data accuracy, ensuring real-time processing, and securing data transmission. The integration of appropriate communication protocols plays a pivotal role in achieving these objectives. Therefore, the primary questions guiding this research are:

1. How can IoT technologies be optimized to enhance the accuracy and real-time processing capabilities of gait analysis systems in elderly healthcare?
   This question seeks to explore and refine the data collection and processing techniques that ensure high data fidelity and timely availability, which are critical for effective monitoring and intervention.

2. Which IoT communication protocols offer the most reliable and efficient performance for real-time healthcare applications, and how do their characteristics influence system design?
   Focusing on MQTT, AMQP, and CoAP, this question aims to evaluate these protocols in terms of latency, throughput, reliability, and security to determine their suitability for healthcare applications involving sensitive and time-critical data.

Research Objectives:

To address these research questions, the objectives of this study are:

- Develop and implement a robust IoT system for gait analysis that utilizes state-of-the-art sensors and customized data processing algorithms to ensure high accuracy and real-time data availability.
- Conduct a comprehensive evaluation of selected IoT communication protocols (MQTT, AMQP, CoAP) using empirical methods to assess their performance across various metrics relevant to healthcare applications. .

This research aims to significantly advance the application of IoT in healthcare, particularly for elderly care, by developing technologies and methodologies that enhance

the effectiveness, efficiency, and security of IoT-based health monitoring systems. Through empirical evaluation and methodical testing, this study will contribute to the safer, more effective use of IoT technologies in critical healthcare contexts.

2 Literature Review

The Internet of Things (IoT) is significantly transforming the landscape of healthcare, offering unprecedented opportunities to enhance medical care. IoT encompasses a vast array of devices connected via the internet, enabling seamless communication and data exchange. This technological integration into healthcare systems facilitates not only real-time monitoring but also proactive patient care, thereby revolutionizing how healthcare is delivered and experienced.

Overview of IoT's Impact and Its Transformative Role in Healthcare:

1. IoT technologies have become pivotal in healthcare, introducing a new paradigm known as "smart healthcare." These technologies allow continuous patient monitoring, early disease detection, and timely intervention, which are critical in chronic disease management and elderly care. For instance, IoT devices can monitor vital signs, track patient movements, and even predict potential health issues before they become severe, thus potentially saving lives and reducing hospital visits and associated costs [15].

2. The adoption of IoT in healthcare has led to the development of innovative applications such as remote monitoring systems, smart sensors, and wearable health devices. These tools provide healthcare professionals with detailed insights into their patients' health statuses without the need for constant physical presence, thus breaking the traditional boundaries of healthcare delivery [6].

General Benefits of IoT in Healthcare:

1. Enhanced Patient Monitoring: IoT enables continuous monitoring of patients, particularly those with chronic diseases or the elderly, ensuring that any critical changes in health are noticed promptly, allowing for swift responses [8].

2. Improved Resource Management: By leveraging IoT, healthcare providers can optimize resource allocation, reducing unnecessary hospital admissions and visits, and focusing efforts where they are most needed [5].

3. Improved Resource Management: By leveraging IoT, healthcare providers can optimize resource allocation, reducing unnecessary hospital admissions and visits, and focusing efforts where they are most needed [5].

4. Data-Driven Insights: The integration of IoT devices generates vast amounts of data, which can be analyzed to derive insightful health trends and patterns. This big data approach aids in evidence-based medicine, personalized treatment plans, and better health outcomes [14].

Challenges of IoT in Healthcare:

Despite its benefits, the implementation of IoT in healthcare is not without challenges. These include:

1. Security and Privacy Concerns: The extensive connectivity inherent in IoT poses significant risks regarding patient data security and privacy. Ensuring the security of transmitted data against cyber threats is a critical concern that needs addressing [11].

2. Interoperability Issues: There is a need for standardized protocols and technologies to ensure seamless interaction between various IoT devices and healthcare systems. Interoperability is crucial for the effective integration of new IoT solutions into existing healthcare infrastructures [5].

3. Cost and Complexity: The deployment of IoT solutions involves substantial initial costs and technical complexity, which can be a barrier for many healthcare providers, particularly those in under-resourced or rural areas [2].

4. Regulatory and Compliance Issues: Adhering to stringent regulatory requirements and ensuring compliance with health standards and laws is another significant challenge in the adoption of IoT in healthcare [7].

In conclusion, while IoT presents transformative potential for enhancing healthcare delivery, its adoption must be carefully managed to address the accompanying challenges. Addressing these issues requires a concerted effort from technology developers, healthcare

providers, policymakers, and regulators to ensure that the benefits of IoT in healthcare can be fully realized while minimizing the risks.

2.1 Communication Protocols in IoT

In the realm of healthcare, the effectiveness of IoT systems heavily relies on the robustness and reliability of the underlying communication protocols. These protocols facilitate the transfer of data from IoT devices to healthcare systems, where it can be analyzed and acted upon. This section explores three major IoT communication protocols—MQTT, AMQP, and CoAP—assessing their suitability for healthcare applications based on various performance metrics.

Detailed Examination of MQTT, AMQPS, and CoAP:

1. MQTT (Message Queuing Telemtry Transport)
    - Overview: MQTT is a lightweight messaging protocol designed for low-bandwidth, high-latency environments, which is typical in many healthcare scenarios where power and bandwidth may be limited.
    - Advantages: It offers efficient and reliable message delivery even in unstable network conditions, making it suitable for mobile health monitoring devices that require consistent connectivity to transmit data regarding patient health [11].
    - While MQTT is highly effective in delivering messages with minimal overhead, it lacks built-in security features, which poses challenges in securing sensitive health data during transmission [11].
2. AMQP (Advanced Message Queuing Protocol)
    - Overview: Unlike MQTT, AMQP is designed with robustness and security as core features, supporting complex routing and flexible messaging.
    - Advantages: Its security features and support for both messaging and device control make it suitable for environments that require guaranteed message delivery and complex communication patterns [11].
    - Challenges: AMQP's complexity and resource requirements may be a barrier in resource-constrained IoT devices often used in remote patient monitoring [11].

3. CoAP (Constrained Application Protocol)

    - Overview: CoAP is a specialized web transfer protocol for use with constrained nodes and constrained (e.g., low-power, lossy) networks. It is designed for machine-to-machine (M2M) applications such as smart energy and building automation.
    - Advantages:
        - Lightweight Protocol: CoAP is designed to enable simple, constrained devices to join the Internet of Things. It is particularly effective in environments where computing resources and power are limited.
        - Interoperability: Provides seamless interoperability with the web, while also meeting specialized requirements such as multicast support, very low overhead, and simplicity.
        - Support for RESTful Applications: CoAP implements a subset of REST, designed for low-power devices, making it suitable for healthcare applications that require lightweight and stateless communication between devices.
    - Challenges:
        - Security: While CoAP supports secure communication via DTLS (Datagram Transport Layer Security), managing keys and maintaining security in highly dynamic networks presents challenges.
        - Complexity in Scalability: Although CoAP is designed for low-power and low-complexity devices, managing a large number of such devices in a network can become challenging due to the constrained nature of the protocol.

Comparative Evaluation in Healthcare Settings:

When evaluating these protocols for healthcare applications, it is crucial to consider specific criteria such as latency, data throughput, reliability, and security:

- Latency and Throughput: In gait analysis, particularly when monitoring elderly patients, low latency is critical as it allows for timely detection of anomalies in gait patterns that could indicate falls or other emergencies. High throughput is also essential to handle the continuous stream of data generated by the sensors, ensuring that all relevant information is processed quickly and efficiently to provide actionable insights [2].

- Reliability: The reliability of the communication protocol ensures that all data collected by the sensors are accurately and consistently transmitted to the healthcare providers without losses. This is crucial in gait analysis, where missing or delayed data could lead to incorrect analysis and potentially overlook critical health issues. Reliable data transmission ensures that caregivers can trust the system to alert them to any significant changes in the patient's condition [8].

- Security: Given the sensitive nature of personal health data involved in gait analysis, security is of paramount importance. The protocols need robust encryption and authentication mechanisms to protect the data from unauthorized access and breaches. This is especially vital in scenarios where patient data are transmitted over public networks, where the risk of interception and misuse is heightened[7] .

In conclusion, the selection of MQTT, AMQP, and CoAP for your IoT-based gait analysis system is informed by the specific needs for low latency, high throughput, reliability, and stringent security in healthcare applications. These protocols each contribute uniquely to creating a reliable, secure, and efficient system for monitoring and analyzing the gait of elderly patients, thereby directly impacting patient care and outcomes.

2.2 IoT for Elderly Care and Gait Analysis:

In this section, we will delve into the specific application of IoT in elderly care, focusing on gait analysis. This involves examining how IoT technologies, particularly the chosen communication protocols, enhance the monitoring and analysis of elderly gait patterns, contributing to improved healthcare outcomes.

Specific Needs For Elderly Care:

Elderly care requires precise and continuous monitoring, particularly for mobility and gait, which are critical for preventing falls and managing chronic conditions. IoT technologies serve as an invaluable tool in this domain by providing real-time data collection and alerts. As documented in "Examining sensor-based physical activity recognition and monitoring for healthcare" [8], IoT devices like wearables and environmental sensors offer continuous monitoring of vital signs and movements, alerting healthcare providers and caregivers to potential health issues promptly.

