CAPITAL UNIVERSITY OF SCIENCE AND TECHNOLOGY, ISLAMABAD

An Invasive Approach for Fall Detection using Wearable and Non-wearable Devices

by

Zahra Binte Irshad

A thesis submitted in partial fulfillment for the
degree of Master of Science

in the
Faculty of Computing
Department of Computer Science

2025

My dissertation work is devoted to My Family, My Teachers, and My Friends. I have a special feeling of gratitude for My beloved parent my brother and my sisters. Special thanks to my supervisor whose uncountable confidence enabled me to reach this milestone.

Capital University of Science & Technology
CUST
Pakistan

CERTIFICATE OF APPROVAL

An Invasive Approach for Fall Detection using Wearable and Non-Wearable Devices

by
Zahra Binte Irshad
(MCS203022)

THESIS EXAMINING COMMITTEE

S. No.	Examiner	Name	Examiner Organization
(a)	External Examiner	Dr. Mussarat Yasmeen	CUI Wah Campus
(b)	Internal Examiner	Dr. Mohammad Masroor Ahmed	CUST, Islamabad
(c)	Supervisor	Dr. Nayyer Masood	CUST, Islamabad

Dr. Nayyer Masood
Thesis Supervisor
January, 2025

Dr. Abdul Basit Sidddiqui
Head
Dept. of Computer Science
January, 2025

Dr. Muhammad Abdul Qadir
Dean
Faculty of Computing
January, 2025

Author's Declaration

I, **Zahra Binte Irshad** hereby state that my MS thesis titled "**An Invasive Approach for Fall Detection using Wearable and Non-wearable devices**" is my own work and has not been submitted previously by me for taking any degree from Capital University of Science and Technology, Islamabad or anywhere else in the country/abroad.

At any time if my statement is found to be incorrect even after my graduation, the University has the right to withdraw my MS Degree.

Zahra Binte Irshad
Registration No: MCS203022

Plagiarism Undertaking

I **Zahra Binte Irshad** solemnly declare that research work presented in this thesis titled “**An Invasive Approach for Fall Detection using Wearable and Non-wearable Devices**” is solely my research work with no significant contribution from any other person. Small contribution/help wherever taken has been duly acknowledged and that complete thesis has been written by me.

I understand the zero tolerance policy of the HEC and Capital University of Science and Technology towards plagiarism. Therefore, I as an author of the above titled thesis declare that no portion of my thesis has been plagiarized and any material used as reference is properly referred/cited.

I undertake that if I am found guilty of any formal plagiarism in the above titled thesis even after award of MS Degree, the University reserves the right to withdraw/revoke my MS degree and that HEC and the University have the right to publish my name on the HEC/University website on which names of students are placed who submitted plagiarized work.

Zahra Binte Irshad
Registration No: MCS203022

Acknowledgement

Say, ”He is Allah, [who is] One. Allah, the Eternal Refuge. He neither begets nor is born. Nor is there to Him any equivalent.” Al-Quran [112:1-4]. I would like to say Alhamdulillah, for blessing me, with the strength to finish this work. My heartfelt thanks to my mother for her love, prayers, and everything that I need. A special thanks to my father. Last but not least are my friends, with whom this journey becomes a garden full of flowers for me. Special thanks to my esteemed supervisor Dr. Nayyer Masood for the support, for his assistance, inspiration, and guidance in the field of research. I have learned not only research under his supervision but also his attitude towards others. Sir ”You are one of those persons in my life, whom I love to follow”. May Allah shower his countless blessing upon all of you, JazakAllah

Zahra Binte Irshad

Abstract

Human fall is an issue that effects especially to the eldery people.Human Fall results in the permanent disability of the person or even in the person's death. It can cause functional disability in elderly and a significant impact on his mobility, independence and life Quality. Fall system is an intelligent system for fall detection of elderly people in an indoor environments that takes the advantage of Internet of Things and machine learning Algorithms ensembling .Fall is detected through wearable and non-wearable devices. Both the approaches have there own pros and cons but wearable devices stand the most because of their continuously monitoring and tracking of human motion capability. Disadvantage of wearable devices is this that they require continuous charging and person's infeasibility because of wearing it on a body. In this research we are comparing invasive and noninvasive devices with each other and concluding that which devices perform better ad give better results. For wearable devices we have collected our own data through Samsung galaxy watch6 on which accelerometer and gyroscope sensors are embedded. These sensors send the data with a time stamp value to the raspberry Pi. We have deployed the trained model on raspberry Pi. We have dataset for non-wearable devices. This data is also collected from accelerometer and gyroscope. We are comparing the results of "Long short-term memory model" results on wearable and non-wearable dataset. Results show that LSTM performs better on Wearable dataset. It is giving 92 percent of accuracy.

Contents

Author's Declaration iv

Plagiarism Undertaking v

Acknowledgement vi

Abstract vii

List of Figures xi

List of Tables xii

Abbreviations xiii

1 Introduction 1
- 1.1 Background . . . 1
  - 1.1.1 Significance of Fall Detection . . . 1
  - 1.1.2 The Growing Problem of Falls . . . 2
- 1.2 Elderly People . . . 3
  - 1.2.1 Impacts of fall on elderly people . . . 3
- 1.3 Remote Health Monitoring . . . 4
  - 1.3.1 Wearable and Non-wearable Devices . . . 5
  - 1.3.2 Impact of IOT devices . . . 5
- 1.4 Comparison of Wearable and Non-Wearable devices . . . 6
  - 1.4.1 Challenges and Solutions of Health Monitoring System . . . 7
    - 1.4.1.1 User Acceptance . . . 7
    - 1.4.1.2 Technical Issues . . . 7
    - 1.4.1.3 Cost consideration . . . 8
    - 1.4.1.4 Integration with Artificial Intelligence (AI) . . . 8
    - 1.4.1.5 Advancements in Wearable Technology . . . 8
    - 1.4.1.6 Enhanced Connectivity . . . 8
  - 1.4.2 Research Gap . . . 9
  - 1.4.3 Problem Statement . . . 9
  - 1.4.4 Research Questions . . . 10
  - 1.4.5 Research Objective . . . 10
  - 1.4.6 Conclusion . . . 10

2 Literature Review 11

- 2.1 Background . . . 11
  - 2.1.1 Proposed Architecture for IoT Based Solution . . . 15
    - 2.1.1.1 Edge Layer:Real-Time Monitoring . . . 15
    - 2.1.1.2 Fog Layer"Localized Event Classification" . . . 16
    - 2.1.1.3 Cloud-Layer: System Evolution and Model Training 16
    - 2.1.1.4 Deployment Strategies . . . 17
    - 2.1.1.5 Iplementation . . . 17
    - 2.1.1.6 Edge to Edge computing . . . 17
    - 2.1.1.7 Prototype Development . . . 17
    - 2.1.1.8 Sensors and Data Collection . . . 18
  - 2.1.2 Data Handling Mechanism . . . 19
    - 2.1.2.1 Transmission and Cloud Integration . . . 19
  - 2.1.3 Feature Selection Process . . . 22
  - 2.1.4 Classification Results . . . 22
  - 2.1.5 Soft Fusion . . . 26

3 Proposed Methodology 30

- 3.1 Introduction . . . 30
- 3.2 Brief of Methodology . . . 31
  - 3.2.1 Proposed Methodology Diagram . . . 33
- 3.3 Dataset Description . . . 33
  - 3.3.1 Accelerometer . . . 34
    - 3.3.1.1 Working Principle . . . 35
  - 3.3.2 Gyroscope . . . 35
    - 3.3.2.1 Working Principle . . . 36
  - 3.3.3 Dataset for Non-Wearable Devices . . . 37
- 3.4 Feature Selection . . . 38
  - 3.4.1 RMS Feature . . . 39
  - 3.4.2 Formula of Sum vector magnitude on horizontal plane . . . 39
  - 3.4.3 Statistics Feature . . . 39
  - 3.4.4 Orientation(Angle)Feature . . . 39
  - 3.4.5 Jerk(Rate of Acceleration change) Feature . . . 40
  - 3.4.6 Dataset for non-wearable devices . . . 40
- 3.5 Experimental Setup . . . 41
  - 3.5.1 Hardware Setup . . . 42
- 3.6 Fall Detection using Wearable Devices . . . 42
  - 3.6.1 Devices Description . . . 42
    - 3.6.1.1 Raspberry Pi . . . 43
- 3.7 Experimentation . . . 45
  - 3.7.1 Dataset for Wearable Devices . . . 46
    - 3.7.1.1 Sensor Data . . . 46
    - 3.7.1.2 Label Data . . . 46
- 3.8 Long Short-Term Memory Model Detailed Overview . . . 48
  - 3.8.1 LSTM Architecture . . . 48
    - 3.8.1.1 Memory Cell . . . 48
    - 3.8.1.2 Gates . . . 48
    - 3.8.1.3 Forget Gate . . . 49

3.8.1.4 Input Gate 49
3.8.1.5 Output Gate 49
3.8.2 Advantages of LSTM 49
3.8.3 Limitations of LSTM 49
3.9 Proposed Methodology 50
3.10 Data Collection 50
3.10.1 Data Collection for Training the Model 51
3.10.2 ML Models Training for Wearable Devices 52
3.11 Passing Data to the Pi 52
3.12 Passing Data to the Model 53
3.12.1 Model Testing For Wearable Devices 54
3.12.2 Model Implementation 54
3.12.3 Scaling Features 54
3.12.4 Splitting Data 54

4 Results and Evaluation **56**
4.1 Introduction 56
4.1.1 Metrices Used for Evaluation 57
4.1.2 Accuracy: 57
4.1.3 Precision 57
4.1.4 Recall 58
4.1.4.1 F1-Score 58
4.1.5 Results for Non-Wearable Devices 58
4.1.6 Results on Wearable Dataset 59
4.1.7 Comparison of the Base Paper Results with LSTM 62

5 Conclusion and Future Work **63**

Bibliography **64**

List of Figures

3.1 Proposed Methodology Diagram 33
3.2 Accelerometer 35
3.3 Gyroscope Sensor 36
3.4 Dataset for Non Wearable Devices Without Feature Selection 37
3.5 Non wearable Devices Dataset With Feature Selection 40
3.6 Raspberry Pi 43
3.7 Samsung Watch Galaxy 6 44
3.8 Dataset For Wearable Devices 47

4.1 Confusion Matrix 57

List of Tables

2.1 Literature review table 23

4.1 Parameters for Wearable and Non-Wearable Devices 62

Abbreviations

IoT	Internet of Things
LSTM	Long Short-term memory
RNN	Recurrent Neural Network

Chapter 1

Introduction

1.1 Background

Human fall can occur at any age but it is common in elderly people. Every year more than 37 million falls occur. Approximately 61 percent of falls occurred indoors, which caused around 10,000 deaths[1]. A human fall occurs because of a loss of balance or being unable to maintain it. It is an incident where a person suddenly descends to the floor or a ground without any apparent loss of consciousness [2]. Falls can lead to increased dependency, inability to independently perform certain tasks, confusion, immobilization, and depression, all of which can adversely impact an individual's daily activities. Falls can be categorized in accidental falls. Falls that are both identifiable and foreseeable. Fall risks are not the same as falls caused by accidents or falls that are unpredictable that arise from predictable external factors.

