PRIVACY-PROTECTED ELDERLY MONITORING SYSTEM USING WI-FI AND WEARABLE SENSORS

By

Jincheng Zhou (jz107@illinois.edu)

Junyue Jiang (junyuej2@illinois.edu)

Pu Lin (pulin2@illinois.edu)

Zizhao Cao (zizhaoc2@illinois.edu)

Final Report for ECE 445, Senior Design, Spring 2024

Sponsor: Professor Huan Hu

10 May 2024

Project No. 35

Contents

1 Introduction 1
  1.1 Problem 1
  1.2 Solution 1
  1.3 Visual Aid 2
  1.4 High-level Requirements Lists 2
2 Design 3
  2.1 Block Diagram 3
  2.2 Physical Design 4
  2.3 Subsystem Overview 4
  2.4 Wi-Fi Source Subsystem 4
    2.4.1 Initial Considerations and Challenges 5
    2.4.2 Transition to Nexmon CSI 5
    2.4.3 Detailed Implementation with Raspberry Pi 4B and Nexmon CSI 5
    2.4.4 Overcoming Challenges 6
    2.4.5 Conclusion 6
  2.5 Wearable Device Subsystem Requirement 7
    2.5.1 Sensor Module 7
    2.5.2 Power Module 7
    2.5.3 Wi-Fi Communication Module 7
  2.6 Detector Subsystem Requirement 8
    2.6.1 Software Development and Integration 8
    2.6.2 Equations and Algorithmic Description 8
    2.6.3 Algorithm Integration and System Design 9
    2.6.4 Integration with Hardware 10
  2.7 Wi-Fi Detector Algorithm Subsystem 10
    2.7.1 Wi-Fi-Based Respiration Monitoring 10
    2.7.2 Motion Sensor Integration 11
    2.7.3 Implementation Details 11
    2.7.4 Additional Considerations 14

2.8 Requirements and Verification 14
2.8.1 Wi-Fi Source Subsystem Analysis 14
2.8.2 Wearable Device Subsystem Analysis 16
2.8.3 Detector Subsystem Analysis 17
3 Cost and Schedule 19
3.1 Cost 19
4 Ethics & Safety 19
References 20

1 Introduction

1.1 Problem

China is undergoing a significant demographic transformation entering an aging society. Nearly a third population will be over 60 years old by 2050 as projected by "National Development Bulletin on Ageing 2020.". Already, over 264 million Chinese citizens are aged 60 and above, making up approximately 18.7% of the nation's total population. [1] This demographic shift has led to an increased demand for elderly care services, placing Elderly Care Centers at the forefront of providing essential care and safety for this vulnerable group. Yet, these centers face substantial challenges in monitoring and responding to emergencies, such as falls or medical crises, without compromising the privacy and dignity of the elderly, as traditional methods of monitoring the elderly, such as hiring professional caregivers, have the main disadvantage of high manpower costs and cannot be extended to most elderly institutions. On the other hand, monitoring with a caregiver or visual recognition program reduces costs, but the demand for surveillance equipment undoubtedly violates the privacy of the elderly to a certain extent. Considering the harm to the psychological health of the elderly, the market urgently needs a monitoring program that can solve the above issues. [2]

A survey among Elderly Care Centers in China revealed a significant lack of advanced mechanisms for detecting emergencies while preserving privacy.[3] This inadequacy in care provision highlights an urgent need for innovative solutions that balance efficient emergency detection with the preservation of privacy and dignity. [4] Developing such systems is essential not only for enhancing elderly care but also for adapting to China's changing demographic landscape, making it imperative to invest in technologies that ensure safety and respect for the elderly simultaneously.[5]

1.2 Solution

To bridge this gap, we propose an Emergency Detection System specifically designed for elderly individuals. This system combines two innovative components to ensure both efficacy and privacy. The first component is Wi-Fi Emergency System: Utilizing advanced signal processing and deep learning techniques, this system interprets Wi-Fi signal disruptions caused by human movement within its coverage area. By analyzing these disruptions, the system can identify unusual patterns indicative of falls or other emergencies without the need for visual surveillance, thereby maintaining privacy. [3]

The second component is Wearable Devices that complements the Wi-Fi Emergency System. These devices are equipped with motion and health sensors. They are designed to be lightweight, unobtrusive, and capable of providing real-time data on the wearer's physical state. In the event of an abnormality, the device can trigger an immediate alert to caregivers for prompt response.