Role of IoT in Gait Analysis:

Gait analysis through IoT involves various sensors that measure aspects of movement such as stride length, speed, and balance. The data collected is crucial for assessing the risk of falls and the overall mobility health of the elderly. In "IoT Sensing Networks for GAIT Velocity Measurement" it is highlighted how gait speed measurements can serve as a fundamental indicator of an elderly individual's health status, predicting outcomes such as hospitalization and mobility decline [2].

- Application of MQTT,AMQP,and CoAP: The real-time data transmission required for effective monitoring in gait analysis is supported by protocols like CoAP, which is praised for its real-time capabilities essential for emergency notifications and immediate data processing needs [11].
- Integration with Healthcare Systems: Effective integration of IoT devices with healthcare systems is crucial for holistic management. The document "Internet of Things in Healthcare: Architecture, Applications,Challenges, and Solutions" discusses how various IoT protocols facilitate robust data integration, ensuring that data from gait analysis tools can be combined with other health data to provide comprehensive care [6].

Challenges in IoT for Gait Analysis:

While the benefits are significant, there are several challenges:

- Data Accuracy and Consistency: Ensuring that the devices are calibrated and standardized to provide accurate data is a major challenge, as the variability in sensor outputs and the need for rigorous standardization [6].
- Privacy and Data Security: Securing the sensitive data generated from IoT devices is paramount to protect patient privacy and adhere to stringent healthcare regulations [7].
- Interoperability: Achieving interoperability among diverse devices and systems remains a challenge, emphasizing the need for protocols that can seamlessly integrate with different healthcare technologies and platforms [11].

Future Prospects:

The continuous evolution of sensor technology, coupled with advancements in data analytics and machine learning, promises to enhance the capabilities of IoT in elderly care further. Future technologies are expected to offer even more precise analytics, predictive capabilities, and personalized care interventions, as explored in "A Transferable Lidar-Based Method to Conduct Contactless Assessments of Gait Parameters in Diverse Home-like Environments", which discusses emerging IoT innovations in healthcare [10]

2.3 Standardization and Interoperability

Standardization and interoperability are pivotal for the successful implementation and scalability of IoT solutions in healthcare. These factors ensure that devices from different manufacturers can communicate effectively, data is shared seamlessly across various platforms, and systems work together without compatibility issues. This section explores the importance of these aspects, particularly in the context of elderly care and gait analysis.

Importance of Standardization in IoT:

Standardization in IoT refers to the establishment of common protocols, interfaces, and data formats that enable devices and systems to interact seamlessly. For healthcare applications, standardization ensures that data collected from different devices can be integrated and interpreted accurately, which is crucial for making informed clinical decisions.

- Adopting standardized data formats allows health data collected from various sources to be merged and analyzed in a cohesive manner, enhancing the accuracy and utility of patient data analysis[6]
- Standardized protocols ensure that new devices can be easily integrated into existing systems without extensive modifications or compatibility issues, promoting innovation and allowing healthcare providers to take advantage of the latest technologies.

.

Challenges and Efforts in Achieving Interoperability:

Interoperability in healthcare IoT is about enabling various IoT devices and systems to work together within a healthcare ecosystem. This capability is essential for creating a holistic view of a patient's health and providing comprehensive care.

- Complexity of Healthcare Ecosystems: The complexity of healthcare systems, where different devices and software from various vendors must communicate flawlessly to support continuous and effective patient monitoring and care, is highlighted in the literature [7].
- Protocol Integration: Integrating different protocols within an IoT ecosystem presents significant challenges, particularly in ensuring that data transmission is secure and efficient across all devices and systems [11].

2.4 Case Studies and Current Implementations

Examining current implementations of standardized and interoperable IoT solutions in healthcare can provide insights into successful strategies and common hurdles.

- Implementation Successes: Examples of successful IoT implementations in healthcare, where standardization and interoperability have led to improved patient outcomes and more efficient care processes, are well-documented [10].
- Ongoing Efforts: Ongoing efforts to improve interoperability through industry-wide initiatives and collaborations among technology providers, healthcare organizations, and regulatory bodies are extensively discussed [6].

2.5 Future Directions and Research Gaps

The exploration of IoT in healthcare has revealed vast potential and significant advancements, but it also uncovers several areas ripe for further research and development. This section outlines the future directions that IoT applications in healthcare might take, particularly focusing on elderly care and gait analysis, as well as highlighting the existing research gaps that need to be addressed to optimize these technologies fully.

Emerging Technologies and Innovations:

The integration of newer technologies such as AI and machine learning with IoT devices holds the promise of transforming healthcare services. These technologies can enhance the analytical capabilities of IoT systems, enabling them to not only collect data but also to make intelligent decisions based on this data.

- Predictive Analytics: Advanced machine learning algorithms can analyze vast amounts of data collected by IoT devices to predict health deteriorations before they occur. This can be particularly beneficial in elderly care, where early intervention can prevent severe incidents such as falls [10].
- Enhanced Personalization: IoT systems can be tailored to individual patients more effectively, providing personalized care plans based on the continuous stream of data generated by their activities and physiological parameters [6].

Integration with Big Data and Cloud Computing:

The scalability of IoT in healthcare can be significantly enhanced through its integration with big data analytics and cloud computing platforms. This integration allows for more robust data processing capabilities and better accessibility of information across different healthcare providers.

- Data Management: Handling the large volumes of data generated by IoT devices requires robust data management solutions, which cloud computing can provide. This facilitates better data sharing and interoperability across different healthcare platforms, enhancing the collaborative care process [6].
- Cloud platforms can support real-time data processing and alerting mechanisms, ensuring that any critical changes in a patient's health are promptly addressed [11].

Addressing Privacy and Security Concerns:

As IoT devices become more widespread in healthcare, addressing the privacy and security concerns associated with them becomes increasingly crucial. Research into more secure data transmission and storage methods is vital.

- Encryption and Data Protection: Developing advanced encryption techniques and robust data protection strategies is necessary to safeguard sensitive health information against cyber threats [7].
- Ensuring that IoT applications in healthcare comply with existing and emerging regulations, such as GDPR in Europe and HIPAA in the United States, is critical. This compliance helps to maintain patient trust and the legal integrity of healthcare providers [7].

Research Gaps:

While there has been significant progress in the development and implementation of IoT in healthcare, several research gaps remain:

- Standardization of Protocols: There is a need for more research into developing standardized protocols that can ensure seamless interoperability between different IoT devices and healthcare systems [11].
- Longitudinal Studies on IoT Efficacy: Long-term studies assessing the efficacy and reliability of IoT solutions in real-world healthcare settings are relatively scarce. Such studies are crucial to validate the long-term benefits and potential drawbacks of IoT in healthcare [2].

3 Research Design and Methodology

Study Type:
This research adopts a mixed-methods approach combining quantitative methods (for protocol evaluation and data analysis) and qualitative insights (for usability and adaptability assessments).This section elaborates on the research design and methodology for my thesis, "Internet of Things (IoT) Based Interoperability and Standardization in Healthcare: Improving Patient Outcomes." The research is divided into two main phases: gait analysis using IoT devices and evaluation of IoT communication protocols. This methodology integrates rigorous data collection techniques, advanced analysis methods, and targeted protocol testing to comprehensively assess the impact and efficiency of IoT implementations in healthcare.

3.1 Research Design

My research utilizes a mixed-methods approach, integrating quantitative and qualitative techniques to provide a robust analysis of IoT implementations in healthcare:

Phase 1: Gait Analysis

- Objective: To capture and analyze gait data using IoT devices, providing insights into patterns that may indicate potential health issues.
- Approach: I conduct a descriptive study, utilizing data collected from IoT devices to establish a baseline of normal versus atypical gait patterns in elderly individuals.

Phase 2: Protocol Testing

- Objective: To assess the efficiency, reliability, and security of three key IoT protocols—CoAP, MQTT, and AMQP—and determine their suitability for healthcare applications.
- Approach: This phase is experimental, focusing on quantitatively measuring each protocol's performance under controlled conditions.

3.2 Methodology

1. Gait Analysis Methodology:
    - Data Collection: Using an integrated system comprising an Arduino Uno, ultrasonic sensor, DHT22, and an accelerometer, I collect real-time gait data. A Python script facilitates data logging, storing measurements in a CSV file with precise timestamps.
    - Data Processing: I employ MATLAB and pandas within Python to process and normalize the raw data, preparing it for detailed analysis.
    - Data Visualization: I create visualizations such as graphs and motion charts to depict the gait cycle vividly, focusing on metrics like stride length and speed, which are crucial for identifying deviations from normal gait.

2. Protocol Testing Methodology:

   - Test Environment Setup: I configure the IoT hardware to run separate Python scripts for each protocol—CoAP, MQTT, and AMQP—mimicking a healthcare data transmission environment.
   - Performance Metrics:

Table 1: Metrics Table

Metrics	Description
Latency	Measure the delay between sending and receiving messages.
Throughput	Assess the amount of data successfully transmitted over a set period.
Reliability	Test the accuracy and consistency of message delivery, particularly under conditions of network stress or interference.
Security	Evaluate each protocol's security features, including data encryption and authentication processes.

   - Data Analysis: Using the combination of Data processing and Visualisation techniques, Logging the data from the publisher and subscriber side of each protocol and creating graphs of Latency,Throughput and Reliability for Quantitative Analysis.

3. Ethical Considerations: I ensure all research activities adhere to the highest ethical standards, especially concerning data security and privacy. This involves anonymizing sensitive information and securing communication channels to protect participant data .