1.1.1 Significance of Fall Detection

It is critical to take quick action after a fall and to give health support services the appropriate notice in order to reduce the harmful effects of human falls. It becomes more critical in case of an elderly or weak person who is not able to stand by himself after the fall. Such a person needs another person who is continuously monitoring this elderly/weak person and who can extend the help when and if required. However, it is not always possible to monitor an elderly or weak person all the time. An automated fall detection system can be of great help in this situation as the system is monitoring the elderly 24 hours a day[3]. An automated system CAN monitor 24 hours a day, like our system, person puts on watch and our system monitors it and if the person falls, the system detects the fall and raises we have explained that message goes to the person registered as caretaker, the same message can go the doctor, ambulance and anywhere we want to the alarm by

sending message to the attendants or to the emergency services. This can also reduce the cost on healthcare as delayed fall detection creates increased demand for emergency services, hospital admissions, and rehabilitation which can lay burden on healthcare resources and lead to higher costs. For example, the treatment of fall-related injuries often requires multiple medical interventions, including emergency care, surgery, and long-term rehabilitation. The societal impact extends beyond healthcare costs. Falls can reduce the independence of elderly individuals, leading to a lower quality of life and increased reliance on caregivers. This shift can affect social dynamics, as elderly individuals may experience isolation or reduced social interaction due to their injuries.

1.1.2 The Growing Problem of Falls

As people age, their physical abilities decline, and their bodies become more used to injuries resulting from falls because they fall more frequently, so it becomes part of their life/routine. According to data from the World Health Organization (WHO), falls are a leading cause of both fatal and non-fatal means that do not cause death, but still people get injured injuries among those over the age of 65. This demographic shift highlights the urgent need for effective fall prevention and detection strategies. The Change in life pattern, age group, physical condition needs all that the consequences of falls can be severe. For elderly person, a fall can lead to fractures, head injuries, and in some cases, death. The risk of these outcomes increases with age, as older adults often have weaker bones, reduced muscle strength, and impaired balance. The emotional and financial impact on families is also significant, as they may need to provide extensive care for their injured loved ones, often resulting in increased stress and financial strain.

1.2 Elderly People

The aging population worldwide is contributing to an increasing number of elderly people who can have risk of experiencing fall. Elderly falls refer to a situation where an older adult or citizen loses his balance and physically descends to the floor or any other surface [5]. Growth in the number of elderly people is increasing day by day elderly people are increasing in number[6]. This is making fall a major issue in elderly people. How ever monitoring their health level/well-being level in the daily routine can prevent fall in elderly people. Several solutions are presented

to detect fall in elderly people which include wearable devices and non-wearable devices. The Internet of Things (IoT) technology plays a big role in tracking and encouraging people's movements and actions. The architecture of IoT (Internet of Things) integrates embedded electronics, sensors, software, and connectivity, often referred to as the "Internet of Everything."This interconnected network allows objects to exchange data through connections between PCs, person-to- person, person-to- things, human-to-human, and things-to-things. With the expansive reach of the internet, this data can be accessed from any location at every time. Consequently, health data can be monitored from anywhere and accessed via IoT.

1.2.1 Impacts of fall on elderly people

In an elderly people fall has become a growing problem. Elderly people are more susceptible to falls. A total of 61 million people over the age of 65 in 2004 has been estimated to rise to 2 billion by 2050 [7]. Severity is aslso caused by Falls.Falls are a leading cause of unconscious injury that is and death of those who are over 65, making them a major public health concern. Fall causes psychological stress and severity of head-trauma [8].Falls have a serious effect.Falls can have lasting consequences not only for the elderly individuals involved but also for their families, healthcare systems, and society.

The world's elderly population is facing a significant threat: falls [9]. As we age, our bodies become more susceptible to injury and as a result falls can lead to serious consequences, including fractures, head injuries, and even death. This vulnerability is compounded by the fact that the global population is aging rapidly, means that larger proportion of people are at risk. The impact of falls does not only effects individuals, but it effects beyond individuals also.Families face emotional and financial burdens caring for injured loved ones. This issue is increasing due to the global rise in the aging population. To address this, there's a growing need for reliable fall detection systems. While individual sensors like wearables and cameras exist, their accuracy is not good because of

the high rate of false alarms. While individual sensors like wearables and cameras exist, they often suffer from inaccuracies, particularly high false alarm rates. There are various aspects of fall detection systems, including data collection, transmission, analysis, and the crucial considerations of security and privacy. By reviewing existing research and benchmark datasets, this work aims to provide a comprehensive literature review on fall detection and propose a solution to the issue. This highlights the growing concern of falls among the elderly population. The impact extends beyond the individual, affecting families, healthcare systems, and society. Overview of the field, highlighting advancements and areas requiring further development [10]. In the past years so much has been done in detection of fall to make the accuracy better and increase the prediction of different systems. This has decreased the chances of false alarm.

1.3 Remote Health Monitoring

The number of patients receiving remote healthcare at home has increased, facilitated by the most used and various nature of mobile devices designed for remote care. The continuous development of the Internet of things and six axis acceleration sensor technology can provide real-time intelligent remote monitoring which is actually target of this research, that is, monitoring an elderly in real-time as he/she is performing his/her tasks. Remote monitoring for elderly falls using intelligent video technology can automatically recognize various situations of users on the screen through real-time video. As an important tool to monitor the activities and posture of the elderly, acceleration sensors can monitor the body posture, acceleration and Angle of the elderly people in real time by detecting body movement and posture changes [11]. It share health information which offers patients a safer and better quality of life. In addition to the specialists, guardians can obtain a comprehensive medical history of the patient. Providing information to guardians enhances the responsibility and accountability of hospital management units, alleviating mental stress for the caregivers [12]. The unintentional landing of a body on the ground, floor, or another lower level is known as fall [1,3].

1.3.1 Wearable and Non-wearable Devices

To address the growing issue of falls in elderly, there has been significant progress in the development of fall detection technologies . Traditional methods, such as wearable sensors and camera-based systems, have been employed to monitor and detect falls. Wearable and non-wearable devices are those that a patient can wear around his body. Our life is improved by sensors that are wearable technologies and IoT have strongest capability to make our lifestyles better by providing us healthcare monitoring systems so that these systems can track our health and physical activities. Accelerometer is a sensor starts mentioning the devices that are used to measure the physical activities that is used to measure the acceleration of a body. Gyroscopes (a sensor) are used to measure the velocity of the body. Doppler sensors (sensor) are used to measure the sound of the body. Non wearable devices are those that are not in contact with patients' bodies, non-wearable devices for fall detection are related to videos obtained through ambient camera.Each of the solutions has their own advantages and disadvantages. Wearable devices can track the continuous tracing of users. The issue with wearable devices is that a person has to wear them all the time and/moreover wearable devices need charging. However, solutions that are non-wearable can tackle these issues by making it possible to monitor the spaces without the need of people to have these devices on them. However non-wearable devices have a disadvantage that they have limitation of space, they are not capable to detect human body movement or falls beyond the specific monitoring of the radius.

1.3.2 Impact of IOT devices

The impact of IoT devices is getting advanced in all fields of life, especially in healthcare. The function of IoT is becoming more dominant when it is combined with the features of mobile. It expands the role of IoT in healthcare area by a large support in mobile health form [13]. The system can collect data related to patient's health and this data comes from multiple sensors. After this process the collected data is transferred to a server which is working remotely for analysis of data and result is displayed which is in real-time [14].

SAFE-RH `https://safe-rh.eu/`is a remote health monitoring system that offers health monitoring facilities in Pakistan's rural areas. By using Sensing, Artificial Intelligence and Edge networking, it aims to provide health facilities through rural health monitoring. There are three target groups in SAFE-RH, that is, maternal, infants and elderly. The work done in this research is related to elderly target group.The major objective is to lower the risk of mortality for women and children, promptly handle maternity-related problems in such areas, and keep an eye on elderly. By utilizing remote health monitoring and healthcare organizations can lessen these difficulties and burden their patients face, such as the need to travel for specialty care-related transportation problems. Additionally, remote heath monitoring can enhance systematic coordination, monitoring and responsiveness. The aim of SAFE-RH in Pakistan is to provide health care facilities for patient of rural areas such as:

i. Minimizing frequent visits and admission to hospitals in case of serious issues. ii. Minimize cost and time for consultation. iii. Ease for medical advice. Reduced death rate due to lack or delay of medical help.

Different computing domains have been explored in this domain such as SVM known as Support Vector Machine, ANN known as Artificial Neural Network, CNN known as Convolutional Neural Network for fall detection in elderly people.

1.4 Comparison of Wearable and Non-Wearable devices

Wearable sensors, including accelerometers and gyroscopes, can track changes in movement and detect falls based on sudden shifts in position. However, these devices can suffer from issues such as discomfort, limited battery life, and inaccuracies in detecting all types of falls. Camera-based systems offer continuous monitoring and can detect falls in various environments. Despite their potential, these systems often face challenges related to privacy concerns and limitations in detecting falls in different settings/situations examples are given ahead, such as when the individual is out of view or in low-light conditions. To address this problem the problem of fall detection, (WHICH PROB) one fall detection approach is used which involves the use of wearable sensors. These devices, which can be worn as belts, pendants, or wristbands, typically include accelerometers and gyroscopes that monitor movement patterns. When a fall occurs, the sensors detect sudden changes in position and can trigger alerts to

caregivers or emergency services. While wearable sensors offer valuable real-time data, they face challenges such as user discomfort, battery life limitations, and the potential for false alarms if the device does not accurately differentiate between falls. In the recent years the term “Internet if Things” has covered the entire world of communication and information[3]. In the age of smart computing, the technologies like (IOT) and sensors have made it possible to assess the wellness of people remotely.These IOT based devices and sensors have made it possible to assess the wellness of the people remotely. Systems that monitor activity are based on classification techniques. These techniques can differentiate between daily living activities and fall activity.

1.4.1 Challenges and Solutions of Health Monitoring System

The implementation of activity monitoring systems may face several challenges:

1.4.1.1 User Acceptance

Older adults/elderly people may have varying levels of comfort with technology. Providing training and support to users can help improve acceptance and engagement with activity monitoring systems.

1.4.1.2 Technical Issues

Some problems, such as sometimes device malfunctions or there are problems that are related to connectivity, can effect the reliability of activity monitoring systems. Regular maintenance and technical support are important to address these issues.

1.4.1.3 Cost consideration

The cost of implementing and maintaining activity monitoring systems can be a concern. Exploring cost-effective solutions and potential funding sources can help overcome financial barriers. Future Directions and Innovations The field of activity monitoring systems is continually evolving, with ongoing research and innovation driving advancements. Some potential future directions include

1.4.1.4 Integration with Artificial Intelligence (AI)

The integration of AI and machine learning algorithms can enhance the analysis of activity data, leading to more accurate predictions and personalized recommendations.