1.3 Visual Aid

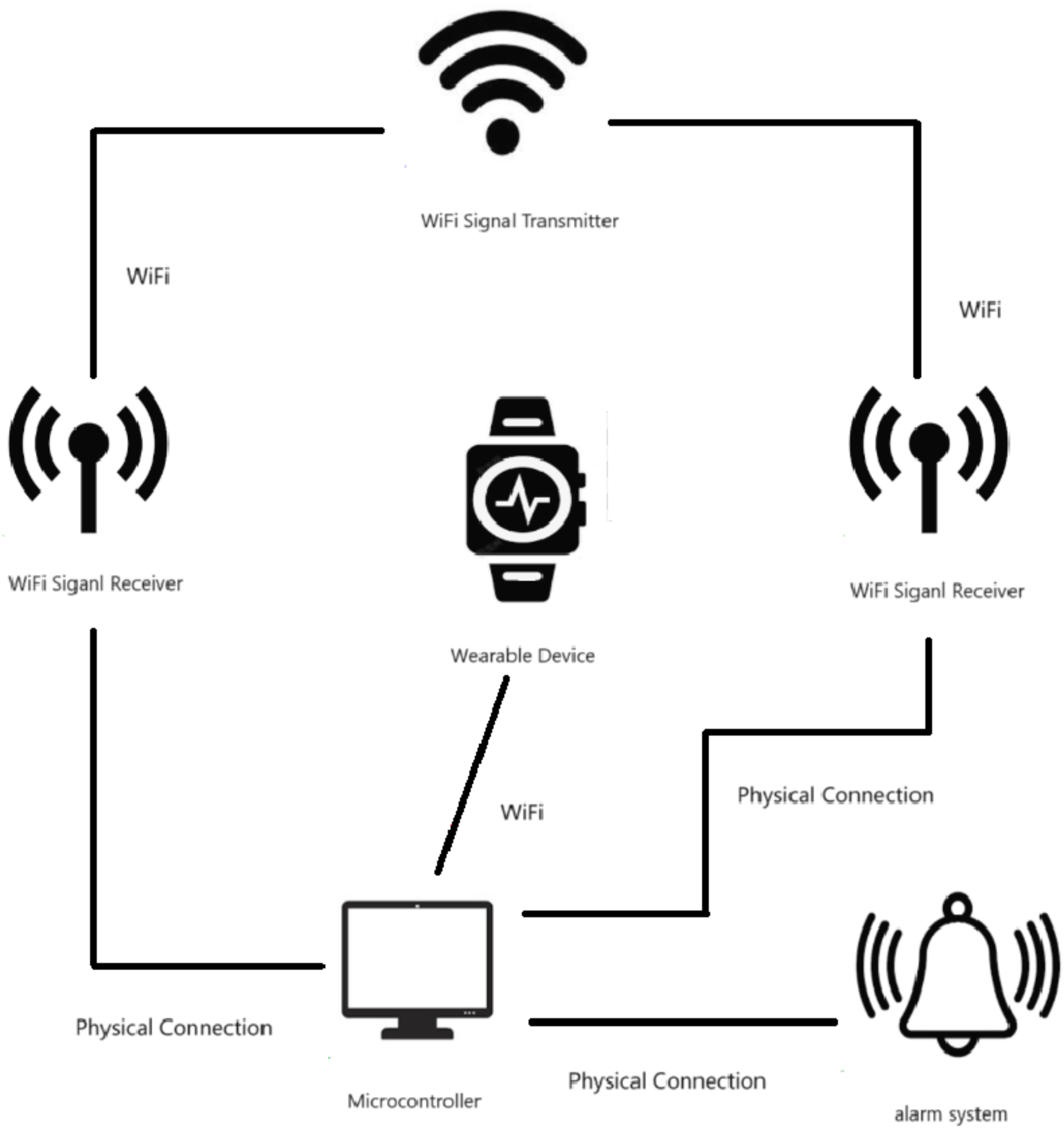

Figure 1: Visual Aid

1.4 High-level Requirements Lists

- For the recognition of the action of falling, the correct rate should be 100%, the probability of false alarms should be maintained at less than 20%, and the probability of miss should be maintained at least at less than 5% and 0 miss should be the target.
- The complete system should be able to operate in a 20 square meter scenario and maintain above recognition accuracy. Real-time signal monitoring should be maintained within this area
- The system alarm should be triggered within 500 ms after the falling action.

2 Design

2.1 Block Diagram

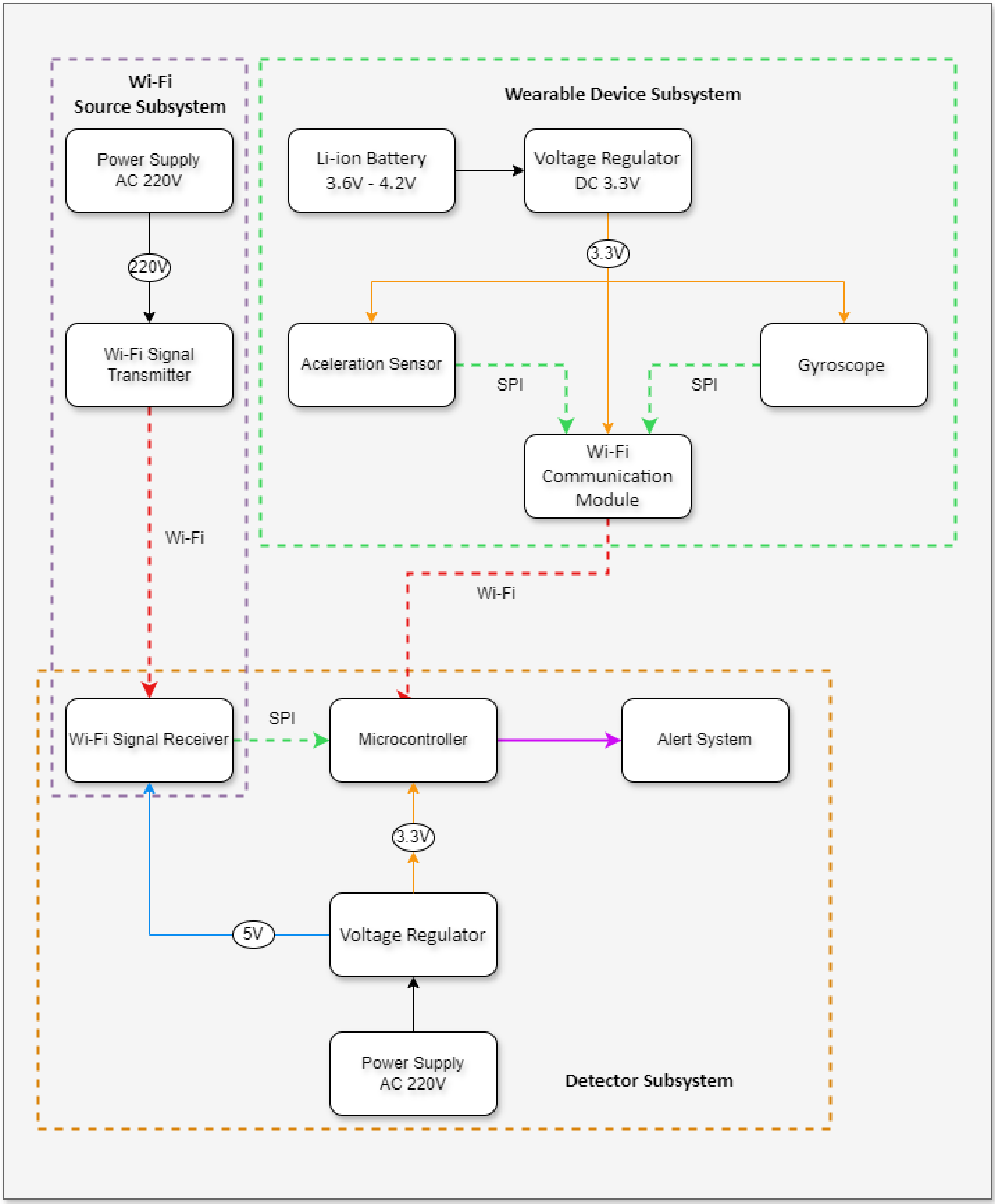

Figure 2: Block Diagram

2.2 Physical Design

2.3 Subsystem Overview

The Wi-Fi Source Subsystem will be a high-performance Wi-Fi router capable of dual-band signaling (2.4 GHz and 5 GHz), providing extensive coverage and penetration through various obstacles commonly found in residential settings. It should be capable of sustaining multiple device connections simultaneously without degradation of service quality. The router will have high-gain antennas to ensure the signal's strength is maintained even at the edges of the coverage area. The Wi-Fi signal will be received by the Detector Subsystem and used to determine the current movement of the tester.