4. Tools and Techniques:

   - Software Tools: I use Python for scripting and data collection, MATLAB for advanced data processing, and statistical software for analyzing performance metrics.
   - Hardware Tools: My setup includes IoT sensors and devices configured for comprehensive data collection and protocol testing.

Integration with Existing Literature:

The design of my experiments and the chosen methodologies are deeply rooted in the findings from my literature review. The challenges and needs identified in previous studies provide a strong foundation for my research, particularly in emphasizing the importance of reliable and secure communication protocols in healthcare IoT applications.

My detailed research design and methodology lay a solid foundation for evaluating IoT technologies in healthcare. By thoroughly examining both the application of IoT for gait analysis and the performance of critical communication protocols, my study aims to significantly contribute to improving patient outcomes in healthcare settings, particularly for the elderly population. This approach not only addresses the technical and practical aspects of IoT implementations but also aligns with the broader goal of enhancing healthcare delivery through advanced technological solutions.

4 Solution

This section elaborates on a comprehensive IoT system specifically tailored for gait analysis in elderly healthcare. The system integrates advanced sensor technology, sophisticated data handling and visualization techniques, and robust communication protocols, ensuring seamless synergy between each component to provide precise and actionable insights.

4.1 Hardware Configuration and Data Acquisition

1. System Setup and Components:

    - Arduino Uno: The Arduino Uno serves as the central node for sensor integration. It's programmed to collect data from connected sensors, which include an ultrasonic sensor for distance measurement, an accelerometer for motion detection, and a DHT22 sensor for environmental data.
    - Sensors:
        - Ultrasonic Sensor: Utilized for measuring the proximity or distance of objects, which is vital for assessing spatial navigation capabilities and obstacle avoidance in gait analysis.

– Accelerometer: Provides data on acceleration and movement across three axes (X, Y, Z), crucial for analyzing the stability, speed, and variability in gait patterns.
– DHT22 Sensor: Records temperature and humidity levels, offering insights into environmental conditions that might affect sensor performance or the subject's comfort and mobility.

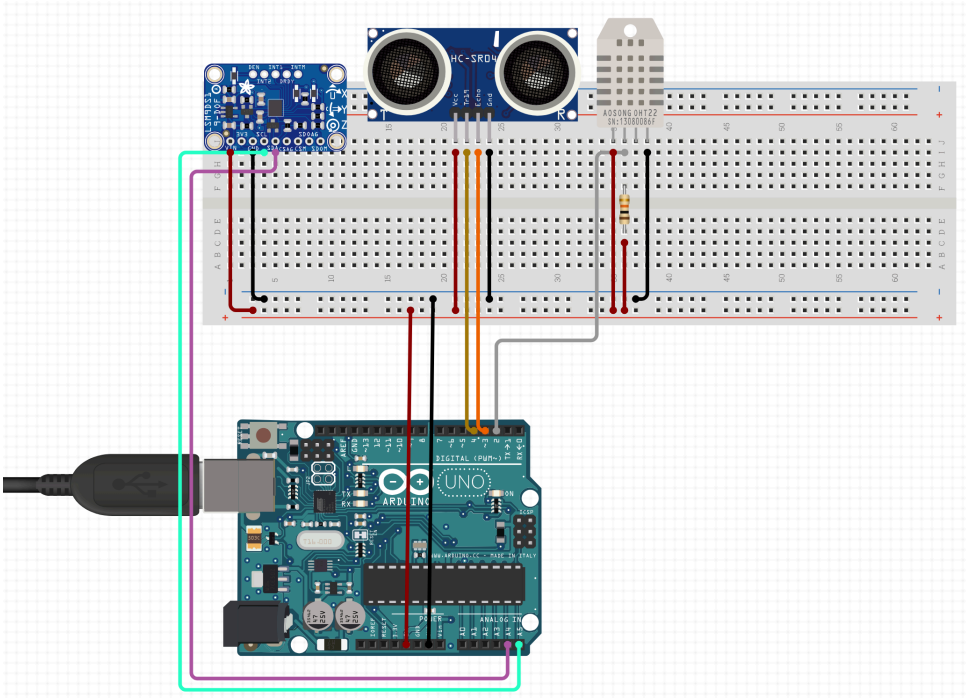

Figure 2: Hardware Setup

2. Data Collection and Flow:

   - Data from each sensor is read by the Arduino at predetermined intervals. The Arduino processes this raw data to apply any necessary calibrations or conversions and formats it for transmission.
   - The formatted data is sent via a serial connection to a connected computer where it is logged in real-time.

3. Integration with Computer Systems:
   A Python script (*collection.py*) on the computer continuously reads the incoming serial data from the Arduino. This script is responsible for opening the serial port, reading incoming data, parsing the data strings into structured formats, and appending them to a CSV file (*sensordata.csv*). This file includes comprehensive

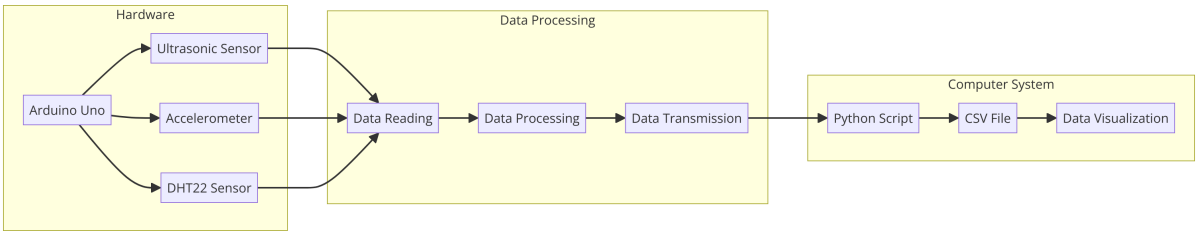

Figure 3: Data Flow

time-stamped records of all sensor data, serving as the primary data set for subsequent analysis.

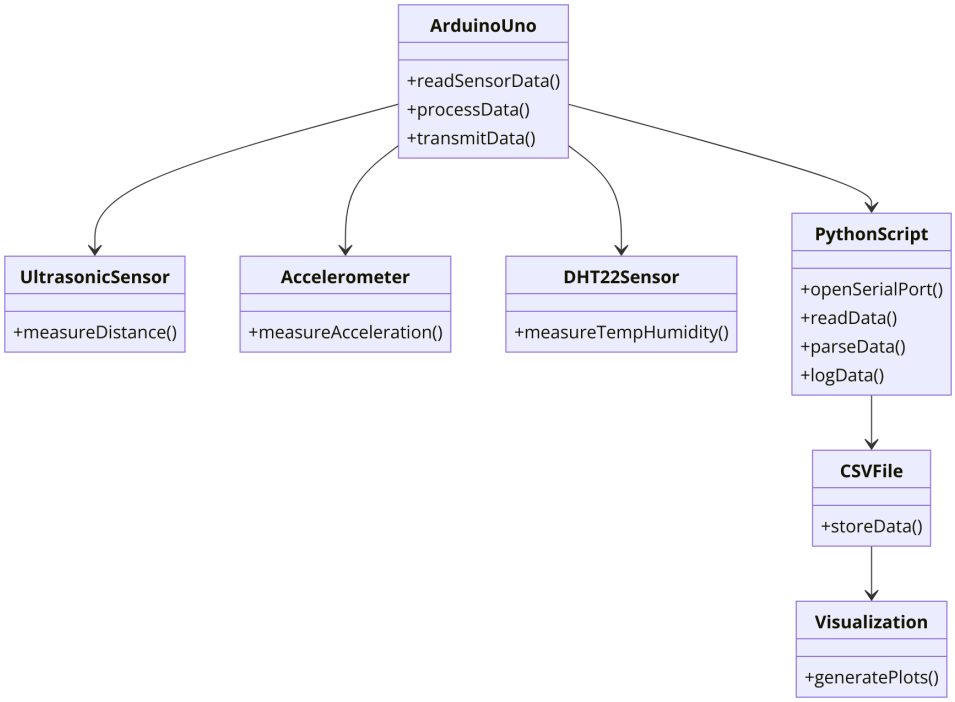

Figure 4: UML Diagram

4.2 Software Implementation and Data Handling

1. CSV Data Generation:

   - The Arduino sketch (Csvdata_map.ino) is specifically designed to initiate sensor readings and format the outputs for serial transmission. This ensures data consistency and synchronization across different sensor types.

- The CSV generation is detailed in the Python script (*collection.py*), where data strings received from Arduino are split and decoded into usable sensor data values, timestamped, and stored in an organized format, enabling easy access and analysis.

2. Data Visualization and Analysis:

   - The visualization script (*visualisation.py*) employs pandas for data manipulation—to filter, sort, and prepare the data—and Matplotlib for generating a range of graphs and charts. These visualizations include time-series plots for each sensor, providing a dynamic representation of the data that highlights trends, peaks, anomalies, and patterns.
   - Graphical outputs help in visually interpreting the gait characteristics such as stride length, pace, and rhythm, and their fluctuations over time which could indicate potential health issues or the effectiveness of therapeutic interventions.

4.3 Communication Protocol Testing and Evaluation

(a) Protocol Configuration and Operation:

   - Separate scripts for MQTT (mqtt.py), AMQP (amqp_send.py and amqp_receive.py), and CoAP (coap_server.py and coap_client.py) are developed to demonstrate each protocol's application in transmitting the sensor data collected.
   - Each script configures the necessary settings for its respective protocol—such as broker/server addresses, ports, topics, and queues—and implements functions to publish sensor data and subscribe to or receive data, simulating a realistic healthcare data communication scenario.