1.4.1.5 Advancements in Wearable Technology

Emerging technologies, such as smart clothing and advanced sensors, may offer new opportunities for monitoring and tracking health metrics.

1.4.1.6 Enhanced Connectivity

Problems of connectivity are addressed, such as 5G networks, can help in transferring the real-time data transmission and communication, more ever increasing the power of activity monitoring systems.

The increasing frequency of falls highlights the urgent need for effective strategies to prevent and detect these incidents to lessen and eventually eliminate their consequences. Innovations in technology, such as wearable sensors, camera systems, and the integration of various sensor data, show potential in improving fall detection and enhancing the safety of older adults. Moreover, proactive measures such as modifying home environments, implementing exercise programs, and providing educational resources are essential for addressing the fundamental causes of falls.

To effectively tackle the challenge of falls, a comprehensive strategy is required that comes technological advancements for increased awareness. By collaborating on these efforts, objectives we can significantly improve the quality of life for older adults and mitigate the broader societal impact of falls.

This research focuses on fall detection in elderly people using wearable device. This study will be helpful in healthcare systems to detect fall and reduce the risks of fall. It compares wearable and non-wearable devices for fall detection. We have collected the real time data for fall from a watch sensor. We are comparing different methodologies on wearable and non-wearable devices for fall detection and comparing the results that which is better technique for fall detection.

1.4.2 Research Gap

Following research gaps have been identified from the literature review:

Most of the detection approaches are based on using wearable devices and there are relatively less approaches for non-wearable devices. Moreover, there is no comparison between wearable and non-wearable devices. Wearable devices are mostly placed with the waste using some belt of other similar thing.

Wearing a belt for longer duration of time is not very convenient specially for the elderly people. We do not find much results regarding the real-time deployment of fall detection systems./There is no approach where wearable and non-wearable devices are being used at the same time.

1.4.3 Problem Statement

Fall is a phenomenon that results in a person coming to rest on the ground or lower level uncontrollably. Delay in knowing the fall in elderly people and when unattended the impact of this delay results in fatality or disability of the patient. Significance: Detection of fall accurately can help in saving the person from any disability, injuries or death. Objective: Previous studies mainly focus on fall detection with wearable devices whereas few approaches exist with non- wearable devices. The objective of this study is to compare these two approaches that highlights the relative merits and demerits of the two. It will the user to pick a better approach.

1.4.4 Research Questions

Following research questions have been addressed in this study:

RQ1: How can we capture data from sensors using wrist watch?

RQ2: Which one, weather invasive or non-invasive /wearable or non-wearable devices offer reliable services in generating highly desirable speedy response for saving precious human lives

1.4.5 Research Objective

The basic goal of this research is to make a model that is capable to classify the fall and activity of daily living to prevent the falls. In this study we can identify the impact of fall with wearable device inac- curacies s and non-wearable devices. The study focuses on how to prevent fall

1.4.6 Conclusion

Falls among elderly individuals present a significant and complex health issue impacting not only the individuals but also their families, healthcare systems, and society at large. The increasing frequency of falls highlights the urgent need for effective strategies to prevent and detect these incidents to lessen their consequences. Innovations in technology, such as wearable sensors, camera systems, and the integration of various sensor data, show potential in improving fall detection and enhancing the safety of older adults. Moreover, proactive measures such as modifying home environments, implementing exercise programs, and providing educational resources are essential for addressing the fundamental causes of fall.

Chapter 2

Literature Review

2.1 Background

In this section we give a knowledge of background work that relates to fall detection focusing on invasive and noninvasive device .This research proposes the architecture and manufacture a single Inertial measurement unit hardware device that can have acceleration and angular velocity signals. It establishes fall detection algorithm depends on deep learning model that can effectively classify falls, near falls and ADLs called DAG CNN. The accuracy of this algorithm is 99 percent.

Another study[15]proposes process an effective and better fall detection method that uses a methodology based on a killer heuristic optimized convolution neural network. Sensors thar use wearable devices consist of magnetometer, gyroscope and convolutional neural network are placed on the person's body in different positions. It uses PCA for dimensionality reduction. The gathered details are then used for different feature extraction for example minimum, mean, standard deviation, acceleration, and maximum. The features that are derived are analyzed in the 'n' different ways array and to select the more related features principle components are derived. During the data gathering activity 16 activities of daily living and information for 20 fall activities are detected.

The sensor is strategically positioned to capture a wide range of scenarios. These scenarios are categorized into falling actions and non-falling actions. Falling actions include a variety of scenarios such as front lying, front protecting lying, various directions like front right or back right, and recovery actions. Non-falling actions are more related to routine activities, such as lying on a bed, rising from bed, and sitting on a bed.

Acceleration Measurement and Calculation is used to effectively distinguish between falls and other activities, the sensor records acceleration data in three dimensions: x, y, and z. Then the total acceleration is computed.

For each recorded event, the analysis is conducted within a specific time window. A 2-second interval is considered before and after the point of interest to capture the dynamic changes in acceleration. The data within this interval is divided into 101 samples, allowing for a detailed examination of the acceleration patterns during the event.This study has done some feature extraction before they have implemented the model.

Feature Extraction is used to increase the ability to differentiate between falls and daily activities, various features are extracted from the acceleration data. These features include:

Mean Acceleration is used to get the average of the acceleration values over the sampled interval, to give the idea about the level of acceleration.

Maximum Acceleration is used to know the high values of acceleration and high intensity level Maximum acceleration feature is used.

Skewness is used to tell the asymmetry of the acceleration data distribution. A skewed distribution tells the unusual patterns in the acceleration data which are helpful for different types of moving activitties and also for activity of falling.

Variance measure tells how much the acceleration values deviate from the mean. High variance gives the idea that there is a high level of fluctuation in the acceleration data, which is realted in differentaiting between different types of activities.

Kurtosis measure tells the "tailedness" of the acceleration data distribution. Higher kurtosis values tells that more extreme deviations in acceleration, powerfully useful for identifying falls or other impactful events.

Autocorrelation tells the correlation of the acceleration signal with itself at different time lags. The first 11 values of autocorrelation are checked to understand the repeating patterns or periodicity in the acceleration data.

Discrete Fourier Transform (DFT) values the DFT is used for the conversion of the time-domain signal into the frequency domain, that gives the information about important frequencies in the acceleration data. The first five peaks in the DFT values are importantly examined to capture significant frequency components that might correspond to specific types of movements or fall

Another study[16] Previous fall detection algorithms have primarily relied on shallow machine learning, deep learning, or rule-based approaches. However, these methods often extracted features that were not highly relevant for accurate fall detection. This paper addresses the research gap by focusing on the extraction of meaningful time-series features that have not been previously used in fall detection datasets, specifically using accelerometer data.

To gain this, the raw acceleration data is converted into a signal using the signal magnitude vector. The goal is to hold cross-disciplinary time-series features, which have the ability to give more accurate fall detection. A large number of features are extracted from the data, necessitating the use of feature selection techniques to identify the most significant ones.

The mutual information selection algorithm was applied to determine the most important features from the extensive dataset. Additionally, the Pearson correlation feature was calculated to eliminate highly correlated data, ensuring that the selected features are independent and informative.

Upon reviewing various papers, it is evident that previous implementations of fall detection models have not been satisfactory in terms of performance, accuracy, and precision. These limitations underscore the need for a more robust approach that incorporates meaningful time-series features and advanced feature selection techniques.

By addressing these gaps, this paper aims to enhance the reliability and effectiveness of fall detection systems. The proposed method offers a new perspective on feature extraction and selection, which could lead to significant improvements in the accuracy and precision of fall detection algorithms.

Another study [17] studied these problems using a three-dimensional convolution neural network-based method for fall detection. To have a deep study in area of interest in each frame a LTSM is used. 3D CNN is trained on Sports dataset which is SPorts-1M with no examples of fall is to train the 3D CNN and then 3D CNN is combined with LSTM to train a classifier with fall dataset. Experiments have validated the proposed scheme on fall detection with 100%accuracy

Another study uses Fall detection approach[1].The approach presented in this paper is based on noninvasive devices. The sensors used in this technique are sound sensor, two accelerometer and a Doppler sensor. The devices have been laced on floor. The first sensors i.e sound sensor reads the sound signals. The second sensors accelerometer reads the x-dimension ,y-dimension and z-dimension signals and a third signal reads the vibration signals. These IoT devices are placed on floor of the room where they enable the real collection of the data that will give permission to the testing and classification of models. The data captured from these devices is passed through machine learning models which is Artificial Neural Network. The proposed IoT consist of three important layers that are used for computation purposes coexist and compete each other.

The edge layer with the hardware consists of the IoT devices is deployed in room. The IoT devices can predict off normal values and gather a time window of data. Fog Layer is used for sending the off-normal values to the classification of data. The edge layer will send the Fog layer will handle the classification of data. The IoT devices-based solution for fall detection using devices that are non-wearable and non-intrusive and a layered architecture that was prototype and two classification models. Model of the data allows the collection of the data that will allow the testing and validation of classification model. Model that is used is Artificial Neural Network. IOT device enable the real gathering of data that will allow the testing and classification of models. The proposed solution is an IoT based model based on IOT structure consist of three computational layers coexist and complete each other. The edge layer with the hardware composed the IOT device is deployed in room. The IoT device can detect off normal values and collect a time window of data. Fog Layer is used for sending the off-normal values the classification of data. The edge layer will send the Fog Layer will manage the classification of events The proposed approach is IOT device is designed to be placed in contact with the room's floor and accelerometer are used to check the floor vibration Both accelerometers are in the closed contact with the floor surface.

2.1.1 Proposed Architecture for IoT Based Solution

The proposed solution utilizes an advanced Internet of Things (IoT) architecture that incorporates three distinct computational layers working in concert to deliver a scalable and efficient system for environmental monitoring and fall detection. This architecture is designed to support the deployment of new nodes and handle varying levels of data processing, as illustrated in Figure 1. The system is structured hierarchically, resembling a tree where each node can have multiple branches or leaves, facilitating flexible and scalable solutions.

2.1.1.1 Edge Layer:Real-Time Monitoring

The basic layer of (IoT) device is edge-layer of the (IoT) device handels the real-time data collection and tracking of the data coming from the environment.The edge-layer comprised of(IoT) devices that contain variety of sensors and micro-controllers.These devices are placed in rooms for-example bed-rooms, hall-ways and bathrooms.The basic role of edge-layer is to continuously observe and collect data from the environment

The IoT devices at the edge-layer are designed to detect any deviations from normal conditions, which could indicate potential incidents such as falls. Upon detecting an off-normal event, these devices collect and store data within a local buffer. This buffer captures a time window that includes three critical segments of data: the period before the event, the duration of the event, and the aftermath of the event. By analyzing this time window, the system aims to accurately classify the nature of the off-normal event. The edge-layer is implemented using microcontrollers capable of reading sensor inputs and processing this data locally

2.1.1.2 Fog Layer"Localized Event Classification"

The next layer, known as the fog-layer, is used for the classification of events that are highlighted by the edge-layer devices. Once an off-normal event is detected and reported by an (IoT) device, the data, which includes the collected time-window, is transferred to the fog-layer. The fog-layer is strategically placed within the facility to minimize latency and avoid the need for external communication. In this layer, the classification of activities is performed using machine learning models implemented on micro-processors or single-board computers, such as the NVIDIA® Jetson Nano™ or the Raspberry Pi 4 Model B. The choice of hardware for this layer is important as it directly impacts the speed and accuracy of event classification. The fog-layer is responsible for handling data from multiple IoT devices at a same time. For events classified as exact falls, the fog-layer is assigned a tasl with alerting facility health supporters to initiate a response.Other than this the fog-layer maintains a record of all classified fall activities, which are subsequently verified by human operators. Verified events are then forwarded to the cloud-layer for further processing and model improvement.