The Wearable Device Subsystem is designed to collect acceleration as well as angle information while being worn on the tester's wrist and send it to the software processing section. The subsystem needs to consist of at least one acceleration sensor module and one gyroscope module to collect enough information. The information is sent to the software processing section via a transmission module. The entire subsystem should be powered by a separate power supply module that can provide 3-5 V voltage. Similarly, the acceleration and angular data collected by the sensors in three dimensions will be received by the Detector Subsystem and used to determine the current movement of the tester.

The Detector Subsystem is a standalone module with its own dedicated casing, designed to be lightweight and compact, potentially the size of a small router or a large smartphone to be placed within the living space of the elderly person. It requires a stable power source, typically 5 V supplied via a USB connection or wall adapter, and must have an internal voltage regulator to provide a clean 3.3 V power supply for its internal electronics, with at least 500 mA current capacity to support its operation.

The core of this subsystem is a high-performance microcontroller or a microprocessor with a fast clock speed, sufficient to process data from both the wearable sensors and the Wi-Fi signal strength information. It should possess robust communication interfaces like SPI and I2C to interface with the Wi-Fi module and possibly additional UART interfaces for debugging and future expansions.

The Wi-Fi module should be capable of operating in dual-band (2.4 GHz and 5 GHz) to ensure comprehensive coverage and the ability to analyze signal strength with a high degree of accuracy. The SPI or UART interface with the microcontroller must support high data rate transfer to prevent any data bottleneck.

Moreover, the Detector Subsystem should feature onboard memory (RAM and flash) to log events and store the necessary analysis algorithms, ensuring that a transient loss in connectivity does not result in data loss. The received Wi-Fi signal and sensor data will be used in this sub-system to determine the tester's movements and cross validate for improved accuracy.

2.4 Wi-Fi Source Subsystem

In this project, we focused on implementing the hardware necessary to capture Wi-Fi Channel State Information (CSI) signals using a Raspberry Pi 4B equipped with Nexmon CSI, in conjunction with a Wi-Fi router. This setup is pivotal for our research in Wi-Fi signal monitoring and analysis.

2.4.1 Initial Considerations and Challenges

The project began with the intent to use the CSI Tool, leveraging its advantages for our initial setups. We procured Intel 5300 network cards based on their compatibility with the CSI Tool, anticipating a straightforward implementation. However, we encountered a significant obstacle: the Intel 5300 cards were incompatible with the laptops available to our team, which halted our progress. After attempting various configurations and considering different laptops, it became evident that this approach was not feasible due to hardware limitations and the specific setup requirements of the Intel 5300 cards.

2.4.2 Transition to Nexmon CSI

Given the setbacks with the CSI Tool and Intel 5300, we pivoted to using Nexmon CSI, which is known for its robust support on Raspberry Pi platforms. This decision came after evaluating various alternatives that could provide similar data quality and reliability.

2.4.3 Detailed Implementation with Raspberry Pi 4B and Nexmon CSI

Implementing the Wi-Fi CSI data collection using the Raspberry Pi 4B with Nexmon CSI involved several crucial steps, each tailored to ensure the system was robust and capable of capturing high-quality CSI data. Below is a detailed breakdown of the process:

Preparation and Initial Setup

- Raspberry Pi 4B Setup: We started by setting up the Raspberry Pi 4B with a fresh installation of Raspberry Pi OS. It was important to ensure the OS was up to date to avoid any compatibility issues with the Nexmon CSI firmware.
- Nexmon CSI Installation: We installed Nexmon, a firmware modification framework for the Broadcom Wi-Fi chips used in the Raspberry Pi. This involved cloning the Nexmon repository from GitHub, installing necessary dependencies like libisl, libmpfr, and libmpc, and building the tools and patches specific to our Broadcom chip model.

Firmware Modification

- Compiling Nexmon Firmware: With the Nexmon environment set up, we compiled a modified firmware that enables CSI collection. This step required careful configuration to match the specific Wi-Fi chipset of the Raspberry Pi 4B.
- Flashing Modified Firmware: After successful compilation, the next step was to flash this modified firmware onto the Raspberry Pi’s Wi-Fi chip. We backed up the original firmware before proceeding to ensure we could revert if necessary.

Router Configuration

- Router Choice and Setup: We selected a Wi-Fi router that supported the specific channels and frequencies compatible with our Nexmon CSI setup. The router was configured to use a static channel rather than auto-selecting channels to maintain consistency in the data collected.

- Transmission Power and Bandwidth Settings: We adjusted the transmission power and bandwidth settings on the router to optimize signal clarity and strength, ensuring that the Raspberry Pi could reliably capture CSI data even at varying distances and through physical obstructions.

Data Collection Setup

- Monitoring Mode Activation: On the Raspberry Pi, we activated the monitoring mode that Nexmon CSI provides. This mode allows the device to listen for Wi-Fi packets without connecting to the network, which is crucial for passive data collection.
- Scripting Automated Data Capture: We wrote scripts to automate the process of starting and stopping data capture, as well as handling data storage. These scripts helped manage large volumes of data efficiently, segmenting captures by time or event, depending on the experiment's needs.

Validation and Testing

- Initial Data Checks: Once the setup was complete, we conducted initial tests to check the integrity and format of the CSI data captured. This was crucial to ensure that the data was not only accurate but also consistent with our expectations for analysis.
- Ongoing Adjustments: Based on initial feedback and data quality, we fine-tuned the system by adjusting the router's settings and the Raspberry Pi's placement. These adjustments were necessary to accommodate different environments and scenarios where signal strength and quality could vary.

Integration with Analysis Tools

- Data Processing Tools Setup: The CSI data collected was integrated with our existing signal processing and analysis tools. We made necessary modifications to these tools to parse and analyze the Nexmon CSI data format effectively.
- Real-time Analysis Capability: We also implemented real-time data processing capabilities to observe the Wi-Fi environment dynamically, which was vital for applications requiring immediate response or adjustment based on the CSI data.