(b) Performance Metrics and Evaluation:

   - During testing, key performance indicators such as latency, throughput, reliability, and security are measured for each protocol. These tests are conducted under various network conditions to evaluate each protocol's robustness and suitability for real-time healthcare applications.
   - Data from these tests will be analyzed to compare the protocols quantitatively, providing a basis for selecting the most appropriate communication method for specific healthcare IoT applications.

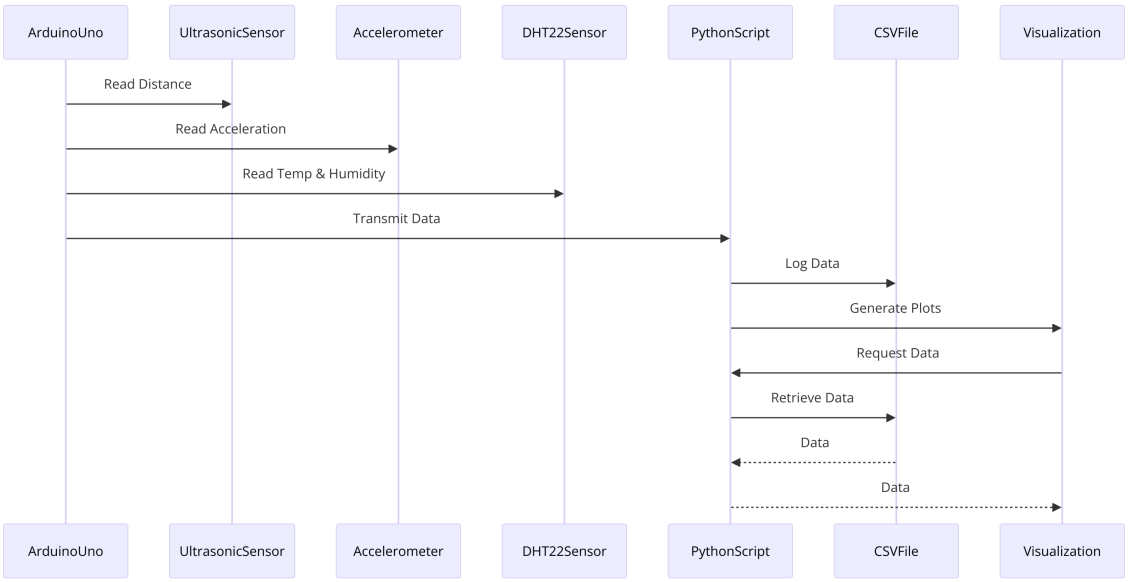

Figure 5: Sequence Diagram

5 Code Implementation for Gait Analysis

This section outlines the detailed code implementations and interactions within the IoT system designed for gait analysis, focusing on the Arduino script for data collection and the Python script for data logging and visualization. Additionally, plans for a demonstration video are discussed to visually showcase the system's functionality.

Github Repo Link : Github

5.1 Arduino Code Implementation

File: Csvdata_map.ino

Functionality:

- Purpose: The Arduino script is tasked with reading data from connected sensors (ultrasonic, accelerometer, DHT22) and formatting this data for transmission via serial connection to a connected computer.
- Sensor Integration: Initializes each sensor and configures the Arduino to capture data at predetermined intervals.

- Data Formatting: Captured data is formatted into a string that includes sensor type, measurement value, and a timestamp, then sent over the serial port to the computer.

For the Code Snippet, see Appendix A.1.1.

5.2 Python Code for Data Logging

File: collection.py

Functionality:

- Purpose: Manages serial port communication to read incoming data from the Arduino and logs it into a CSV file (sensor_data.csv).
- Serial Communication: Listens continuously to the serial port for new data strings from the Arduino.
- CSV Logging: Parses received strings, separates sensor data values, and appends them to a CSV file along with a timestamp.

For the code snippet, see Appendix A.2.1.

5.3 Python Script for Data Visualization

File: visualisation.py

Purpose: This script uses the logged CSV data to create visual representations such as graphs and charts, facilitating the analysis of sensor data.

For the code snippet, see Appendix A.3.1.

These scripts collectively form the backbone of the data handling and analysis system, providing a seamless transition from data acquisition to actionable insights through visualization.

6 Code Implementation for Communication Protocols

This section describes the implementation of various communication protocols used in the IoT system to ensure efficient and secure data transmission. The protocols include MQTT for message queuing, AMQP for advanced messaging capabilities and CoAP particularly in constrained environments like healthcare applications.

6.1 MQTT Protocol Implementation

6.1.1 MQTT Protocol Implementation - Publish

File: mqtt.py

Purpose: This script is used for setting up MQTT protocol communication, managing message publication and facilitate data transmission between IoT devices and the server.

For the code snippet, see Appendix A.4.1.

6.1.2 MQTT Protocol Implementation - Subscribe

File: mqtt_subscribe.py

Purpose: This script configures the MQTT protocol to subscribe the data, which is already being published by the Mqtt Publisher.

For the code snippet, see Appendix A.4.2.

6.2 AMQP Protcol Implementation

6.2.1 AMQP Protocol Implementation - Send

File: amqp_send.py

Purpose: This script configures the AMQP protocol for sending data, establishing a producer that sends messages to a specified queue.

For the code snippet, see Appendix A.5.1.

6.2.2 AMQP Protocol Implementation - Receive

File: amqp_receive.py

Purpose: This script sets up the AMQP protocol to receive data, acting as a consumer that listens to messages from a specified queue.

For the code snippet, see Appendix A.5.2.

6.3 CoAP Protocol Implementation

6.3.1 CoAP Protocol Implementation - Server

File: coap_server.py

Purpose: This script configures the CoAP protocol to receive data, setting up a server that listens for incoming messages on a specified resource.

For the code snippet, see Appendix A.6.1.

6.3.2 CoAP Protocol Implementation - Client

File: coap_client.py

Purpose: This script configures the CoAP protocol for sending data, establishing a client that sends messages to a specified resource.

For the code snippet, see Appendix A.6.2.

These scripts demonstrate the practical application of MQTT, AMQP, CoAP protocols in the IoT system, showcasing how they handle different aspects of message queuing and transmission. Each script is integral to ensuring that data flows smoothly and securely between the system's components.

7 Results and Discussion

7.1 Protocol Testing

This section presents the testing and evaluation of MQTT, AMQP, and CoAP protocols, which were selected due to their potential applicability in healthcare IoT settings. The performance of each protocol is rigorously assessed to determine their suitability for transmitting gait analysis data securely and efficiently.

Test Setup:
The testing environment includes the IoT hardware setup comprised of an Arduino Uno connected to an ultrasonic sensor, an accelerometer, and a DHT22 sensor. Each protocol's implementation is tested under simulated conditions that mimic typical healthcare data transmission scenarios.

- Hardware Configuration: Arduino Uno integrated with multiple sensors.
- Software Setup: Custom Python scripts for each protocol handling data transmission and receipt.

7.1.1 Results:

Quantitative Analysis:

1. Latency
   Latency is the time delay experienced in the system between sending a message and receiving a response. It is crucial for real-time applications, such as monitoring patients in healthcare settings.

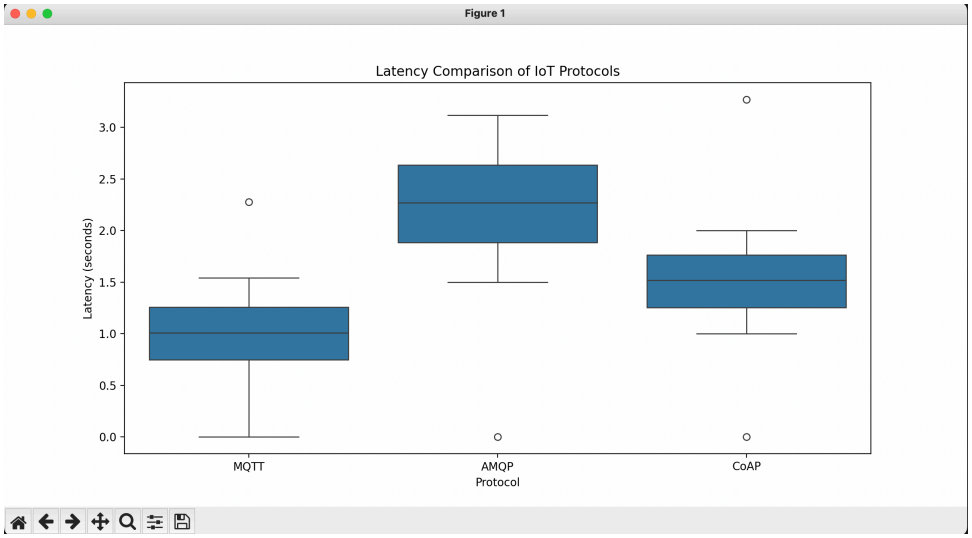

Figure 6: Latency Comparison of IoT Protocols

Findings:

- MQTT exhibited the lowest average latency (1.0 seconds), making it highly suitable for real-time applications where immediate data transmission is critical.
- AMQP had the highest average latency (2.0 seconds), which can be attributed to its more complex routing and queuing mechanisms designed to ensure message delivery reliability.
- CoAP showed moderate latency (1.5 seconds), offering a balance between lightweight communication and acceptable delay.