2.1.1.3 Cloud-Layer: System Evolution and Model Training

The cloud-layer works as the central hub for system evolution and improvement. Its basic tasks include the training of new classification models and updating existing models based on the data received from the fog-layer. Given the cloud's important computational resources, it is well-suited for handling the heavy processing that is used for training and refining machine learning models. The cloud-layer also manages a push notification system to communicate alerts and updates to users and facility staff. Other than this, it operates a storage server for archiving anonymized fall activity data. This data is crucial for continuous system improvement, as it provides detai information that drive the development of more accurate and reliable classification models. Although the cloud-layer does not participate in real-time fall detection, it plays an important role in the overall functionality of the system by enabling ongoing enhancements and adaptations.

2.1.1.4 Deployment Strategies

The deployment of this (IoT) architecture includes the placing of edge-layer and fog-layer components directly within the facility to ensure real-time monitoring and immediate event classification. Meanwhile, the cloud-layer operates from a central location, servicing multiple facilities to leverage its extensive computational capabilities for model training and system updates.

2.1.1.5 Iplementation

The roles of edge-layer and fog-layer have been initially developed and tested using Python on standard desktop systems. However, for practical deployments, it is suggested to use micro-controllers for the edge-layer and micro-processors or single-board computers for the fog-layer to make optimal performance.Still the cloud-layer stays very important fr the improvement and adaption of the system continuously wich supports both the evolution of classification models and the management of fall activity data.

2.1.1.6 Edge to Edge computing

The edge-side (IoT) device described is integral to the solution's data collection strategy, functioning as the primary sensor unit. This device has undergone important development through two different prototypes, each using different micro-controllers but maintaining the same set of sensors.

2.1.1.7 Prototype Development

i Microcontroller: The first version of the device was built using the Arduino UNO R30, which consists of the Atmega328 microcontroller.

ii Data Transmission: Data was transmitted via a USB connection, which gives permission to be processed by a Python script designed to handle and classify fall data.

iii Microcontroller: The next iteration of the device used the ESP8266 micro-controller, specially the NodeMCU model. This upgrade was made to facilitate data transmission over a Wi-Fi network, which gives more flexibility and ease of deployment. The design can be easily adapted for future upgrades.

iv Data Transmission: The second prototype utilizes the Message Queuing Telemetry Transport (MQTT) protocol for transmitting data, a step up from the previous USB-based approach. MQTT is a lightweight messaging protocol that ensures efficient data communication, particularly in scenarios requiring real-time data transmission.

2.1.1.8 Sensors and Data Collection

Both prototypes are equipped with a consistent set of sensors to ensure comparable data collection capabilities:

i Analog Sound Sensor: Captures sound levels to detect relevant acoustic events.

ii MPU-9250 Accelerometer: Measures acceleration along multiple axes.

iii GY-521 Accelerometer: Another accelerometer used for vibration measurement, positioned to be in close contact with the floor.

iv HB-100 Doppler Sensor: Measures the Doppler shift in reflected signals to assess movement or changes in the environment. Data Handling Mechanism

v Data Acquisition: The sensors continuously gather data without interruption. The frequency of data collection is determined by the internal clock speed of the ESP8266 micro-controller.

vi Event Detection: To identify significant events, such as falls, the device uses predefined level thresholds. When these thresholds are exceeded, the device activates a buffer system to capture relevant sensor data.

vii Time Window Creation: Following an event trigger, the device creates a time window of approximately 700 milliseconds. This window includes data from a pre-trigger period and a post-trigger period, encompassing about 70 samples from each sensor.

2.1.2 Data Handling Mechanism

i Data Acquisition: The sensors continuously gather data without interruption. The frequency of data collection is determined by the internal clock speed of the ESP8266 microcontroller.

ii Event Detection: To identify significant events, such as falls, the device uses predefined level thresholds. When these thresholds are exceeded, the device activates a buffer system to capture relevant sensor data.

iii Time Window Creation: Following an event trigger, the device creates a time-window of approximately 700 milliseconds. This slot includes data from a pre-trigger period and a post-trigger period, having about 70 samples from each sensor.

2.1.2.1 Transmission and Cloud Integration

i. Data Transmission: The data which consists of off line events is the only data that is transmitted using the MTT protocol.This selective transmission ensures that only significant data is sent, optimizing network usage and the efficiency of processing.

ii. Threshold Adjustment: The thresholds used for detecting off-normal activities can be managed dynamically through cloud communication. This feature permits the real-time updates and fine-tuning based on ongoing data analysis and environmental changes.

This[18] proposes a architecture which uses edge computing where data is not sent to the cloud but wearable devices send data to a nearby edge device called laptop or mobile device for the analysis in a real-time. The sensor used in this approach is MetaMotionR. Laptop is an edge device that collects the data that is coming from the wearable sensors and preprocessed send it to edge devices when a magnitude gets higher than the predefined threshold. At this point data was analyzed using Tensor Flow where a model was trained using data, model classifies the data as fall or non- fall. The sensor is placed at the waist and 50Hz model was able to detect fall from the real-time data with 0.75 precision, 0,64 recall and 85.2.

According to[19] a system is proposed that can detect fall. A fall diagnosis system consists of 3 steps as prevention, prediction and detection. Internet of things (Iot) has given edge, Fog and cloud as an IoT layers in order to implement stages for all diagnosis system. For this purpose, IoT architecture involving smart sensors measure fall related signals and then machine learning algorithms detect events of fall and communication channel transformation information. Lot of components control all actions and data flow.

This study[18] presents an artificial vision-based algorithm low cost fall detector for smart homes. Detector makes a combination of several algorithms like subtraction of background, Alman Filtering and Optical Flow and give it as an input to a machine learning algorithm which gives high fall detection accuracy. Detection rate of greater than 96 is given by the tests thar are conducted on fall given videos.

Another study [19] studied these problems using a three-dimensional convolution neural network-based method for fall detection. To have a deep study in area of interest in each frame a LTSM is used. 3D CNN is trained on Sports dataset which is SPorts-1M with no examples of fall is to train the 3D CNN and then 3D CNN is combined with LSTM to train a classifier with fall dataset. Experiments have validated the proposed scheme on fall detection with 100% accuracy

Another study[20] uses a fall detection system. It has four main parts. A device that is wearable, a wireless communication network, a smart IoT gateway and cloud services. A decision tree model is built by using SisFall dataset. For data classification decision tree is used and it is the better methods for data collection and regression amongst many other algorithms as it has a non-parametric structure design which is suitable for large dataset. In this paper SisFall records are initially used to train the model. Model is trained on the the fall detected by the smart internet of things gateway Sisfall is the dataset that is publicly available for benchmarking and development of fall detection system. This dataset has 38 participants of 18 are elderly people. In this dataset used to make model includes 955 ADL records and 2865 falls records recorded by by accelerometer. Decision tree Model has trained on dataset, it classifies fall activity or as an ADL activity. The thickness of the branches shows the amount of data for training. The leaf node tells the prediction.To detect falls big data creates the local instance of the model. If results of the prediction is fall than emergency alert handler is warned by the system.

This paper[21] shows the frame work that is based on LSTM.LSTM is a type of Recurrent neural network that is used for modeling the data that is based on time series process. In this study LSTM is used for checking and classification purposes. In this work three wearable inertial sensors on waist, wrist and ankle and continuous activity streams from FMCW radar is collected as a data.The LSTM implements soft fusion method which means it concatenates feature from both sensors at each time stamp and it ensures that concatenated feature vector maintain temporal order. This LSTM fusion method is between wearable sensors data and radar. A hybrid fusion approach is then presented to combine both soft and hard fusion for the classification. Soft fusion of data means that each sensors data is combined and classified by the classifier. Hard fusion means each data and sensor processes its data separately and do classification. This novel hybrid approach is then compared to conventional sliding window approach and it gives accuracy of 96Data Preprocessing for radar signals AT first, they have done data preprocessing and feature selection for radar signals. Features are selected for reducing the noise from the data and for classification purposes. Firstly, a fast Fourier transform is applied and its inverse efficiently to the data. Fast Fourier is an algorithm used to compute the discrete Fourier. This FFT is applied on the raw data matrix to transfer it into Range Time Domain. After this background is removed by applying filter. This helps in highlighting the range bins that contains the target on which short Time Fourier method is implemented which has a hamming window of 0.2 seconds. This study collects the data with 15 male and 1 female participants. The participants were made to perform daily activities in the activity zone in front of the radar with 1m duration and three wearable IMU's which are placed on the participant's body Data Preprocessing for inertial sensor data. After that some DC components for example gravitational effect occurs when acceleration measures the magnetic field of earth and acceleration changing which changes because of change in activities. A part from this noise from the vibration from outside also needs to be removed. Tilting also effects the data and performance of inertial sensor so it is also need to be removed. For this a simple filter called bandpassWDC was used to reduce DC components and vibrations. Before filtering these Fast Fourier transform was required to plot the spectrum for selecting the right frequencies, that are set at lower 10 frequency which is 0.1Hz and higher frequency band which is 25Hz.

2.1.3 Feature Selection Process

Radar sensors and three wearable sensors give 194 features. Feature selection is implemented to extract the most suitable feature of the data. This paper Sequential backward selection (SBS) combined with a quadratic SVM classifier. Sequential backward selection is a feature selection process used to low the dimensionality of the data by iteratively removing features. This is helpful when subset of features is required. A threshold of remaining 50 features were selected as hard stop criterion for features approximately 25 percent.

2.1.4 Classification Results

A conventional sliding-window method is to process data, in this method a bunch of record is taken and it is label as one single activity and it is then passed to model. Conventional sliding window is used for time-series data or continuous my data. This method divides the data into smaller segments for feature extraction and for classification..5 “Leave one participant out” cross validation method is then used where 15 participants were used to trained the data and 26th subject is used to test the data. This process is repeated for16 times for all the people, data and averaged that are available. The cross-validation method is better than all other methods as compared to other methods for example ‘Hold out’ or ‘k fold’ partition methods because there is no chance for the classifier to record data from end user. Cross validation ensures that it is not getting data from the same user it previously used.