2.4.4 Overcoming Challenges

The shift from Intel 5300 and the CSI Tool to Nexmon CSI required us to adapt our data processing scripts and analysis tools. The data format and capabilities provided by Nexmon CSI differed slightly, necessitating modifications to our existing algorithms and processing techniques.

2.4.5 Conclusion

The transition to Nexmon CSI on the Raspberry Pi 4B, despite initial challenges, proved to be successful. This setup not only met our requirements for capturing Wi-Fi CSI data but also enhanced our project's flexibility and scalability. The Raspberry Pi 4B and Nexmon CSI combination is a cost-effective and efficient solution for Wi-Fi signal analysis, particularly suitable for research and development projects involving environmental sensing and motion detection.

2.5 Wearable Device Subsystem Requirement

2.5.1 Sensor Module

The sensor module should be able to measure and output three-way acceleration and three-way rotational angular velocity data with a certain degree of accuracy under the power supply module, and transmit the data in real time in a certain format to the Wi-Fi data transmission module, which means that the module should at least contain acceleration sensors and gyroscopes to complete the measurement of the data, and should support the I2C or TCP protocols to transmit the data to the corresponding module. Based on the above requirements, we believe that the IMU sensor (MCU6050) can meet the needs.

The MPU6050 is a widely used motion-tracking device that integrates a 3-axis accelerometer and a 3-axis gyroscope on a single chip. This powerful sensor module is an integral part of the wearable sensor subsystem in the project. Its ability to accurately measure acceleration and rotational movement makes it ideal for monitoring the activity and movements of elderly individuals.

2.5.2 Power Module

The power module will be used to power the sensor module and the Wi-Fi data transmission module. Considering the size requirement of wearable devices: no more than twice the size of a common watch and the portability requirement: as little as possible obstruction of the test movement while wearing it, we ruled out the option of powering it through a power cord and initially planned that the power module should contain a power supply module that acts as a voltage regulator that converts an input voltage of 3.7 V - 5 V into an output voltage of about 3.3 V. The power supply module should contain a voltage regulator that converts the 3.7 V – 5 V input voltage into an output voltage of about 3.3 V for the sensor module and about 5 V for the Wi-Fi data transmission module. The power supply will be powered by a battery pack that holds up to two common Li-ion 5 or 7 batteries, which we expect to be able to provide an output voltage of at least 6 V at an expected output voltage of 7.4 V and an input current of up to 1 A to meet the input requirements of the power supply module.

The wearable sensor subsystem will use a battery pack with 4 AAA batteries, providing 6V. Since the ESP8266 operates at 3.3V and the MPU6050 requires 5V, appropriate power regulation and distribution are essential.

2.5.3 Wi-Fi Communication Module

The Wi-Fi data transmission module, powered by the power supply module, will receive the data transmitted from the sensor module in real time and perform operations such as formatting, after which the module will send the processed data to the processor subsystem. This requires that the module should support common Wi-Fi protocols such as IEEE 802.11b/g/n Wi-Fi protocol and have a suitable processor chip to perform the potentially required data processing operations. Based on the above requirements, we believe that the ESP8266 wireless Wi-Fi module can meet our needs.

In the project, the Wi-Fi communication module is critical for transmitting sensor data from the wearable device to a central monitoring system. By utilizing the Station (STA) mode on the ESP8266 and the User Datagram Protocol (UDP), we can achieve efficient and reliable data transmission to a PC or server.

2.6 Detector Subsystem Requirement

2.6.1 Software Development and Integration

The development and integration of the software that analyzes data from both the Wi-Fi and wearable device subsystems involved:

1. **Algorithm Design**: Creating algorithms capable of interpreting disruptions in Wi-Fi signals and inputs from wearable devices to detect falls. The algorithms use a combination of signal processing and machine learning techniques to distinguish between normal movements and potential falls.
2. **Data Processing**: Developing scripts to preprocess and filter the data received from the hardware subsystems. This includes applying filters to Wi-Fi Channel State Information (CSI) and accelerometer data to enhance the accuracy of the fall detection process.
3. **Machine Learning Implementation**: Implementing and training machine learning models to predict falls based on processed data. The models were trained using the FARSEEING database, which provides extensive real-world fall data, ensuring the models are robust and reliable in various scenarios.

2.6.2 Equations and Algorithmic Description

In our fall detection system design, we have incorporated insights and techniques based on real-world data analysis from the study by Bagalà, which provides a benchmark for accelerometer-based algorithms. The evaluation highlighted the complexity of accurately detecting falls in real scenarios, emphasizing the need for robust detection mechanisms that can differentiate between falls and regular activities.

Key Equations and Algorithm Parameters:

1. **Impact Detection**:

    - The fall detection starts by identifying significant impacts. The standard approach uses a threshold on the sum vector SV of the accelerometer outputs:

    $$SV = \sqrt{a_x^2 + a_y^2 + a_z^2}$$

    - An impact is suspected if SV exceeds a predetermined threshold, typically derived from empirical data to optimize sensitivity (SE) and specificity (SP).

2. **Orientation Change**:

- Post-impact, the orientation change is calculated using the dot product of acceleration vectors before and after the impact. This helps in confirming a fall by analyzing the change in body orientation:

$$Change\ in\ orientation = \text{acos}\,(\frac{a_{pre} \times a_{post}}{|a_{pre}||a_{post}|})$$

- A significant angular change supports the detection of a fall.

3. **Posture Analysis**:

   - After detecting a fall, the algorithm checks the subject's posture using low-pass filtered (LPF) signals to determine if the person remains lying down, which is critical in confirming fall events:

$$Posture\ check = mean(LPF(a_z)) < 0.5g$$

These methods have been tested and validated under real-world conditions, where falls were recorded among elderly patients with high fall-risk, significantly contributing to the tuning of our detection algorithms.[2]

Feature Extraction

To effectively detect falls, the system extracts various statistical features from the accelerometer and gyroscope data within a sliding window. The following features are computed for each axis (ax, ay, az, gx, gy, gz) as well as for the magnitude of acceleration and gyroscope data. Each feature provides unique insights into the movement and orientation data, helping to distinguish between normal activities and falls: **Mean, Standard Deviation, Variance, Min value, Max value, Skewness, Kurtosis, Spectral Entropy.**

2.6.3 Algorithm Integration and System Design

The system design integrates these algorithms to perform sequential checks:

1. **Data Acquisition**: Continuous monitoring using tri-axial accelerometers placed at strategic body locations to capture movement data.