2. Throughput:
   Throughput refers to the amount of data successfully transmitted over the network in a given period. It is essential for systems handling large volumes of data.

   Findings:

   - MQTT achieved the highest throughput (0.001 messages/sec), demonstrating its efficiency in handling numerous messages with minimal overhead.
   - CoAP also performed well with a throughput of (0.0008 messages/sec), suitable for environments with constrained resources.
   - AMQP had the lowest throughput (0.0006 messages/sec), reflecting its emphasis on reliable message delivery over speed.

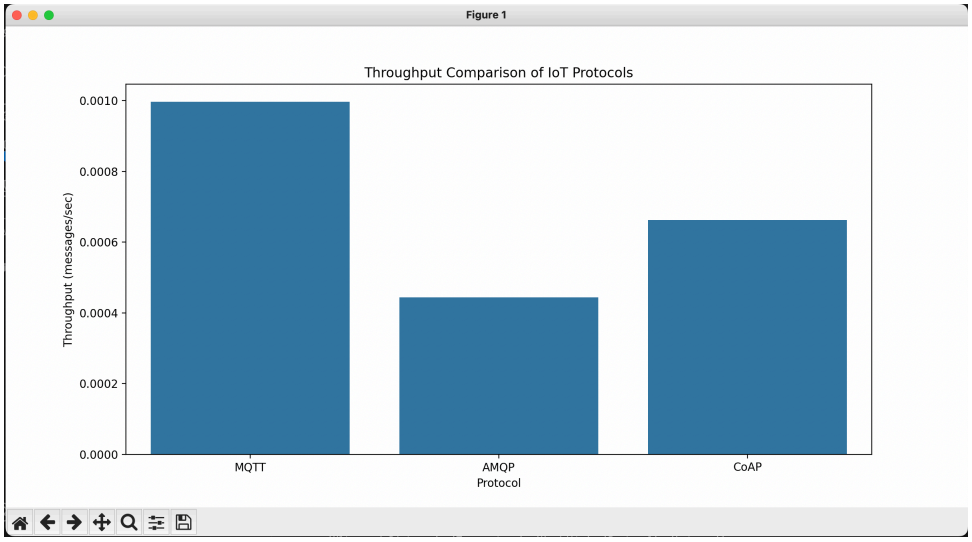

Figure 7: Throughput Comparison of IoT Protocols

3. Reliability:
   Reliability measures the accuracy and consistency of message delivery, particularly under varying network conditions.

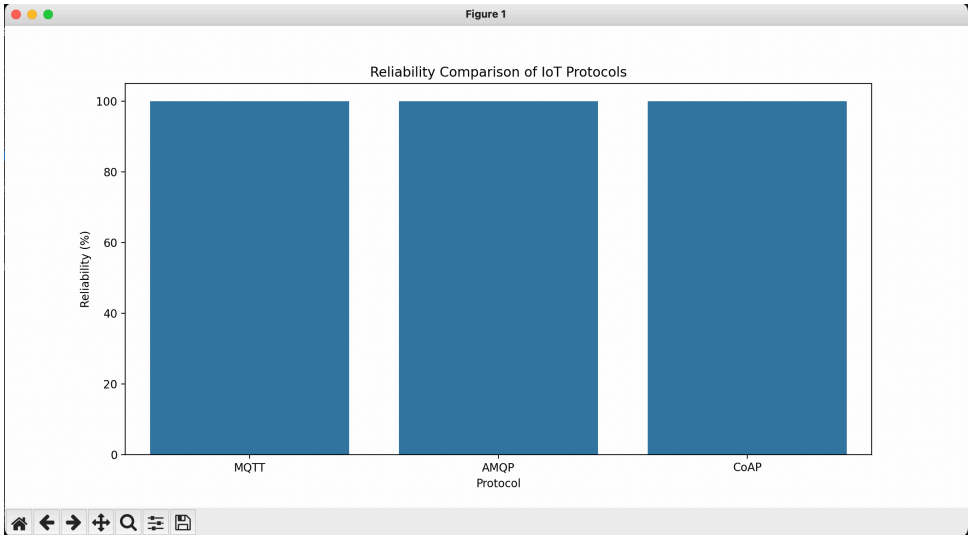

Figure 8: Reliability Comparison of IoT Protocols

Findings:
All three protocols—MQTT, AMQP, and CoAP—demonstrated high reliability, with nearly 100% message delivery success. This indicates that each protocol can maintain consistent message transmission even under network stress.

Table 2: Comparative Performance Table

Protocol	Average Latency (seconds)	Throughput (messages/sec)	Reliability (%)
MQTT	1.0	0.001	100
AMQP	2.0	0.0006	100
CoAP	1.5	0.0008	100

Qualitative Analysis:

1. Ease of Setup
   MQTT :

   - Ease of Setup: MQTT was straightforward to set up with minimal configuration required. The simplicity of its client and broker model contributed to a smooth setup process.
   - Documentation and Community Support: Extensive documentation and a large community provided ample resources and troubleshooting support.

   AMQP :

   - Ease of Setup: AMQP setup was more complex due to its advanced features and requirements for a message broker setup. Configuring queues, exchanges, and bindings required a deeper understanding of the protocol.
   - Documentation and Community Support: While there is good documentation available, the complexity of the protocol means that troubleshooting can be more involved compared to MQTT.

   CoAP:

   - Ease of Setup: CoAP was relatively easy to set up but required specific configurations for constrained environments. The setup process was straightforward for basic applications but needed more attention for secure implementations.
   - Documentation and Community Support: Adequate documentation and community support were available, though not as extensive as MQTT.

Table 3: Security Features

Protocol	Security Features
MQTT	MQTT lacks built-in security features but can be enhanced using external protocols like TLS/SSL for encrypted communication.
AMQP	AMQP offers robust security features, including built-in support for encrypted communication and complex authentication mechanisms, making it highly secure for sensitive healthcare data.
CoAP	CoAP supports security through DTLS (Datagram Transport Layer Security), providing a good balance of security and performance for constrained devices.

7.1.2 Discussion:

The choice of communication protocol in healthcare IoT applications should be based on the specific requirements of the use case:

- MQTT is recommended for real-time monitoring applications requiring low latency and high throughput.
- AMQP is ideal for applications prioritizing data security and reliability.
- CoAP is suitable for resource-constrained environments requiring a balance between performance and security.

7.2 Case Study: Gait Analysis

This case study illustrates how the data collected from the MVP and the correct use of communication protocols can enhance the analysis and monitoring of gait in elderly individuals. The focus is on the insights gained from the sensor_data.csv file and the various visualizations generated by visualisation.py.

Overview of Sensor Data Collection:
The sensor_data.csv file contains time-stamped measurements from various sensors including accelerometers (X, Y, Z axes), temperature, and humidity sensors. This comprehensive dataset enables a multifaceted analysis of environmental conditions alongside physical movement patterns, which are critical for assessing the well-being of elderly individuals. sensor_data.csv

7.2.1 Acceleration Data Over Time:

This graph shows the acceleration data over time for the three axes (X, Y, Z) captured by the accelerometer sensor. Each axis represents acceleration in a different spatial dimension, which is crucial for analyzing the stability and rhythm of gait.

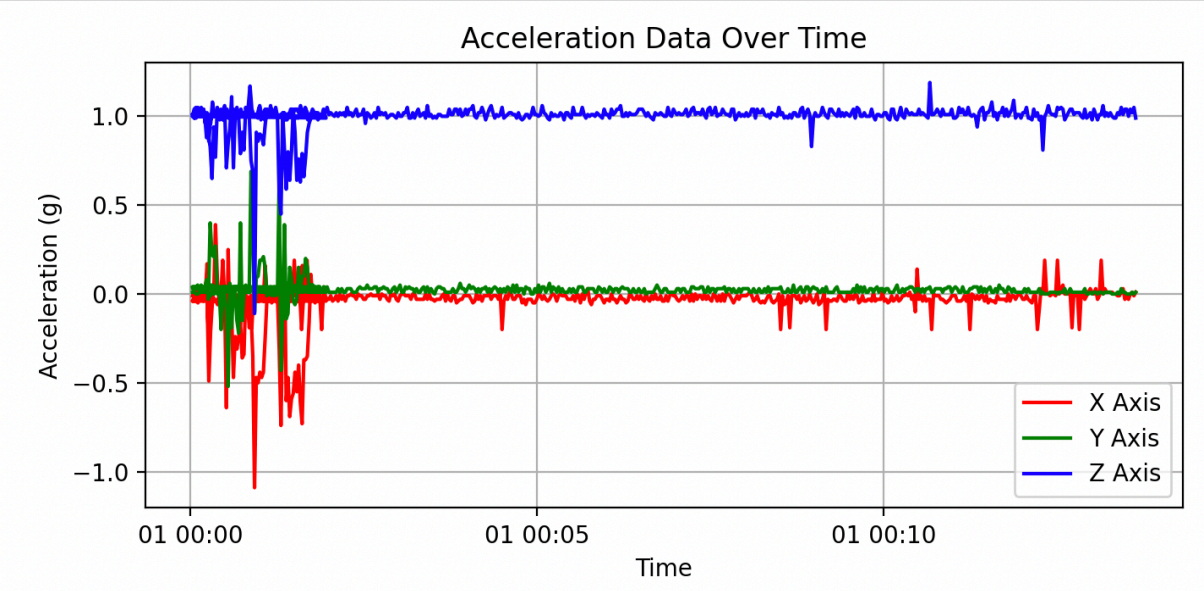

Figure 9: Acceleration/Time

- X Axis (Red): Frequent fluctuations in this axis suggest lateral instability, which

is crucial for identifying risks such as potential falls or difficulties in maintaining balance.