Bi LSTM based Neural Network Radar is basically a kind of recurrent neural network (RNN) common for the its ability of reducing time series data modeling. It has a simple architecture containing 3 gates. Their names are input gate, forgot gate and output date. All the gates have different functions for example input gate makes decision that which information is to be remembered and forget gates makes a decision that which information is not important and drop it out. Output gate is a procedure to decide which input from the memory is important to make an output. The working principle of LSTM is this that it makes prediction at each time unit. For the classification task for the time related data both past and input features are important. LSTM has the ability to learn the backward and forward long-term dependencies between small timestamps of the data sequence. Bi LSTM is comprised of the single input layer and one soft max layer. It takes data (continuous values) from the sensors as input. The output dense layer turns the output vector from the network into equal length probability matrix. The class that has highest probability is taken as label. To compare the limitation of conventional sliding window method and which makes classification better. A double layer Bi-LSTM network architecture is used for classifying continuous activities and verified leaving one participant out method.

Most precise results are given by the IMU that is on wrist of the personwhen used the radar. A valid dataset for LSTM networks is also chosen to make the training process possible. All the activities of the participants are recorded and it is used for searching optimal hyper parameters and to fine tune after the first training The features are used as inputs of the proposed Bi-LSTM are metrics extracted from the original radar spectrograms and wearable data as a function of time. For the radar sensor, mean, standard deviation, skewness, kurtosis upper and lower envelopes are considered calculated for each bin.

TABLE 2.1: Literature review table

Ref	**Method-/Technique**	**Evaluation parameter**	**Contribution**	**Limitation**	**Future Directions**
[16]	Mutual Feature Selection algorithm to select features. The Pearson correlation coefficient SVM for classification Brouta feature selection algorithm modified DAG-CNN model	Acuuracy= 0.99 Sensitivity=0.97 Specificity=1.00	Improved accuracy of up to 99%. DAG-CNN architecture that can increase the dimension of input features. Better performance	Lack of available indicators directly related to dynamic near fall	Various other models will be used. In depth study on detailed classification

[3]	In this approach,the model used is one class Support Vector Machine. technique is used to remove background noises.	AUC value=0.998 TPR=0.968 Optimal FNR= 0.313	Privacy issues Are reduced. Less sensitive	Computational cost is high. This technique is more prone to errors	Future work considers robust techniques for these realistic environments thatprovide an accurate estimate of the number of sources in the mixture and source separation
[20]	In this approach, the model used PCA for dimensionality reduction and Killer Heuristic Algorithm for Fall Detection.The collected details are then used to extract features such as nean,minimum standard deviation and acceleration.PCA is used to collect most relevant features	Algorithm Accuracy RF 0.99 SVM 0.97 NB 0.99 LR 0.97 KNN 0.97	Improved accuracy of up to 99% DAG-CNN architecture that can increase the dimension of input features Better performance	Lack of available indicators directly related to dynamic near fall	Various other models will be used. In depth study on detailed classification
[13]	Fall detection is performed by applying the intelligent Alex Net convolution network.	Algorithm comparison F1 Measure KHANCN 99.48% OFSVM 97.87% ANN 98.42 % RNN 98.86% LWNN 99.063%	The fall detection based on the created wearable sensor device is evaluated using simulation results, and the system recognizes a fall with maximum accuracy and minimum complexity.	Not new learning classifiers are used for classifying fall and non-fall events	The excellence of the system will be further improved using optimized learning classifier and feature selection processes.

[19]	Data were analyzed using Tensor Flow where a pretained LSTM model was used to classify the data as fall or no fall. Multilayer perceptron (MLP), LSTM, support vector machine (SVM), weighted K-nearest neighbors, and bagging and boosting trees	Mobi-Act=99.62 KNN=96.2 SVM=96.2 Threshold=81.3 LSTM=95.8	99% accuracy With cheap wearable sensors	Machine learning models to identify other activities and analyses biometrics like the heartbeat of the subject	Framework to Support multiple types of sensors, parallel data-processing pipelines, and a cloud platform
[12]	A Kalman filter is used to reduce noisy data.	Sensitivity= 96% Specificity=97.6% Precision=96% Accuracy 96.9%	Performs at a speed of approximately 7–8 analyzed images	Greater computational cost for image processing. Performance is currently limited by the non-optimized threading	Improve fall detection accuracy as a color independent method for tracking subjects through a scene and acquiring new data
[17]	3D CNN is developed which is combined with LTSM. Data obtained from 3D CNN is fed into LTSM	3D CNN= 84.87% LTSM =92.01% 3DCNN And LTSM = 84.27%	100% accuracy is obtained which is the best result ever. Superior Performance in activity recognition	Inclusion of more frames with 3D CNN will increase the computational cost	Accuracy will be improved in futrue
[15]	Data is collected from two sensors. IMU sensors and Radar sensors. Bi-LSTM network is used for classification. This LSTM uses fast fusion.They are not doing real time experiments in wider environment	Bi-LSTM 89.1% Hybrid Fusion 96% SoftFusion Method 94.7%	It is giving hybrid approach for fall detection which uses soft fusion and hard fusion both in LSTM model	It is not doing validation in the wider environment. No Deep architecture neural network	Their future work includes format of input data for example radar data from range time domain. Testing of classification with different sensors

[21]	A design of a software Architecture based on recurrent neural networks based on LSTM cells	SVM 97.93	This paper is proposing a software architecture that takes less memory and has less computational cost	Lack of complete dataset and annotations	Tests with the real subject in real life scenarios will be obtained. Extensive datasets will be used that will help in classification

2.1.5 Soft Fusion

Soft Fusion With radar and wearable IMU using radar Combining of data at decision level between the results of the radar and IMU is used for improving the results[19]. At SoftMax layer produces a confusion matrix, it tells the network chooses the correct output class. To generate the new prediction label soft fusion is used. Soft fusion is a procedure to make the new prediction label by combining the scores from separate sensors, in this case, radar and wrist IMU. Soft fusion has low computational load and still provides significant improvement. In thus research average classification increased to 94.7Hard fusion uses the prediction results from radar and wrist IMU directly. There is no need for combining and weighting for confidence level. Naive Bayes is used for hard fusion approaches. Future work of this paper is that it will seek to validate the methods into a wider perspective and activities including a large set of measurement environments aspects angles with respect to radar Deeper architectures are considered for neural networks and more amount of data is needed to avoid the overfitting problems. This paper future's work will also contain testing of model from different sensors.

Falls can cause severe injuries to the people especially in elderly people. If no immediate care is given it can lead to death. A fall detection system aims to detect a fall as soon as the falls occurs and it generates an automatic alarm request. For this, wearable and embedded sensors are in use. Wearable sensors are considered as the most used sensors because they can continuously track and monitor the activity of the person. Also the benefit of these sensors is this that they are more effective, more feasible and less intrusive. But the dataset that are obtained with the help of these sensors are not really devised for supervised learning. This study discusses the recurrent neural network based software architecture that can be better for fall detection. The dataset that is used in this paper is SiS Fall datset. This dataset acquired 38 participants among them 15 are elderly people. This dataset is gathered by the accelerometer that is fixed in the person's body. This dataset is refined and preprocess so that Recurrent neural network can use it. This work then shows the best process that shows the architecture can be minimized and accurate hyper parameters can be selected and they lead to a working model which can be compared with existing techniques This work discusses the implementation and design of DL technique that is used for fall detection. Its limitation is resource constrained wearable device. An Artificial Neural Network is proposed that is based on LSTM cells. There is a tradeoff between detection performance, power consumption and architectural complexity. For this several architecture is proposed. SisFall dataset is evaluated and three classes are added manually which are BKG, ALERT and FALL. At the end evaluation parameters are checked which are specificity, accuracy and sensitivity. Evaluation parameters tell the RNN architecture can be reached to a comparable performance to the state of art. The proposed approach is viable because of the best use memory occupation and battery duration is also good. So, for online real-time processing it can last for several days. Results of this approach shows that the DL approach is better for fall detection. For meeting the desired architecture that can achieve the desired performance level, and less computational power and less memory requirement. The experience made in this work is this that there was no proper complete datasets with the annotations is experiencing an uncontrolled transition towards a wanted state for example near fall. BKG The BKG class aim is to contain all daily life activities that are not based on fall. These activities are to walk, to go up and down the stairs, to sit on the chair and so on. Design and Implementation of RNN LSTM cells are popular structure of DL. Due to their ability of efficiently capturing long term relationships in the data The tensors size inside each LSTM is called the inside dimension and it is a parameter of design which is denoted by n. n value should be large enough to provide the good capabilities for prediction and good generalization. It also should be small to contain the small network size. The number of layers of recurrent neural network is denoted by 'k'. In this proposed architecture the value of k should be in the range of 1,2. There are many different architectures that varies from this architecture because they have different values for k and n and by selecting different input networks. Firstly, the input data is preprocessed. It is processed at the fully connected Layer1, whereas, a second fully connected layer Collects the output from the LSTM cell at Layer 4 or Layer 6 and this output is given to the final Softmax classifier.

It classifies the data according to the number of classes considered which are ALERT, BKG and fall. Batch normalization layer is also considered by the architecture. Normalization layer is layer2 and layer 2 is to preparing data and three dropout layers. Layer ,5 and 7 are used for training to improve generalization. Temporal unfolding is then used for the training of RNN temporal unfolding means that input data for training is divided in sub-sequences. These subsequences are called windows These windows have specific length that is predefined. Each input is sampled into fixed period of length. The principle of this methodology is this that all the LSTM cells is unfolded into 'w' copies of themselves. The outputs that are given to each copy is the output from the previous cell and the second input is the input signal(x) from the window that is at the corresponding index. With the help of temporal unfolding training of a RNN like the LSTM, as non-recurrent deep network happens which has the capability of all the techniques that exist in the DL field. The parameter 'w' is very important parameter. It is a width of a window. As 'w' decreases in the larger parts of the events the activities 'FALLS' or 'ALERTS' do not come in a single window, it is making it harder for the network to record the full activity. Because if the window size is small the activity will be split into multiple windows so it is difficult for the network to capture the full window.For example if in a first window person walks and on the second window he is walking and on third window fall occurs than model will predict it non fall. Because windoow size is smaller and and activity is splitting into multile windows. We have also trained the model that if number of falls are more than number of non falls it shuld dtect fall but when model takes the window size large than in a first window if persson walks and in a secnd window he is still walking and in a third window he falls so model will detect fall. In this study we have studied that human fall is an issue that can affect elderly people. Fall detection can be classified as wearable and non-wearable devices. Each of these approaches have their own pros and cons. Most of the approaches use wearable devices because it enables a continuous tracking off user. But wearable devices have issue that they required continuous charging as well as daily usage issues. Non wearable approaches do not have these issues because people do not need to wear on them. In a non-wearable approach, it is common to find a solution based on ambient network or computer vision. However, the non-wearable devices are limited in space and they are not able to detect fall beyond the radius. Base paper provides a non-wearable solution which uses an IoT based architecture and an IoT based architecture can handle accelerometer and gyroscope. Base paper is based on three layers (IoT) structure, Fog layer, Cloud layer and Edge layer. The main aim of edge layer is to acquire in real-time and have continuous monitoring of system. The fog layer is used for the classification of event. The edge layer when detecting an off normal event will send data to fog layer for classification. The cloud layer is responsible for the system evolution and classification model. In base paper the model that is used for classification of activities is Artificial Neural Network.