2. **Real-time Processing**: The accelerometer data is processed in real-time to calculate the sum vector and check against the impact threshold.

3. **Orientation and Posture Analysis**: Upon passing the impact threshold, orientation changes are calculated, and posture is analyzed to confirm a fall.

4. **Alarm Triggering**: If a fall is confirmed, the system automatically triggers an alert to notify caregivers or medical personnel.

This integration ensures that the fall detection system is not only sensitive to the various dynamics of a fall but also minimizes false positives, a common issue in less discriminating systems.

Performance Metrics

- **Sensitivity (SE)**: Measures the percentage of falls correctly detected.
- **Specificity (SP)**: Measures the accuracy in disregarding non-fall events.
- **False Alarms**: Number of incorrect fall detections per day, which is critical for user acceptance.[3]

The algorithms have been refined based on feedback from initial testing, where real-world fall data helped in calibrating the detection thresholds to balance sensitivity and specificity effectively.

By utilizing these advanced algorithms and incorporating real-world testing data, the fall detection system is optimized to provide reliable and timely detection, thereby enhancing the safety and confidence of the elderly population.

2.6.4 Integration with Hardware

To ensure that the software components were well-integrated with the hardware, following integrations are necessary:

- Regular testing sessions with the hardware teams to ensure data consistency and reliability, adjustments to data transmission protocols to optimize speed and reduce latency.
- Setting Sensor Configuration: Utilized tri-axial seismic acceleration sensors (LIS3LV02DQ STMicroelectronics) [4] fixed at the lower back, mirroring the setup in the study. This sensor configuration captures movements in three dimensions (vertical, medio-lateral, and anterior-posterior), essential for detecting the direction and impact of falls.
- Sampling and Resolution in data acquisition: Adjusted our system to handle data from sensors with variable sampling frequencies (50 Hz to 250 Hz) and acceleration ranges (±2 g to ±12 g), ensuring the system's robustness against the so-called “clipping effect” observed at ±2 g in some of the recorded falls.[3]

2.7 Wi-Fi Detector Algorithm Subsystem

2.7.1 Wi-Fi-Based Respiration Monitoring

CSI Data Acquisition: Continuous monitoring and collection of Channel State Information (CSI) from WiFi devices provide the raw data necessary for analysis. The system utilizes the fine-grained temporal fluctuations in CSI caused by the respiratory movement of a person's chest and abdomen to detect breathing patterns.

Signal Processing and Filtering: The raw CSI data is processed to filter out noise and irrelevant information. Techniques such as band-pass filtering are applied to isolate the frequency bands that are most affected by human respiration, typically within the 0.1 Hz to 0.5 Hz range, which corresponds to normal human breathing rates.

Multi-Domain Analysis: By employing a multi-domain analysis that includes both Doppler shifts (frequency domain) and Angle of Arrival (AoA) (spatial domain), the system can construct a two-dimensional Doppler AoA map (DAM). This map enables the differentiation of respiration signals from multiple individuals in the monitored environment.

Super Resolution Doppler AoA Map Construction: To enhance the resolution of the DAM and improve the accuracy of respiration rate estimation, a super-resolution technique is implemented. This approach is crucial for accurately clustering and identifying individual breathing patterns, especially when the number of Wi-Fi antennas is limited.

Clustering and Respiration Rate Estimation: The peaks within the DAM represent potential respiration signals. A clustering algorithm, such as DBSCAN, groups these peaks based on their proximity, each cluster corresponding to an individual's respiration signal. The centroid of each cluster is used to estimate the respiration rate accurately.

2.7.2 Motion Sensor Integration

Motion Detection: Motion sensors placed strategically around the living space detect physical movement. This data provides context to the Wi-Fi-based respiration monitoring, enabling the system to differentiate between different types of movements (e.g., walking, falling).

Event Classification: The system classifies detected movements into normal activities and potential emergencies using machine learning algorithms trained on datasets of elderly movement patterns. Features such as the intensity, speed, and nature of the movement are analyzed.

Data Fusion and Analysis: Information from the motion sensors is fused with the respiration rate data from the Wi-Fi-based monitoring. This combined analysis allows for more accurate emergency detection, as it can identify situations where abnormal respiration patterns coincide with unusual physical movements.

2.7.3 Implementation Details

Phase 1: Planning and Design

Define Objectives: Clearly outline what emergencies the system will detect (e.g., falls, abnormal inactivity indicating potential medical issues).

Research: Investigate current technologies in Wi-Fi signal processing and motion detection. Focus on literature and existing projects that discuss Channel State Information (CSI) analysis for human movement and respiration rate monitoring.

Design System Architecture: Decide on the architecture of the system. This includes the selection of Wi-Fi devices for CSI data collection, types of motion sensors, and how data from these sources will be integrated.

Phase 2: Hardware Setup

Select Wi-Fi Devices: Choose Wi-Fi routers or access points that support CSI data extraction, such as those with an Intel 5300 chipset.

Acquire Motion Sensors: Obtain motion sensors that can be easily integrated with your system, like PIR (Passive Infrared) sensors for movement detection.

Install and Position Devices: Install the Wi-Fi devices and motion sensors in a simulated environment. Ensure optimal placement for maximum coverage and data accuracy.

Phase 3: Software Development

CSI Data Collection Tool: Implement or utilize existing tools for collecting CSI data from WiFi devices. Ensure the tool can capture data in real-time.

Signal Processing Module:

Develop a module to filter and process the CSI data, extracting features relevant to respiration and movement.

Apply techniques like wavelet transform for noise reduction.

Motion Data Processing: Create software to read and interpret data from motion sensors, distinguishing between normal movement and potential falls.

Integration and Analysis:

Fuse data from both Wi-Fi and motion sensors to analyze for emergency situations.

Implement simple algorithms to start, such as threshold-based detection for falls and basic pattern recognition for abnormal inactivity.

Alert System: Develop a mechanism to alert caregivers or emergency services when an emergency is detected. This could be via SMS, email, or an app notification.