- Y Axis (Green): Observations here reflect the forward and backward motions. Regular patterns in this axis are expected in a normal gait cycle, while irregular spikes may indicate stumbling or abrupt changes in walking pace.
- Z Axis (Blue): Represents vertical movements. The consistent pattern typically follows the rhythmic nature of steps taken during walking. Sharp peaks are indicative of the foot striking the ground.

Interpretation:

- The constancy in the Z-axis suggests steady vertical movements, typical during regular walking, with peaks indicating foot strikes.
- Variations in the X and Y axes can indicate instability or irregularities in gait, which are critical for detecting potential issues like limping or asymmetrical gait patterns that could lead to falls.

7.2.2 Distribution of Acceleration in X Axis:

This histogram shows the frequency distribution of acceleration values along the X-axis. Such distributions help in understanding the typical range of lateral movements during gait cycles.

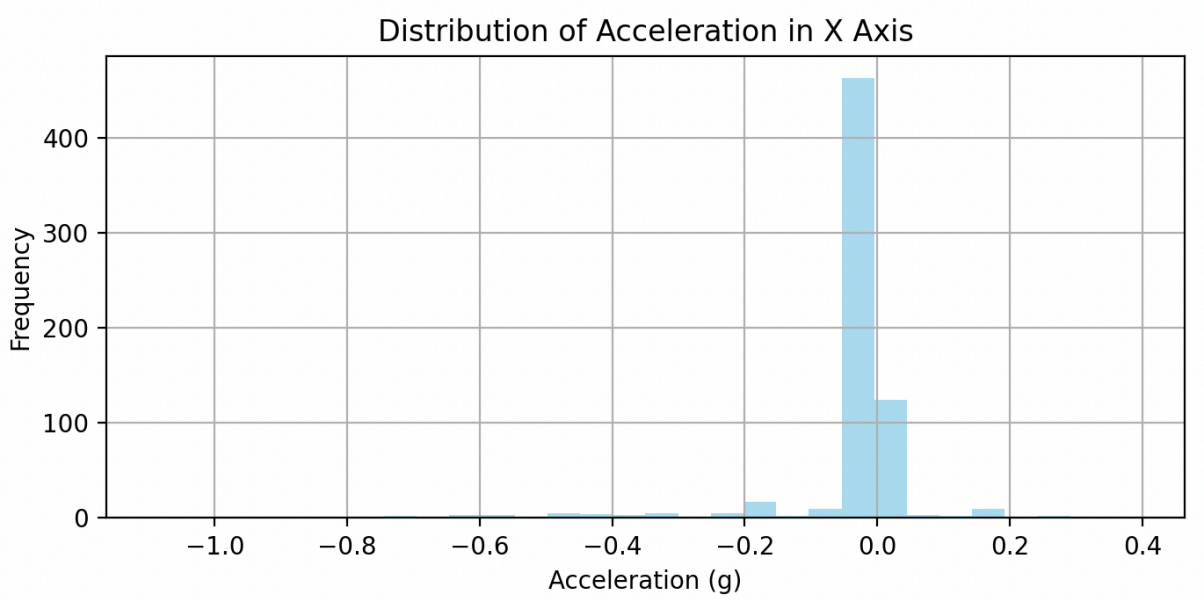

Figure 10: Frequency of Acceleration

Interpretation:
A narrow distribution centered around zero suggests stable lateral movements with minimal sway. Wide distributions or significant outliers may indicate unsteady gait patterns, highlighting potential areas of concern such as the risk of falling.

7.2.3 Temperature vs. Acceleration X:

This scatter plot correlates ambient temperature readings with X-axis acceleration, potentially to explore if environmental conditions affect gait characteristics.

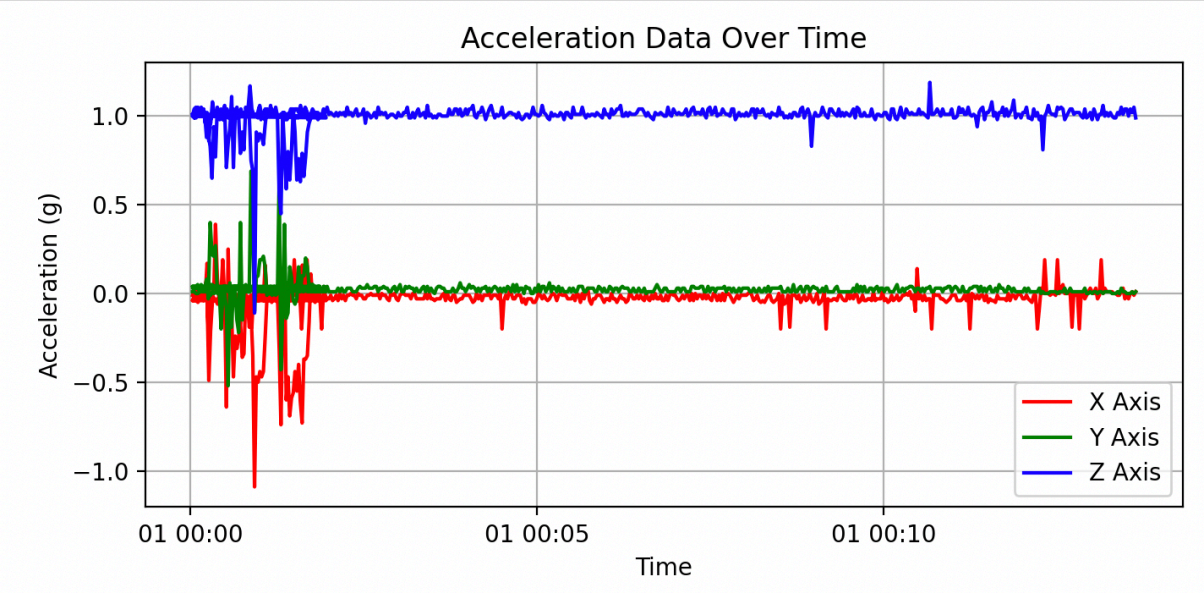

Figure 11: Acceleration over Temperature

Interpretation:
Clustering of data points might indicate consistent gait patterns under similar temperature conditions. Variations could suggest that external temperature influences the person's gait stability or comfort levels.

7.2.4 Comprehensive Sensor Data Analysis:

The multi-panel plot includes various scatter plots and histograms that provide a comprehensive overview of the relationships and distributions of all measured variables: acceleration across all axes, temperature, and humidity.

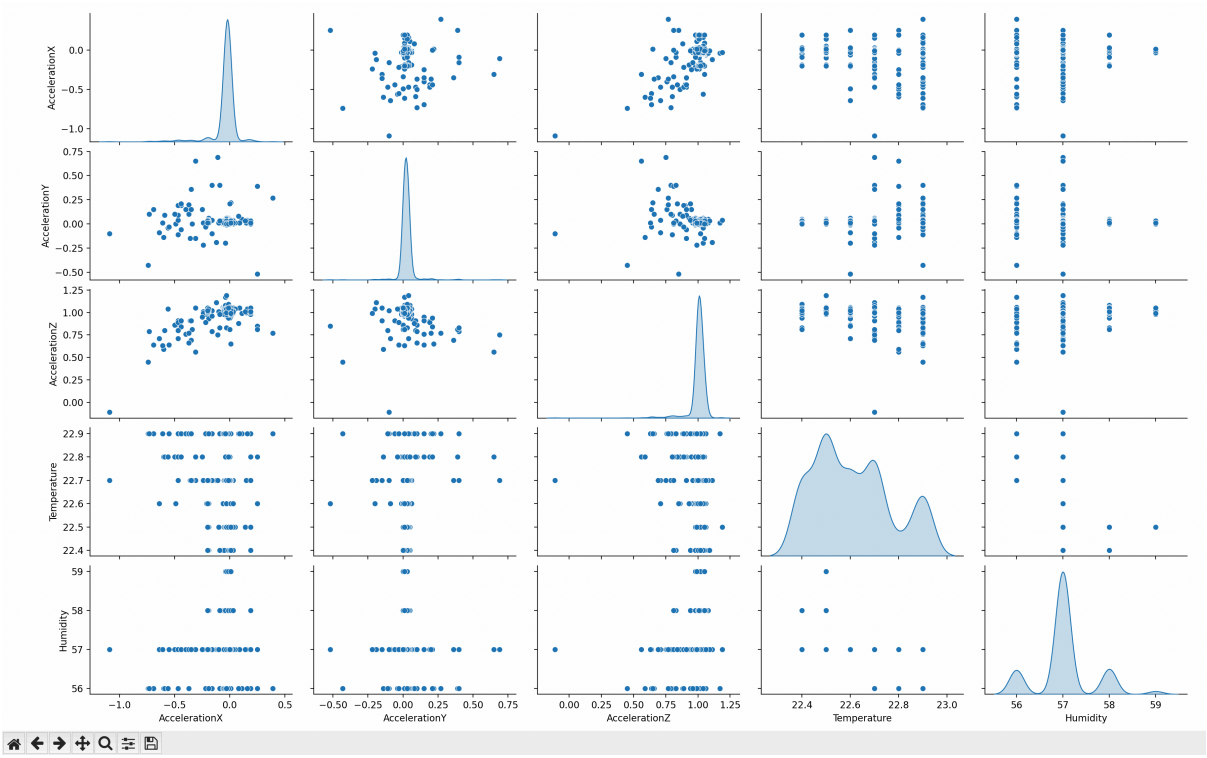

Figure 12: Comprehensive Look

Interpretation:

- This complete overview is invaluable for detecting correlations between different variables. For instance, if higher humidity levels correlate with decreased stability in the Z-axis, it might suggest that environmental conditions affect the person's gait.
- Patterns observed across these graphs can lead to insights about how different conditions or times of day affect the mobility and stability of elderly individuals.