Chapter 3

Proposed Methodology

3.1 Introduction

The chapter 2 presented an overview and analysis of fall detection approaches that are based on wearable and non-wearable devices. The analysis in the previous chapter provided the basis for the problem statement and later on the methodology adopted in this work. The main objective of this study is to compare the performance of both types of approaches for fall detection. There are certain inherent pros and cons of both approaches but the focus in this study is the comparison on the basis of performance w.r.t accuracy in fall detection. This chapter discusses our methodology in detail focusing on the devices used in our methodology, the classifiers, the datasets used to train these models and the setup that was established for our experiments.

i. Sensors:It checks changes in the related area and transfers them into signals that are electrical. ii. Actuators: It transfer signals that are electrical into physical motion that is physical or other ways of output. iii. Micro structures: Very small parts that are mechanical such as gears, springs, and levers. iv. Microelectronics It is the processes of signals from the sensors and controls the actuators.

The structure of the chapter is as follows: section 3.1 presents an overview of our methodology mentioning all steps briefly. Section 3.2 to Section 3.6 shows the description of the rest of the chapter. The rest of the chapter describes the overall methodology that we are using in our research. According to our methodology we are performing experiments on wearable and non-wearable devices and comparing the results. Firstly, we have performed experiments on the K-Fall dataset but our results on dataset were not good so we have chosen our own dataset on a real time. Then we have performed our experiments on this dataset and also extracted some useful features discussed in Section3.3. Results show that LSTM model is giving more accurate results on collected dataset. For non-wearable devices we are using the dataset of base paper. We have extracted the same features described in section 3.3 and performed the experiments using LSTM model. Results are improved when we compare them with the base paper results. At the end we compare wearable and non-wearable results. We have seen that accuracy of wearable devices are more than the accuracy of non-wearable devices. Below is the brief description of methodology.

3.2 Brief of Methodology

Although the objective in our proposed work was to perform experiments using both the wearable and non-wearable devices, but the later approach could not be applied due to different reasons:

i. Initially, the dataset for the non-wearable devices was not available publicly. For this purpose, we contacted the authors of the base paper multiple times, and ultimately, they provided us with their data. This data was used to ML train model for the non-wearable devices.e have not performed real time experiments on this because the environment was not suitable for this.We can compare the results with wearable devices dataset with already non wearable devices dataset.

ii. The setup and the details of the non-wearable devices were not available in spite of multiple time contacts with the authors. We needed to acquire these devices to establish the setup for fall detection with non-wearable devices. Due to unavailability of the devices and setup, we could not perform these experiments and mainly relied on the results produced in the base paper.

For the fall detection using wearable devices, we performed multiple experiments that comprise following steps:

Our datasets are taking reading from two sensors, that is, accelerometer and gyroscope. The working and basic intro of these two sensors are given at the start of section 3.3.

i. From literature, we selected and downloaded the dataset for the fall detection using wearable devices. The description of the dataset is given in the later part of section (3.7.1).

ii. We analyzed different ML classifiers and finalized Long Short Term Memory Model due to its better characteristics for the problem and performance (3.8).

iii. Then we trained our fall detection model using the dataset mentioned in (a) above (3.3).

We selected hardware (3.5.1) for

i. Capturing the movement readings of the subject (person under observation), and

ii. The fog node where the ML model was to be deployed and the data had to be processed. Moreover, we also finalized the communication protocol betwe en these two devices.

iii. Third hardware is basically the PC on which the MIS of SAFE-RH is running.
iv. Deployed the ML model (of step (c)) on the fog node (3.12.2).

iv. We requested a person (referenced as subject in remaining text) to perform the fall event. The device selected for movement recording was given to subject to wear.

v. Subject performed different types of falls and the data through sensors was being recorded. This data was passed through the ML model which was predicting different activities including Fall(section).

vi. The setup was tuned and adjusted again and again for better data processing and improved accuracy (section).

vii. Finally, results were compiled and evaluated with non-wearable results (chapter 4).

3.2.1 Proposed Methodology Diagram

Our methodology has been explained in detail in the following.

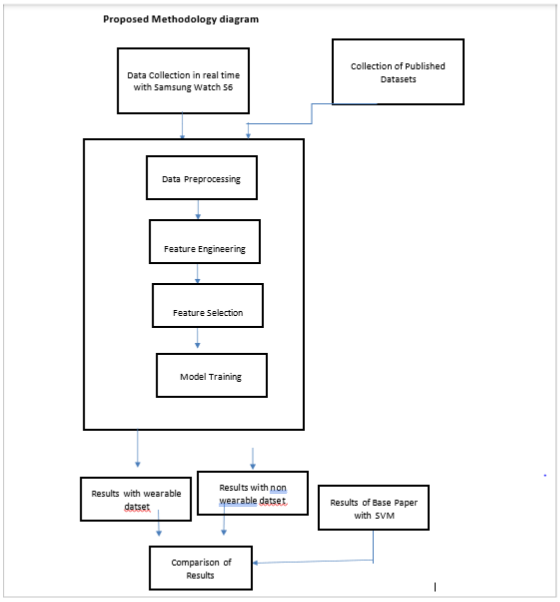


FIGURE 3.1: Proposed Methodology Diagram

3.3 Dataset Description

In this section, we are describing the datasets that we have used in this work. We have mainly two types of datasets; one is from non-wearable devices that has been taken from the base paper [1]. For second type of dataset, we used two approaches, one, we used a published dataset to train our model, called K-fall dataset [24], deployed model onto the fog node and performed experiments. This model was not producing good results in fall detection, mainly due to the reason that the K-fall dataset has been generated by wearing sensors around the waste, whereas we are using a wrist watch for movement data capturing. In the second approach, we generated our own dataset where we performed experiments in which we recorded the values of the sensors and labeled associated activities. The model trained on this data gave much better results so we ultimately used this model for comparison of results. Before explaining the two datasets, we are giving brief description of the sensors that are mainly used to record the movements of the subjects [10]. Most commonly used sensors are accelerometer and gyroscope [These sensors are accelerometer and gyroscope that are briefly explained in the following:

3.3.1 Accelerometer

Accelerometer is an automatic sensor. Microelectromechanical Systems (MEMS) have become the cornerstone of modern accelerometry, enabling the development of compact, low-power, and highly precise devices that are ubiquitous in our daily lives[23]. In the realm of modern technology, the seamless integration of sensors, actuators, microstructures, and microelectronics has become paramount in the development of intelligent systems capable of perceiving, processing, and responding to their surrounding environment. Accelerometer sensor checks changes in the related area and transfers them into signals that are electrical. Accelerometer consists of an actuator which transfer signals that are electrical into physical motion that is physical or other ways of output. It also has microstructures that are very small mechanical parts such as gears, springs, and levers. A small object mass called proof mass is hanged in the device that walks with respect to the accelerometer frame because of external acceleration. It has capacitive plates. These are the plates use to check the displacement of the proof mass and transfers into an electrical signal. Accelerometer checks acceleration, which is the acceleration with respect to free-fall. This has both parts of acceleration (such as gravitational force) that cannot be changed as well as those parts that can be changed. When the device is accelerated, the proof mass in the accelerometer moves due to inertial forces. As the proof mass moves, it changes the capacitance between the proof mass and fixed place in the accelerometer. Changes in capacitance are converted into electrical signal proportional to the acceleration. A small object mass called proof mass is hanged in the device that walks with respect to the accelerometer frame because of external acceleration. It has capacitive plates.These are the plates use to check the displacement of the proof mass and transfers into an electrical signal. Accelerometer checks acceleration, which is the acceleration with respect to free-fall. This has both parts of acceleration (such as gravitational force) that cannot be changed as well as those parts that can be changed. When the device is accelerated, the proof mass in the accelerometer moves due to inertial forces. As the proof mass moves, it changes the capacitance between the proof mass and fixed place in the accelerometer. Changes in capacitance are converted into electrical signal proportional to the acceleration Embedded in Samsung.

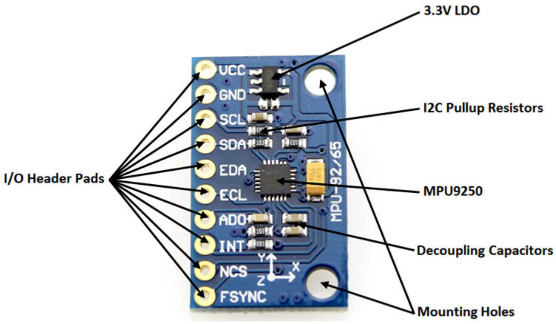


FIGURE 3.2: Accelerometer

3.3.1.1 Working Principle

The above diagram shows the MPU9250 sensor module, a 9-axis motion tracking device integrating an accelerometer, gyroscope, and magnetometer. The module features I/O header pads for interfacing with external circuits, including a VCC pin for power supply and a GND pin for grounding. For I2C communication, the module includes SCL (Serial Clock Line) and SDA (Serial Data Line) pins, while additional EDA and ECL pins handle data and clock lines for the embedded magnetometer. The ADO pin allows selection of the I2C address, and the INT pin can generate interrupts to notify a microcontroller when new data is available. The module also supports SPI communication via the NCS pin and includes a FSYNC pin for synchronizing with other systems. A 3.3V low dropout regulator (LDO) ensures stable power to the MPU9250, while onboard I2C pullup resistors maintain proper signal levels on the communication lines. Decoupling capacitors filter power supply noise, ensuring the stable operation of the MPU9250. The module also has mounting holes for secure attachment.

3.3.2 Gyroscope

Gyroscope is similar to accelerometer, but it is designed to measure angular velocity (rate of rotation) Vibrating Structure Gyroscope. Typically gyroscopes are three orthogonal and accelerometer that are measure an angular velocity and acceleration linear in a three dimensional space are contained in an IMU[24]. VSG is a common type in consumer electronics, where a vibrating mass detects Coriolis force caused by rotation. Functionality: Gyroscope measures the rate of rotation around one or more axis. They help determine orientation, angular velocity and changes in orientation.

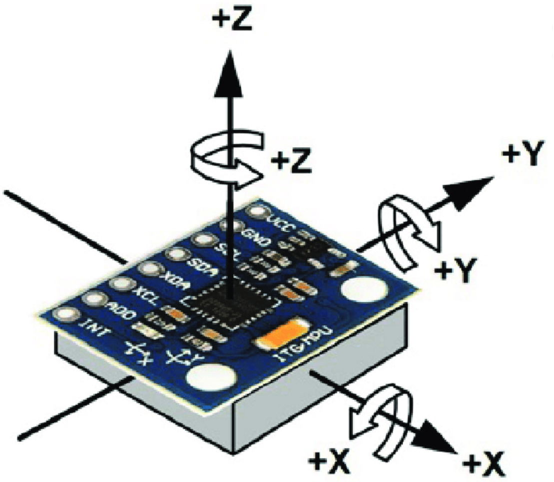


FIGURE 3.3: Gyroscope Sensor

3.3.2.1 Working Principle

It has Coriolis effect it occurs when the device rotates, the vibrating structure inside the gyroscope experiences Coriolis force perpendicular to both the input axis and velocity of the input. This force is detected as changes in capacitance or other measurable quantities.