Specific Methods we choose:

The fully connected deep neural network (DNN) architecture has been applied to Wi-Fi sensing by directly processing CSI features to classify instances as motion or non-motion. DNNs are capable of learning intricate relationships within the data, especially where a larger amount of labeled data is available for training. The large amount of training-labeled data prevents overfitting in the case of DNN. Using appropriate activation functions, regularization techniques, and optimization algorithms, DNNs can effectively handle motion detection tasks using CSI data.

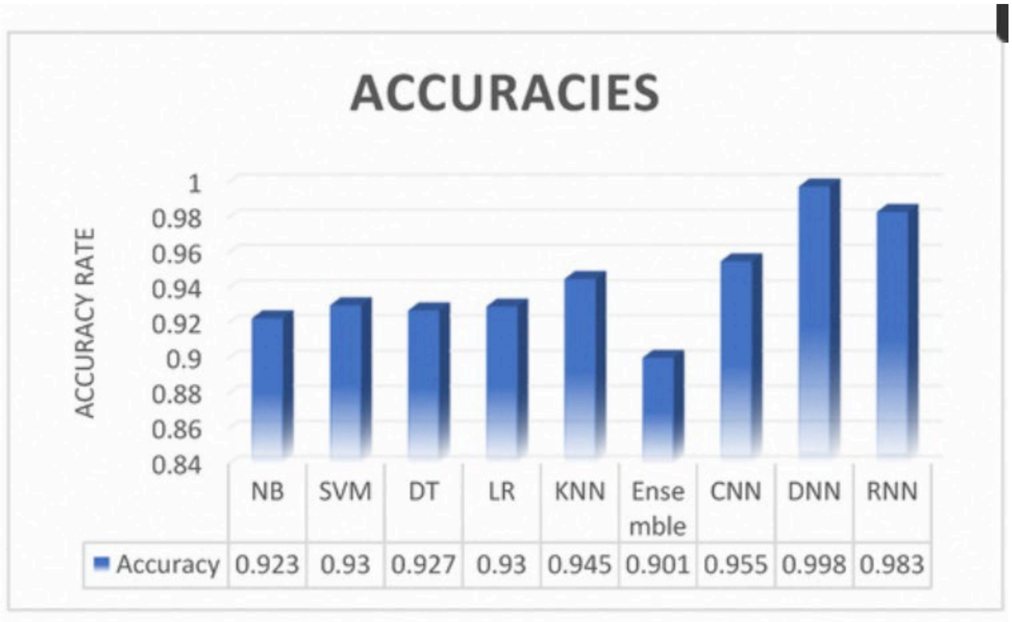

Figure 3: Accuracies

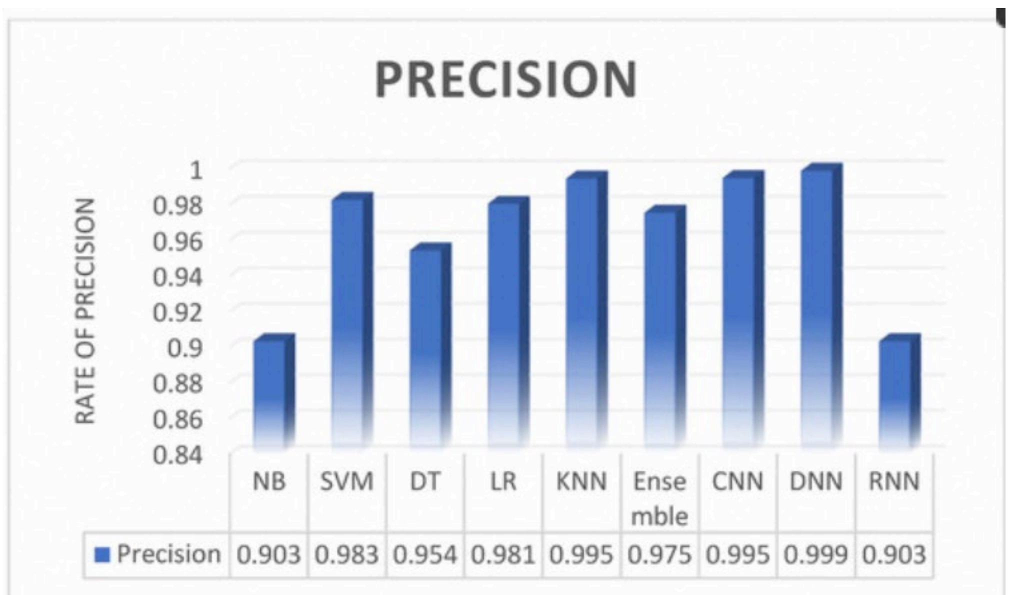

Figure 4: Precision

Phase 4: Testing and Evaluation

Simulated Testing: Begin with controlled tests to simulate emergency scenarios and evaluate the system's response. Adjust detection algorithms based on test outcomes.

Real-world Testing: If possible, conduct tests in a real environment with volunteers to simulate natural movement and emergencies, refining the system further based on this feedback.

Performance Evaluation: Assess the system's accuracy, speed of detection, and false positive/negative rates. Document any limitations or challenges encountered.

Phase 5: Iteration and Refinement

Review Feedback: Analyze data and feedback from testing phases to identify areas for improvement.

Refine Algorithms: Based on feedback, make necessary adjustments to the signal processing and emergency detection algorithms to improve accuracy and reliability.

User Interface Improvements: Enhance the alert system and user interface based on user feedback, ensuring it's intuitive and effective.

Documentation: Compile detailed documentation on the system's design, implementation process, and operation guidelines.

2.7.4 Additional Considerations

Collaboration: Work in teams, dividing tasks based on individual strengths, to ensure efficiency and learning opportunities for all members.

Ethics and Privacy: Consider the ethical implications and ensure privacy protection for all data collected and processed.

Budget Management: Keep track of expenses related to hardware purchases and any software licenses to stay within your project budget.