7.2.5 How These Results Aid Gait Analysis:

The visualization of this data plays a pivotal role in medical and healthcare applications, particularly in gait analysis for the elderly:

- Early Detection of Mobility Issues: Continuous monitoring and analysis can help

in early detection of changes in gait that may signify health issues such as joint problems, neurological disorders, or risks of falling.

- Tailored Healthcare Responses: Understanding individual gait patterns and their variations under different conditions allows healthcare providers to customize treatments and interventions, such as physical therapy regimes tailored to specific gait abnormalities.
- Monitoring and Evaluation: Over time, data from these graphs can be used to evaluate the effectiveness of medical treatments or rehabilitation efforts, monitoring improvements or deterioration in mobility.

This case study demonstrates how the correct implementation of communication protocols can facilitate the effective collection and analysis of gait data, thereby contributing to better healthcare outcomes for elderly individuals.

8 Conclusion & Future Work

This research underscores the transformative potential of IoT technologies in the field of healthcare, particularly through the implementation of gait analysis for elderly care. Gait analysis is crucial for monitoring the mobility and stability of elderly individuals, providing essential data that can help in early detection of potential health issues and improving the overall quality of care. The integration of advanced sensor technology, including ultrasonic sensors, accelerometers, and DHT22 sensors, enabled comprehensive and real-time monitoring of gait patterns, offering valuable insights into the mobility of elderly individuals.

However, the effectiveness of gait analysis heavily relies on the communication protocols used to transmit sensor data. The choice of protocol impacts the latency, throughput, and reliability of data transmission, which are critical for timely and accurate analysis. This study evaluated three widely used IoT communication protocols—MQTT, AMQP, and CoAP—to determine their suitability for healthcare applications.

MQTT demonstrated the lowest latency, making it highly suitable for real-time applications where immediate data transmission is crucial. Its simplicity and widespread community support make it an accessible and practical choice for many healthcare scenarios. AMQP offered the highest reliability and robust security features, making

it ideal for environments where data integrity and security are paramount. However, its higher latency and complexity in setup may limit its use in some real-time applications. CoAP provided a balanced performance with moderate latency and throughput, particularly suitable for resource-constrained environments.

The quantitative metrics, including latency, throughput, and reliability, highlighted the strengths and weaknesses of each protocol. MQTT's low latency and high throughput make it ideal for real-time gait analysis applications, ensuring timely data transmission for immediate analysis and intervention. AMQP's robust security and reliability are crucial for maintaining the integrity of sensitive healthcare data, though it may not be as responsive for real-time applications. CoAP's moderate performance and suitability for constrained environments make it a versatile choice for various IoT applications in healthcare.

Qualitative insights into the ease of setup, availability of documentation, and community support further influenced the choice of protocol. MQTT stood out for its simplicity and quick deployment, while AMQP was preferred for secure and reliable data transmission.

The findings of this study emphasize the importance of selecting the right communication protocol based on the specific requirements of healthcare applications. Low latency and high throughput are critical for real-time monitoring, while security and reliability are essential for protecting sensitive patient data. Standardizing a communication protocol for gait analysis in healthcare IoT systems will ensure consistent and reliable data transmission, ultimately leading to better patient outcomes

8.1 Future Work

The research presented in this thesis lays a strong foundation for further exploration and development in the field of healthcare IoT. Several avenues for future work can build on the findings and insights gained from this study.

- Expansion of Sensor Network:
  Future research should aim to integrate additional sensors such as gyroscopes, heart rate monitors, and pressure sensors. This would provide a more comprehensive view of a patient's health, capturing a broader range of physiological data that could enhance the accuracy and scope of gait analysis.

- Longitudinal Studies:
  Conducting long-term studies to evaluate the efficacy and reliability of the IoT system in real-world healthcare settings is essential. These studies would help in understanding the long-term benefits, potential challenges, and practical applications of IoT technology in patient care over extended periods.

- Advanced Data Analytics:
  Leveraging machine learning and artificial intelligence for advanced data analytics can significantly improve the interpretation of the vast amounts of data generated by IoT devices. Predictive analytics could identify patterns and trends that are not immediately apparent, leading to earlier detection of health issues and more personalized care plans.

- Enhanced Security Measures:
  Developing and implementing advanced encryption techniques and secure communication protocols will be critical for protecting patient data. Ensuring compliance with healthcare regulations such as GDPR and HIPAA is necessary to maintain data privacy and security.

- Standardization and Interoperability:
  Promoting the development of standardized protocols and interfaces will ensure seamless interoperability between different IoT devices and healthcare systems. This will facilitate the integration of new technologies into existing healthcare infrastructures, enabling more cohesive and efficient patient care.

By addressing these areas, future research can continue to push the boundaries of IoT in healthcare, ultimately leading to more advanced, secure, and effective patient monitoring systems. These efforts will help realize the full potential of IoT in transforming healthcare delivery and improving patient outcomes.

A Appendix

A.1 Appendix 1

A.1.1 Arduino Code Implementation

File: Csvdata_map.ino

```
#include "Arduino_SensorKit.h"

#define TRIGGER_PIN 7 // Ultrasonic sensor trigger pin
#define ECHO_PIN 8    // Ultrasonic sensor echo pin

void setup() {
  // Initialize serial communication
  Serial.begin(9600);
  while (!Serial); // Wait for the serial port to connect

  // Sensor setup
  pinMode(TRIGGER_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
  Oled.begin();
  Oled.setFlipMode(true);
  Accelerometer.begin();

  // Print a single header line
  Serial.println("Timestamp,AccelerationX,AccelerationY,
      AccelerationZ,UltrasonicDistance,Temperature,Humidity");
}

void loop() {
  // Read accelerometer data
  float accelerationX = Accelerometer.readX();
  float accelerationY = Accelerometer.readY();
  float accelerationZ = Accelerometer.readZ();

  // Read ultrasonic sensor data
  long duration, distance;
  digitalWrite(TRIGGER_PIN, LOW);
```

```
  delayMicroseconds(2);
  digitalWrite(TRIGGER_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIGGER_PIN, LOW);
  duration = pulseIn(ECHO_PIN, HIGH);
  distance = duration * 0.034 / 2;

  // Read DHT22 sensor data
  float temp = Environment.readTemperature();
  float humidity = Environment.readHumidity();

  // Print data in CSV format
  Serial.print(millis());
  Serial.print(",");
  Serial.print(accelerationX, 2);
  Serial.print(",");
  Serial.print(accelerationY, 2);
  Serial.print(",");
  Serial.print(accelerationZ, 2);
  Serial.print(",");
  Serial.print(distance);
  Serial.print(",");
  Serial.print(temp, 2);
  Serial.print(",");
  Serial.println(humidity, 2);

  delay(1000); // Adjust delay as needed
}
```

A.2 Appendix 2

A.2.1 Python Code for Data Logging

File: collection.py

```
import serial
import csv
from datetime import datetime

# Open serial connection
def open_serial_connection():
    port = '/dev/tty.usbmodem142301'
    baud_rate = 9600
    try:
        return serial.Serial(port, baud_rate, timeout=1)
    except serial.SerialException as e:
        print(f"Error opening serial port: {e}")
        return None

def main():
    ser = open_serial_connection()
    if ser is None:
        print("Failed to open serial connection.")
        return

    file_path = '/Users/jayant/Documents/Trimester1-2024/SIT724/
       sensor_data.csv'
    with open(file_path, mode='a', newline='') as file:
        writer = csv.writer(file)
        print(f"File opened successfully at {file_path}")

        while True:
            line = ser.readline().decode('utf-8').strip()
            if line:
                print(f"Received line: {line}")  # Debugging output
                data = line.split(',')
                writer.writerow(data)
                file.flush()  # data is written to disk
                print("Data written to file.")  # Confirming data
```

```
                    write

if __name__ == "__main__":
    main()
```

A.3 Appendix 3

A.3.1 Python Script for Data Visualization

File: visualisation.py

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

file_path = '/Users/jayant/Documents/Trimester1-2024/SIT724/
    sensor_data.csv'
data = pd.read_csv(file_path)
data['Timestamp'] = pd.to_datetime(data['Timestamp'], unit='ms')
data.set_index('Timestamp', inplace=True)

plt.figure(figsize=(14, 7))
plt.subplot(2, 2, 1)
plt.plot(data.index, data['AccelerationX'], label='X Axis', color='r
    ')
plt.plot(data.index, data['AccelerationY'], label='Y Axis', color='g
    ')
plt.plot(data.index, data['AccelerationZ'], label='Z Axis', color='b
    ')
plt.title('Acceleration Data Over Time')
plt.xlabel('Time')
plt.ylabel('Acceleration (g)')
plt.legend()
plt.grid(True)

plt.subplot(2, 2, 2)
plt.hist(data['AccelerationX'], bins=30, color='skyblue', alpha=0.7)
plt.title('Distribution of Acceleration in X Axis')
plt.xlabel('Acceleration (g)')
plt.ylabel('Frequency')
plt.grid(True)

plt.subplot(2, 2, 3)
plt.scatter(data['Temperature'], data['AccelerationX'], alpha=0.7,
    color='purple')
```