Gyroscope makes electrical signals proportional to the rate of rotation around each axis as an output. Embedded in Samsung. Samsung utilizes gyroscope alongside accelerometer for enhanced functionality. Gyroscope accurately track the rotational movements for gaming, virtual reality (VR) and augmented reality (AR) applications.

The image illustrates a gyroscope, a sensor that measures angular velocity and helps in determining the orientation of an object in space.

The gyroscope functions by detecting the rate of rotation around its three axes: X, Y, and Z. These axes correspond to the three-dimensional space in which the device operates, allowing it to measure rotational movement in all directions. The arrows in the image indicate the positive directions of rotation for each axis, which are essential for interpreting the sensor's output correctly.

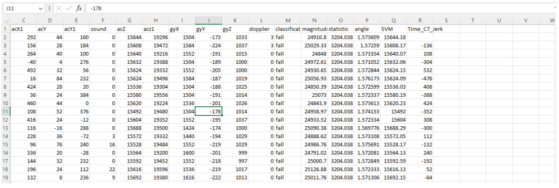
3.3.3 Dataset for Non-Wearable Devices

For non-wearable devices, we have a dataset from[1].We are usinng this as a base ppaer because they are using the same technique which is fall detection using IoT based srchitecture.It is a time series data. Experimental setup shows that two accelerometers and one gyroscope are used for data collection. All are placed on floor and are embedded in an IoT device. The authors recorded values of sensors while subject performing different activities like, walk, sit and different types of falls. Data set with columns acX, acY, acZ, acX1, acY1, acZ1, gyX, gyY and GyZ contain values of two accelerometers and one gyroscope. As the subject performs different activities the sensors that placed on the ground sense by floor vibration and record the value.

Apart from the attributes recorded in the base paper, we have also extracted some features from this dataset using the approach presented in [17]. For example firstly we have calculated magnitude for acX, acY, acZ, acX1, acY1, acz1, gyX, gyY and gyZ. This feature takes the mean square of the values along x-axis, y-axis and z-axis. This formula is derived from the Euclidian distance in three-dimensional space representing how far a point (x, y, z) is from the origin which effectively measures the overall intensity of the signal at that time stamp.

This feature matters because instead of analyzing three separate components, the SVM provides a simplified, unified measure of motion or orientation.

This feature helps in overall intensity of the fall event rather than specific direction of the motion which can be less relevant. In our case of comparing wearable and non-wearable devices for distinguishing between fall and non-fall the magnitude of the movement during a fall is much higher than during normal activities. Using LSTM helps to identify this difference in magnitude making it easier to classify such activities.



J11 fx -178

	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	acX1	acY	acY1	sound	acZ	acz1	gyX	gyY	gyZ	doppler	classificat	magnitud	statistic	angle	SVM	Time_C7_Jerk
2	292	44	160	0	15644	19296	1504	-173	1033	3	fall	24910.8	3204.038	1.573609	15644.18	
3	156	28	184	0	15608	19472	1584	-224	1037	3	fall	25029.33	3204.038	1.57259	15608.17	-136
4	264	40	100	0	15640	19216	1552	-191	1015	0	fall	24848	3204.038	1.573354	15640.07	108
5	-40	4	276	0	15632	19388	1504	-189	1000	0	fall	24972.61	3204.038	1.571052	15632.06	-304
6	492	32	56	0	15624	19332	1552	-205	1000	0	fall	24930.65	3204.038	1.572844	15624.15	532
7	16	84	232	0	15624	19496	1584	-187	1019	0	fall	25056.93	3204.038	1.576173	15624.09	-476
8	424	28	20	0	15536	19304	1504	-188	1025	0	fall	24850.39	3204.038	1.572599	15536.03	408
9	36	24	384	0	15580	19556	1504	-191	1014	0	fall	25073	3204.038	1.572337	15580.19	-388
10	460	44	0	0	15620	19224	1536	-201	1026	0	fall	24843.9	3204.038	1.573613	15620.23	424
11	108	52	376	0	15492	19480	1504	-178	1014	0	fall	24958.97	3204.038	1.574153	15492	-352
12	416	24	-12	0	15604	19352	1552	-195	1037	0	fall	24933.52	3204.038	1.572334	15604	308
13	116	-16	288	0	15688	19500	1424	-174	1000	0	fall	25090.38	3204.038	1.569776	15688.29	-300
14	228	36	-72	3	15572	19332	1440	-194	1029	0	fall	24888.62	3204.038	1.573108	15572.05	112
15	96	76	240	16	15528	19484	1552	-219	1029	0	fall	24986.76	3204.038	1.575691	15528.17	-132
16	336	20	-28	0	15564	19200	1600	-201	999	0	fall	24791.02	3204.038	1.572081	15564.13	240
17	144	32	232	0	15592	19452	1552	-218	997	0	fall	25000.7	3204.038	1.572849	15592.59	-192
18	196	24	112	22	15616	19596	1536	-219	1017	0	fall	25126.88	3204.038	1.572333	15616.13	52
19	132	8	236	9	15692	19380	1616	-222	1013	0	fall	25011.76	3204.038	1.571306	15692.15	-64

FIGURE 3.4: Dataset for Non Wearable Devices Without Feature Selection

3.4 Feature Selection

We are performing feature selection on wearable and non-wearable dataset because with the help of feature selection model is able to used te most important data for fall detection and give accurate results.We are not using feature fusion because we are gettiing data from single source and not using data from multiple sources and not combining them at the end.

The SVM on the horizontal plane is valuable feature for analyzing the magnitude of the motion or orientation change specifically in the horizontal direction. By focusing on the x, y components of our sensor data, this measure provides a valuable feature of and direction related view of horizontal motion which can be particularly useful in contexts such as detecting falls where horizontal motion might be more indicative of certain activities or events. Another feature that is extracted is the angle between the z-axis and vertical. The formula calculates the angle between the vector formed by ax, az and negatively ay component. This sensor can be thought of as measuring the tilt the sensor in the vertical plane which is practically useful for detecting falls or significant orientation change. By calculating this angle for each dataset of relevant sensor readings in dataset we can obtain a feature that captures important orientation information in a simplified and intuitive manner. Measure of the sensor; orientation with respect to the vertical direction which can be useful for detecting falls or other events where the device's orientation changes significantly. Then the dataset with Magnitude, magnitude with respect to horizontal plane and angle between z-axis and vertical is given to LSTM model. Magnitude Feature SVM calculates the magnitudes of the columns of 'accx' 'accy' 'accz' 'gyrox' gyroy' 'gyroxz' and the result is saved to the original column which is named as magnitude and the output is shown "Magnitude computed and saved in original files" after column is successfully saved in the original file. Formula: Sum vector magnitude =

$$C1[k] = \mathrm{RMS}(\tilde{a}[k]) = \sqrt{a_x^2[k] + a_y^2[k] + a_z^2[k]} \tag{3.1}$$

3.4.1 RMS Feature

Root Mean Square (RMS) feature is calculated in above code. Files are reading from the directory the ‘D:’ directory one by one in a loop. Then columns are given on which magnitude feature is calculated. These columns are ‘acX’, ‘acY’ ‘acZ’, ‘acX1’, ‘acY1’, ‘acZ1’, ‘gyX’, ‘gyY, ‘gyZ’ Output is shown as “RMS values are computed and saved in the original file. Orientation of the columns “acX1”,” acY1”,” acz1”, ”gyX”, “gyY”, “gyZ” are computed and saved in the original file.

3.4.2 Formula of Sum vector magnitude on horizontal plane

$$C_2[k] = \sqrt{a_x^2[k] + a^2[k]}$$

3.4.3 Statistics Feature

This feature calculates the magnitude of the standard deviation in the x and z directions which represent two components of acceleration on the horizontal plane. By combining these components using the square root of the sum of their squares we get a singular scalar value that represents the overall variability of the acceleration on the horizontal plane at each time step-k. This feature is used in the motion analysis where the standard deviation magnitude on the horizontal plane helps quantify the variability of movement or acceleration in that plane, which could be important in accessing stability, movement patterns or detecting specific types of motions.

3.4.4 Orientation(Angle)Feature

This feature calculates the angle between the z-axis and vertical by considering both the horizontal’s plane acceleration and the vertical acceleration and the vertical acceleration. This angle is crucial for understanding orientation relative to gravity and it is widely used in posture analysis, motion tracking and navigation tracking.

Formula

$$C4[k] = \operatorname{atan2}\left(\sqrt{\tilde{a}_x[k]^2 + \tilde{a}_z[k]^2} - \tilde{a}_y[k]\right) \tag{3.2}$$

3.4.5 Jerk(Rate of Acceleration change) Feature

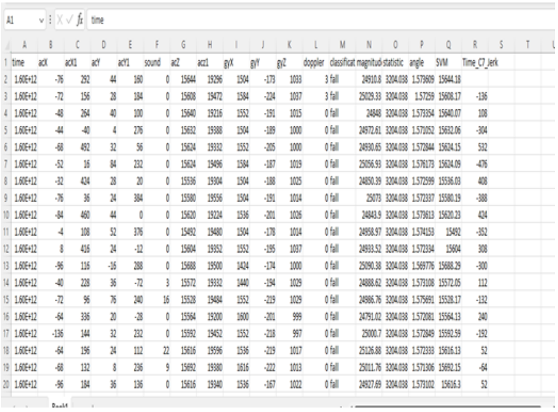
This feature tells how quickly the acceleration is changing over time, which is important for understanding dynamic behavior in systems like wearable sensors

Formula

$$C7[k] = \frac{\tilde{a}_x[k] - \tilde{a}_x[k-N]}{t[k] - t[k-N]} \tag{3.3}$$

3.4.6 Dataset for non-wearable devices

First column 'acX1' has 95 records total. It is a time series data that means sensors are sending values in a continuous time. This data has 100 milliseconds period readings. It means one second is divided into 100th milliseconds. For example, in a first row and second column. The value 76 represents the x-axis of first accelerometer. In a real time, it can be identified that which activity is performed when accelerometer gives the reading -76. The value -76 indicates the significant change in the negative x-axis. Similarly, 44 value indicates the moderate change in y-axis. The value 15664 indicates a strong change in acceleration value. The value at x-axis of gyroscope is 1504 which indicates the movement in the direction of x-axis and, it is considered as falling forward or falling. The value at 'gyY' is 173 which indicates the rotation at sideways .The value at 'gyZ' is 1033 which indicates the body twist at z-axis.