2.8 Requirements and Verification

2.8.1 Wi-Fi Source Subsystem Analysis

Requirements	**Verification**
Frequency and Channel Requirements Requirement: The Wi-Fi device must operate on the 5 GHz frequency band to align with the operational frequency. Quantitative Detail: The device should support configuration for at least 30 subcarriers, essential for compatibility with UT-HAR and Widar datasets. For adherence to NTU-HAR requirements, the capability to handle up to 114 subcarriers is necessary. Moreover, specific channel operation, such as channel 165 (5.825 GHz) as indicated for the Widar dataset, must be configurable.	Frequency and Channel Verification Equipment: Utilize a spectrum analyzer or Software Defined Radio (SDR) for direct measurement. Procedure: Configure the device to operate on the specified channels and frequencies. Measure and record the frequency and subcarrier distribution using the SDR or spectrum analyzer. Result Presentation: Graphical representation of the frequency spectrum and subcarrier allocation.
Bandwidth Requirements Requirement: Essential bandwidth configuration of 40 MHz under the 5 GHz frequency band to	Bandwidth Verification Equipment: Network analysis tools (e.g., Wireshark).

meet the specifications listed for the NTU-HAR dataset. Quantitative Detail: The device must provide an option to select and operate under a 40 MHz bandwidth, ensuring the collection and analysis of Wi-Fi signals are conducted under precise and stipulated conditions.	Procedure: Set the device to the 40MHz bandwidth mode. Use network analysis tools to capture and analyze the Wi-Fi traffic, verifying the operational bandwidth. Result Presentation: Summary report detailing bandwidth utilization and configuration adherence.
AP Support Requirement: The system should integrate seamlessly with at least two TP-Link N750 APs or equivalent models to establish a network environment as per the NTU-HAR dataset's setup. Quantitative Detail: Ensure the Wi-Fi device's compatibility with specific Access Point (AP) models and support simultaneous connections, facilitating a robust network framework for data collection.	AP Support Verification Equipment: TP-Link N750 APs or equivalent, multiple Wi-Fi enabled devices. Procedure: Connect the Wi-Fi device to the APs set in the required configuration. Test the stability and throughput with multiple connected devices under various conditions. Result Presentation: Performance metrics report, including throughput rates, connection stability, and device compatibility confirmation.
Hardware Interface and Compatibility Requirement: Provision of comprehensive hardware interface options, including USB, Ethernet, or Wi-Fi, to ensure seamless connectivity with a variety of operating systems (Windows, Linux, macOS) and programming environments. Quantitative Detail: Define clear compatibility metrics and interface standards (e.g., USB 3.0, Ethernet 10/100/1000 Mbps) to facilitate straightforward integration into the existing data collection and analysis infrastructure.	Hardware Interface and Compatibility Verification Equipment: Various computing platforms with different OS. Procedure: Connect the WiFi device using its provided interfaces to the computing platforms. Test compatibility through data transmission tasks. Result Presentation: Compatibility matrix and performance analysis across different platforms.
Environmental Adaptability Requirement: The Wi-Fi device must exhibit high performance and reliability across diverse environments – indoor, outdoor, varying temperatures, and humidity levels. Quantitative Detail: Establish operational parameters, such as operating temperature range (-10°C to 50°C) and humidity tolerance (10% to 90% non-condensing), ensuring device resilience and consistent data collection quality across environments.	Environmental Adaptability Verification Equipment: Environmental test chamber or equivalent setup. Procedure: Operate the device within an environmental test chamber set to varying conditions. Monitor performance and data integrity. Result Presentation: A detailed report highlighting performance metrics under each tested condition.
Data Precision and Stability Requirement: Achieve high precision and stability in data collection across different environmental and operational conditions, crucial for accurate behavior recognition and analysis.	Data Precision and Stability Verification Equipment: Standardized test environments and data analysis software.

Quantitative Detail: Set specific performance metrics, like a maximum data variance threshold of ±5% under predefined conditions, to ensure reliability and accuracy in data capture and processing.	Procedure: Collect data using the Wi-Fi device under controlled conditions. Analyze the data for precision and stability metrics. Result Presentation: Statistical analysis report, including variance and error rates, with comparisons against predefined performance thresholds.

2.8.2 Wearable Device Subsystem Analysis

Requirements	**Verification**
Accelerometers should have a range of at least ±2 g, gyroscopes should have a range of at least ±360°	We will use the MCU6050 IMU sensor and test it, while it is fixed on the wrist and other parts of the test personnel in the normal adult male standard larger test movements and real-time data recording to determine the required maximum range of range maintains a more stable condition.
According to the need to output about 3.3 V (3 V - 3.6 V) and 5V (4.7 V- 5.3 V) more stable voltage. Output current no higher than 1 A.	A multimeter is used to measure the battery and power supply module before official operation under simulated real load conditions, and the overall circuit is regularly monitored during the test to ensure that the power supply module maintains a more stable condition.
1. Delays due to data processing operations should be less than 200 ms. 2. The packet loss rate for data transmission over the Wi-Fi protocol should be no more than 1 per cent.	1. After writing the data processing code on the finished processor, we try to calculate the time required using the timing-related libraries in the environment, and if the results do not meet the requirements, we evaluate the parts that must run on that processor and transfer the code that can be moved to the processor part to reduce the transfer latency After the subsystem docking test, we will conduct a transmission stability test of no less than 2 hours, where the tester wears a wearable device to move around the room and simulate a normal life situation, and the processor subsystem part will be added in advance to monitor the packet loss rate and stability and alert the police if it exceeds the stipulated limits.

2.8.3 Detector Subsystem Analysis

Requirements	Verification
Processor for Signal Processing: A Raspberry Pi 4 or a computer with sufficient processing power to handle real-time signal processing and analysis. Implementation Steps: Signal Acquisition: Set up the Wi-Fi router in a central location within the environment to ensure broad coverage. Use software tools like Linux 802.11n CSI Tool on the receiver to capture CSI data from the Wi-Fi signals. Signal Preprocessing: Implement a high-pass filter using Python or MATLAB on the received CSI data. This can be done by designing a Butterworth filter with scipy.signal.butter in Python, specifying the high-pass frequency cutoff according to the expected minimum movement frequency of humans. Feature Extraction: Write scripts to calculate amplitude variance, phase change rate, and Doppler shift frequencies from the preprocessed CSI data. Use numpy and scipy libraries in Python for efficient computation	Conduct experiments simulating various human movements within the coverage area of the Wi-Fi signals. Capture the CSI data and use a spectrum analyzer software or a MATLAB tool to analyze the frequency and phase accuracy of the processed signals.
Noise Reduction: Implement a low-pass filter in the data processing script to remove unwanted high-frequency noise from the sensor data. The filter can be implemented similarly to the Wi-Fi signal preprocessing step. Feature Identification: Develop algorithms to identify features indicative of a fall from the motion sensor data, focusing on parameters such as peak acceleration and impact duration.	Perform controlled drop tests with dummies equipped with motion sensors to simulate falls. Record the sensor data and analyze it to verify the accuracy of feature identification and the effectiveness of noise reduction.
Model Training: Preprocess the collected Wi-Fi and motion sensor data and split it into training and testing sets. Use scikit-learn to train SVM models on the Wi-Fi data and Random Forest models on the motion sensor data. Optimize the models using grid search for hyperparameters. Ensemble Integration: Implement a weighted voting system in Python where each model's vote is weighted by its accuracy on a validation set.	Test the ensemble method on a diverse dataset that includes both fall and non-fall scenarios not seen during training. Use k-fold cross-validation to ensure the model's generalization capability and document the system's performance in terms of sensitivity, specificity, and false alarm rate.