```
30 plt.title('Temperature vs. Acceleration X')
31 plt.xlabel('Temperature (°C)')
32 plt.ylabel('Acceleration X (g)')
33 plt.grid(True)
34
35 plt.tight_layout()
36 plt.show(block=False)
37
38 sns.pairplot(data[['AccelerationX', 'AccelerationY', 'AccelerationZ'
       , 'Temperature', 'Humidity']], diag_kind='kde')
39 plt.suptitle('Pair Plot of Sensor Data', y=1.02)
40 plt.show()
```

A.4 Appendix 4

A.4.1 MQTT Protocol Implementation - Publish

File: mqtt.py

```
import serial
import paho.mqtt.client as mqtt

# Configure serial port (replace with your correct port)
serial_port = '/dev/cu.usbmodem144201'
baud_rate = 9600
ser = serial.Serial(serial_port, baud_rate, timeout=1)

# MQTT broker configuration
broker_ip = 'localhost'  # Replace with your broker's IP
topic = 'sensor/gait'

# Initialize MQTT client
client = mqtt.Client()
client.connect(broker_ip, 1883, 60)

while True:
    # Read from Arduino
    line = ser.readline().decode('utf-8').strip()
    if line:
        print(f"Received data: {line}")
        client.publish(topic, line)
```

A.4.2 MQTT Protocol Implementation - Subscribe

File: mqtt_subscribe.py

```
import paho.mqtt.client as mqtt

# MQTT broker configuration
broker_ip = 'localhost'  # Replace with your broker's IP
topic = 'sensor/gait'

# Callback function when a message is received
def on_message(client, userdata, message):
    print(f"Received message: {message.payload.decode()}")

# Initialize MQTT client
client = mqtt.Client()
client.connect(broker_ip, 1883, 60)

# Set the callback function
client.on_message = on_message

# Subscribe to the topic
client.subscribe(topic)

# Loop to listen for messages
client.loop_forever()
```

A.5 Appendix 5

A.5.1 AMQP Protocol Implementation - Send

File: amqp_send.py

```
import serial
import pika

# Configure the serial port (replace with your correct port)
serial_port = '/dev/cu.usbmodem144201'
baud_rate = 9600
ser = serial.Serial(serial_port, baud_rate, timeout=1)

# RabbitMQ setup
amqp_host = 'localhost'  # Replace with your RabbitMQ broker's IP if
    needed
queue_name = 'sensor_queue'

# Connect to RabbitMQ
connection = pika.BlockingConnection(pika.ConnectionParameters(host=
    amqp_host))
channel = connection.channel()

# Declare the queue
channel.queue_declare(queue=queue_name)

while True:
    # Read data from Arduino
    line = ser.readline().decode('utf-8').strip()
    if line:
        print(f"Received data: {line}")
        # Publish data to RabbitMQ
        channel.basic_publish(exchange='', routing_key=queue_name,
            body=line)
```

A.5.2 AMQP Protocol Implementation - Receive

File: amqp_receive.py

```
import pika

# RabbitMQ connection parameters
amqp_host = 'localhost'  # Replace if RabbitMQ is on a different host
queue_name = 'sensor_queue'

# Establish a connection to RabbitMQ
connection = pika.BlockingConnection(pika.ConnectionParameters(host=amqp_host))
channel = connection.channel()

# Ensure the queue exists
channel.queue_declare(queue=queue_name)

# Callback function to process messages
def callback(ch, method, properties, body):
    print(f" [x] Received {body.decode('utf-8')}")

# Set up consumer to listen to the specified queue
channel.basic_consume(queue=queue_name, on_message_callback=callback, auto_ack=True)

print(' [*] Waiting for messages. To exit press CTRL+C')
channel.start_consuming()
```

A.6 Appendix 6

A.6.1 CoAP Protocol Implementation - Server

File: coap_server.py

```
import asyncio
from aiocoap import Context, Message, CHANGED
import aiocoap.resource as resource

class SensorDataResource(resource.Resource):
    def __init__(self):
        super().__init__()

    async def render_post(self, request):
        payload = request.payload.decode('utf-8')
        print(f"Received data: {payload}")
        return Message(code=CHANGED, payload=b'Data received')

async def main():
    root = resource.Site()
    root.add_resource(['sensor', 'data'], SensorDataResource())

    await Context.create_server_context(root, bind=('localhost',
        5683))
    print("CoAP server running on coap://localhost:5683")
    await asyncio.get_running_loop().create_future()

if __name__ == "__main__":
    asyncio.run(main())
```

A.6.2 CoAP Protocol Implementation - Client

File: coap_client.py

```
import asyncio
import serial
import json
import time
from aiocoap import *

# Configure serial port (replace with your correct port)
serial_port = '/dev/cu.usbmodem141201'
baud_rate = 9600
ser = serial.Serial(serial_port, baud_rate, timeout=1)

async def main():
    protocol = await Context.create_client_context()

    while True:
        line = ser.readline().decode('utf-8').strip()
        if line:
            sensor_data = {
                'timestamp': time.time(),
                'data': line
            }
            payload = json.dumps(sensor_data).encode('utf-8')
            request = Message(code=POST, payload=payload, uri='coap
                ://localhost/sensor/data')
            response = await protocol.request(request).response
            print(f"Sent data: {sensor_data}, Response: {response.
                payload.decode('utf-8')}")
            await asyncio.sleep(1)

if __name__ == "__main__":
    asyncio.run(main())
```

References

[1] I. Brass, L. Tanczer, M. Carr, M. Elsden, and J. Blackstock, Standardising a moving target: The development and evolution of iot security standards, (2018).

[2] J.-J. Chou, C.-S. Shih, W.-D. Wang, and K.-C. Huang, Iot sensing networks for gait velocity measurement, International Journal of Applied Mathematics and Computer Science, 29 (2019), pp. 245–259.

[3] M. Dauwed, J. Yahaya, Z. B. Mansor, and A. R. Hamdan, Determinants of internet of things services utilization for health information exchange, ARPN Journal of Engineering and Applied Sciences, 12 (2017), pp. 10490–10501.

[4] L. et al., Iot and data analytics in healthcare monitoring systems, in Journal of Information and Computational Science, vol. 11, ISSN: 1548-7741, 2021, Journal of Information and Computational Science, pp. 210–224.

[5] K. T. Kadhim, A. M. Alsahlany, S. M. Wadi, and H. T. Kadhum, An overview of patient's health status monitoring system based on internet of things (iot), Wireless Personal Communications, 114 (2020), pp. 2235–2262.

[6] V. S. Naresh, S. S. Pericherla, P. S. R. Murty, and S. Reddi, Internet of things in healthcare: Architecture, applications, challenges, and solutions, Computer Systems Science & Engineering, 35 (2020), pp. 411–421.

[7] B. Pradhan, S. Bhattacharyya, and K. Pal, Harnessing the opportunities of iot in healthcare (hoit): Trends, challenges, and future directions, Journal of Healthcare Engineering, 2021 (2021), pp. Article ID 6632599, 18 pages.

[8] J. Qi, P. Yang, A. Waraich, Z. Deng, Y. Zhao, and Y. Yang, Examining sensor-based physical activity recognition and monitoring for healthcare using internet of things: A systematic review, Journal of Biomedical Informatics, 87 (2018), pp. 138–153.

[9] J. Saleem, M. Hammoudeh, U. Raza, B. Adebisi, and R. Ande, Iot standardisation: Challenges, perspectives and solution, in Proceedings of the 2nd international conference on future networks and distributed systems, 2018, pp. 1–9.

[10] M. Single, L. C. Bruhin, A. Colombo, K. Möri, S. M. Gerber, J. Lahr, P. Krack, S. Klöppel, R. M. Müri, U. P. Mosimann, et al., A transferable lidar-based method to conduct contactless assessments of gait parameters in diverse home-like environments, Sensors, 24 (2024), p. 1172.

[11] T. M. Tukade and R. M. Banakar, Data transfer protocols in IoT–an overview, International Journal of Pure and Applied Mathematics, 118 (2018), pp. 121–138.

[12] L. S. Vailshery, Number of iot connected devices worldwide 2019-2023, with forecasts to 2030, (2023).

[13] D. Xu, H. Zhou, W. Quan, X. Jiang, M. Liang, S. Li, U. C. Ugbolue, J. S. Baker, F. Gusztav, X. Ma, et al., A new method proposed for realizing human gait pattern recognition: inspirations for the application of sports and clinical gait analysis, Gait & Posture, 107 (2024), pp. 293–305.

[14] S. Zeadally and O. Bello, Harnessing the power of internet of things based connectivity to improve healthcare, Internet of Things, 14 (2021), p. 100074.

[15] S. Zeadally, F. Siddiqui, Z. Baig, and A. Ibrahim, Smart healthcare: Challenges and potential solutions using internet of things (iot) and big data analytics, PSU Research Review, 4 (2020), pp. 93–109.