The ensemble method should output a fall detection result based on the combined votes. Fall Detection: Integrate the ensemble method into the system's central processing unit. The system should analyze incoming data in real-time and trigger an alert if a fall is detected.	
Programming the Microcontroller: Write a script for the microcontroller that sends a predefined message to the Bluetooth module whenever the ensemble method detects a fall. This message can include details like the time of the fall and the location if known. Developing the Notification App: Create a simple mobile application that listens for Bluetooth messages from the paired Bluetooth module. Upon receiving a message, the app should display a notification, sound an alarm, or even send an SMS to a predefined contact list, depending on the severity of the alert and user preferences.	Conduct extensive testing to ensure the Bluetooth module reliably connects to the notification device and that the alert system activates correctly under fall conditions. Simulate various environments and distances to ensure the system's robustness.

Cross-Validation:

- **Objective**: To ensure the model's robustness and its ability to perform consistently across different datasets.
- **Methodology**: Application of k-fold cross-validation techniques to the fall detection algorithms, using both the simulated and real-world data to evaluate model consistency and generalizability.[6]

Expected Results for Testing Metrics:

- **Sensitivity and Specificity**: These metrics will be closely monitored to ensure the system maintains high accuracy in fall detection (targeting >85% sensitivity) and minimizes false positives (aiming for >90% specificity).
- **False Alarms**: The number of false alarms will be recorded and analyzed. The goal is to reduce false alarms to less than one per day per test subject in a real-world testing environment.

3 Cost and Schedule

3.1 Cost

Our fixed development cost is ￥40/hour for four people for 10 hours per week.[7] We believe that

Within the semester (10 weeks), we will have completed all the final design, so the estimate of our labor cost is:

$$\frac{￥40}{\hbar r} \times \frac{10\hbar r}{wk} \times 10wk \times 4 = ￥16000$$

Our parts and manufacturing prototype costs are estimate as ￥749.64 each:

Parts	Cost (Prototype)
Breadboards, Dupont cables, data cables, etc. (Taobao; generic)	￥20
IMU sensor module (Taobao; MPU6050)	￥10.4
Wi-Fi module (Taobao; ESP8266)	￥36.5
Battery box and power supply module (Taobao; 18650/RunesKee)	￥26.24
Wi-Fi router (Taobao; TP-LINK AX1500)	￥159
Microcontroller (Taobao; Raspberry PI 5)	￥479
Bluetooth module (Taobao; HC-02)	￥18.5

4 Ethics & Safety

Our Emergency Detection System for Elderly Care is designed with a strong commitment to ethical standards and safety, drawing guidance from the IEEE Code of Ethics and the ACM Code of Ethics. Key ethical considerations include the protection of privacy and confidentiality, as outlined in the ACM Code, and the imperative to avoid harm, a fundamental aspect of the IEEE Code.[6] We prioritize these ethical principles by implementing data encryption, anonymization, and employing system designs that minimize the risk of false alarms and missed emergencies.

Proactive measures to avoid ethical breaches include regular reviews by an ethics board, comprehensive team training on ethical conduct, and strict data protection measures. Potential safety concerns, such as device malfunction and data breaches, will be mitigated through rigorous testing, the development of clear emergency response protocols, and advanced cybersecurity measures.

Our approach ensures that the Emergency Detection System not only enhances the safety and care of the elderly in care facilities but does so with utmost respect for their dignity and privacy, embodying the principles of ethical responsibility and safety compliance.

References

[1] Department of Ageing and Health, "National Development Bulletin on Ageing 2020," 15 10 2021. [Online]. Available:

[2] “Older persons invisible and vulnerable in emergencies,” OHCHR. Accessed: Mar. 10, 2024. [Online]. Available: https://www.ohchr.org/en/stories/2019/10/older-persons-invisible-and-vulnerable-emergencies

[3] L. Low, D. Lee, and A. Chan, “An exploratory study of Chinese older people’s perceptions of privacy in residential care homes,” Journal of advanced nursing, vol. 57, pp. 605–13, Apr. 2007, doi: 10.1111/j.1365-2648.2006.04116.x.

[4] J. Hung, “Smart Elderly Care Services in China: Challenges, Progress, and Policy Development,” Sustainability, vol. 15, no. 1, Art. no. 1, Jan. 2023, doi: 10.3390/su15010178.

[5] N. Bao et al., “The Intelligent Monitoring for the Elderly Based on WiFi Signals,” in Advances in Multimedia Information Processing – PCM 2017, B. Zeng, Q. Huang, A. El Saddik, H. Li, S. Jiang, and X. Fan, Eds., Cham: Springer International Publishing, 2018, pp. 883–892.

[6] J. Hung, “Smart Elderly Care Services in China: Challenges, Progress, and Policy Development,” Sustainability, vol. 15, no. 1, Art. no. 1, Jan. 2023, doi: 10.3390/su15010178.

[7] IEEE. “”IEEE Code of Ethics”.” (2016), [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html (visited on 02/08/2020).

[8] “Undergraduate Earnings Analysis Book Database.” Accessed: Mar. 26, 2024. [Online]. Available: https://www.pishu.com.cn/skwx_ps/initDatabaseDetail?siteId=14&contentId=14490234&contentType=literature&type=%25E6%258A%25A5%25E5%2591%258A

[9] Vellas, B., Cayla, F., Bocquet, H., De Pemille, F., & Albarede, J. L. (1987). Prospective study of restriction of activity in old people after falls. Age Ageing, 16(3), 189-193.