A1 time

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	time	acX	acX1	acY	acY1	sound	acZ	acz1	gyX	gyY	gyZ	doppler	classificat	magnitud	statistic	angle	SVM	Time_C7_Jerk
2	1.60E+12	-76	292	44	160	0	15644	19296	1504	-173	1033	3	fall	24910.8	3204.038	1.573609	15644.18	
3	1.60E+12	-72	156	28	184	0	15608	19472	1584	-224	1037	3	fall	25029.33	3204.038	1.57259	15608.17	-136
4	1.60E+12	-48	264	40	100	0	15640	19216	1552	-191	1015	0	fall	24848	3204.038	1.573354	15640.07	108
5	1.60E+12	-44	-40	4	276	0	15632	19388	1504	-189	1000	0	fall	24972.61	3204.038	1.571052	15632.06	-304
6	1.60E+12	-68	492	32	56	0	15624	19332	1552	-205	1000	0	fall	24930.65	3204.038	1.572844	15624.15	532
7	1.60E+12	-52	16	84	232	0	15624	19496	1584	-187	1019	0	fall	25056.93	3204.038	1.576173	15624.09	-476
8	1.60E+12	-32	424	28	20	0	15536	19304	1504	-188	1025	0	fall	24850.39	3204.038	1.572599	15536.03	408
9	1.60E+12	-76	36	24	384	0	15580	19556	1504	-191	1014	0	fall	25073	3204.038	1.572337	15580.19	-388
10	1.60E+12	-84	460	44	0	0	15620	19224	1536	-201	1026	0	fall	24843.9	3204.038	1.579613	15620.23	424
11	1.60E+12	-4	108	52	376	0	15492	19480	1504	-178	1014	0	fall	24958.97	3204.038	1.574153	15492	-352
12	1.60E+12	8	416	24	-12	0	15604	19352	1552	-195	1037	0	fall	24933.52	3204.038	1.572334	15604	308
13	1.60E+12	-96	116	-16	288	0	15688	19500	1424	-174	1000	0	fall	25090.38	3204.038	1.569776	15688.29	-300
14	1.60E+12	-40	228	36	-72	3	15572	19332	1440	-194	1029	0	fall	24888.62	3204.038	1.573108	15572.05	112
15	1.60E+12	-72	96	76	240	16	15528	19484	1552	-219	1029	0	fall	24986.76	3204.038	1.575691	15528.17	-132
16	1.60E+12	-64	336	20	-28	0	15564	19200	1600	-201	999	0	fall	24791.02	3204.038	1.572081	15564.13	240
17	1.60E+12	-136	144	32	232	0	15592	19452	1552	-218	997	0	fall	25000.7	3204.038	1.572849	15592.59	-192
18	1.60E+12	-64	196	24	112	22	15616	19596	1536	-219	1017	0	fall	25126.88	3204.038	1.572333	15616.13	52
19	1.60E+12	-68	132	8	236	9	15692	19380	1616	-222	1013	0	fall	25011.76	3204.038	1.571306	15692.15	-64
20	1.60E+12	-96	184	36	136	0	15616	19340	1536	-167	1022	0	fall	24927.69	3204.038	1.573102	15616.3	52

Book1

Figure 3.5: Non wearable Devices Dataset With Feature Selection

For non-wearable device a dataset with 170 events was created using the IoT device prototype. Each event has one of the following types human body models falling (40 records), chair falling (30 records) cellphone falling (30 records) water bottle falling (30 records) and noise environment (40 records). The dataset is recorded with a 100-millisecond period of readings with two accelerometers and one gyroscope. The classification column had the values Fall and FALSE. This dataset is handling with time values that continue in time. It is important for that dataset to have time windows i.e a time series as input for the Classifier model (LSTM).We have also used SVM as because it is good for hroizontal plane because for handling magnitudes of accelerometer and gyroscope on x-axis, y-axis and z-axis and as we are doing fall detection so we have used SVM. SVM has the ability to seperate classes in multidimensional data and dealing with high-dimensional data that why it is better for our dataset but because it is not giving accurate results because it is not good for time-series data so we have not used it and replaced it with LSTM model because LSTM is good for time series data.

Firstly, we are taking the dataset for non-wearable device The classification column had the values Fall and FALSE. This dataset is handling with time values that continue in time. It is important for that dataset to have time windows i.e a time series as input for the Classifier model (LSTM). The graph above shows the values of the amplitude. When blue line has a peak curve, it represents that the value of the amplitude is high. It shows that the fall has occurred at that place. When blue line has small amplitude, it means ADL activity and when line is a straight line it means a person is still

3.5 Experimental Setup

Our Experiment is divided into two parts. In the first part, we used dataset for non -wearable devices descried in section (3.2.3) and applied LSTM model on it. Then we evaluated the results like accuracy and precision shown in chapter 4. For improving the accuracy of the model, we are using features that we engineered from the published data, like magnitude and RMS, as explained in section (3.2.3).These features extraction help in improving the accuracy of our method. In the second part of our experiments, we applied model on the data which we have collected ourselves along with the features engineered from this data. At the end we compared the results of our first and second technique evaluated the results

3.5.1 Hardware Setup

Sensor setup – The accelerometer and gyro sensors embedded in the Apple Watch Series 6 (Watch S6). Unlike the waist-mounted sensors in the K Fall dataset, the Watch S6 sensors were positioned on the wrist, reflecting a significant positional disparity. This discrepancy introduced a notable challenge: the model trained on waist sensor data struggled to generalize and perform accurately when presented with input from wrist-mounted sensors. Recognizing the necessity of addressing this issue to ensure the practical applicability of the fall detection system, we sought alternative approaches. One such strategy involved exploring the use of the magnitude derived from the accelerometer and gyro data. By combining the readings from both sensors and calculating the magnitude, we aimed to capture a more comprehensive representation of motion, irrespective of sensor placement. To our satisfaction, training a model on this magnitude data yielded promising results, demonstrating improved performance in real-time fall detection scenarios.

3.6 Fall Detection using Wearable Devices

In this section, we are describing our approach to detection the fall using wearable devices. Firstly, the description of the devices is given, followed by the dataset and the machine learning models that have been used in the experiments. We are using wearable devices because they are feasible in our case e.g wrist watch. Where as non-wearable devices are used but they can not be functional beyond some limited area.

3.6.1 Devices Description

For comparing wearable and non-wearable devices we are using Samsung watch and Raspberry Pi for data collection. Mostly wearable devices contain watches, pendants or bracelet that have inbuild sensors.

3.6.1.1 Raspberry Pi

In the data collection process, we are using the raspberry Pi and Samsung watch S6. Raspberry Pi is the name of a series of single-board computers made by Raspberry Pi Foundation, a UK charity that aims to educate people in computing and create easier access to computing education. The Raspberry Pi launched in 2012 and there have been several iterations and variations released then[16]. The original Pi had a single core 700 MHz CPU and just 256MB RAM. All over the world, people use raspberry Pi to learn programming skills, build hardware projects, do home automation implement cubernetes clusters and Edge computing and even them in industrial applications. The Raspberry Pi, a com-]pact and affordable single-board computer, has become a popular choice for various computing and automation projects. One of the key features of the Raspberry Pi is its ability to be controlled remotely, enabling users to access and manage the device from another device on the same network [25]

Figure 3.6: Raspberry Pi

The Raspberry Pi is a very cheap computer that runs Linux, but it also provides a set of GPIO (general purpose input/output) pins that allow to con- trol electronic components for physical computing and explore the internet of things (IoT). Raspberry Pi controlled remote monitoring system. Raspberry Pi is a credit card sized single board computer with ARM11 microprocessor. It weighs 50g.It uses power rating of 5V, 700Ma and cost effective than actual computer.

Some purchase Raspberry Pis with the intention of learning to program, while others who are already proficient in this area can use the Pi to learn how to code electronics for tangible projects. The open-source community loves the Pi because it gives users autonomy over proprietary closed systems. In a world where technology continues to transform our daily lives, smart watches have evolved from simple time keeping devices into sophisticated personal health monitors. One of the most groundbreaking features in recent year is fall detection[27]. In this case Samsung galaxy S6 watch stands out with its sleek design and bright display. Fall detection is a key feature, automatically sending alerts to emergency contacts if the wearer does not respond to the audible alarm. The LTE variant allows for independent communication without a phone. This smartwatch integrates health features like ECG and skin temperature enhancing its utility. However, it needs frequent charging and false positive can occur during physical activities. It is a reliable choice for those who are seeking safety features.

Samsung Electronics is the developer of the Wear OS smart watch line, which includes the Samsung Galaxy Watch S6. On July 26, 2023, at Samsung's biannual Galaxy event, it was revealed. To send the data to raspberry pi using Samsung watch6 through WIFI. For that both devices should be connected to the same network. Transferring data between a Samsung galaxy watch6 and a raspberry Pi involves several underlying technologies and protocols that enable data transfer between these devices. WIFI this technology allows two devices to connect to each other directly with- out needing a router. It simplifies the process of connecting the watch to a Raspberry Pi. Standard protocols like TCP/ICP are used for reliable data transfer over mobile. TCP/IP protocol is a suite of communication protocols used to interconnect network devices on the internet, it provides

Pulse: 88.0

Tap to Check Temperature

Figure 3.7: Samsung Watch Galaxy 6

end to end data connection specifying how data should be packetized, addressed, transmitted, routed and received. The TCP/IP protocol suite is essential for establishing a standardized set of rules for communication over the internet and other computer network For Data collection subject wears the Samsung Galaxy watch6 on arm.

Firstly, we have to synch the watch.In the data collection process, we have recorded the activities fall, walking and still activities are recorded. For collection of activities subjects wear the watch on arm and starts walking. As the object moves the watch sends the data to the raspberry pi .32 records are sent to server after every 5 seconds. The x, y and z axis of accelerometer and gyroscope change. These values are saved in a csv formatted datasheet. This data sheet has a timestamp column, three columns for x, y and z axis of accelerometer. Three columns for x, y and z axis of gyroscope. And a label column. When the objects start walking data is sent to the raspberry pi and after five seconds. The graph against that data is plotted the peak value of the graph shows the exact fall while other values show the still and walking activities. The line of the graph that has no change shows the still activity whereas the line of the graph that has no abrupt change shows the walking activity. The x, y and z axis values for still activities are same whereas the x,y and z axis values for walking have small change whereas the x,y and z axis of fall are changing abruptly. Accelerometer and gyroscopes are essential components in modern electronic devices like smartphones, including those manufactured by Samsung.

3.7 Experimentation

This section discusses how we have performed experiments, first, to capture the data for model training and later using this model to detect fall in real time. Our experiments are mainly based on wearable devices, that is, sensors in Samsung Galaxy S6 watch. Later, we compared our results with those generated through non-wearable devices for fall detection. Following steps are followed in our experimentation: a) Data collection from wearable devices and labelling b) ML models training and testing after pre-processing c) Model Deployment on Pi (our fog node) d) Capturing the data in real time and passing data to the model e) Generating the alarm if model detects fall f) Evaluating the results generated by model in real time

3.7.1 Dataset for Wearable Devices

We utilized the K Fall dataset, a well-known benchmark dataset extensively used for fall detection research. This dataset contained accelerometer and gyro data collected from sensors positioned on the waist of individuals undergoing various activities, including falls. The goal of KFall dataset is to help in technology development for elderly fall detection and injury prevention. It has 32 young subjects with 21 types of activities of daily living (ADLs) and 15 types of falls from an inertial sensor attached on low back. In total, it contains 5075 motion files with 2729 .ADL motions and 2346 fall motions. In addition, for each fall motion, ready-to-use fall labels (fall initialization and fall impact moment) based on synchronized video references were also included.K-Fall dataset is comprised of two folders.

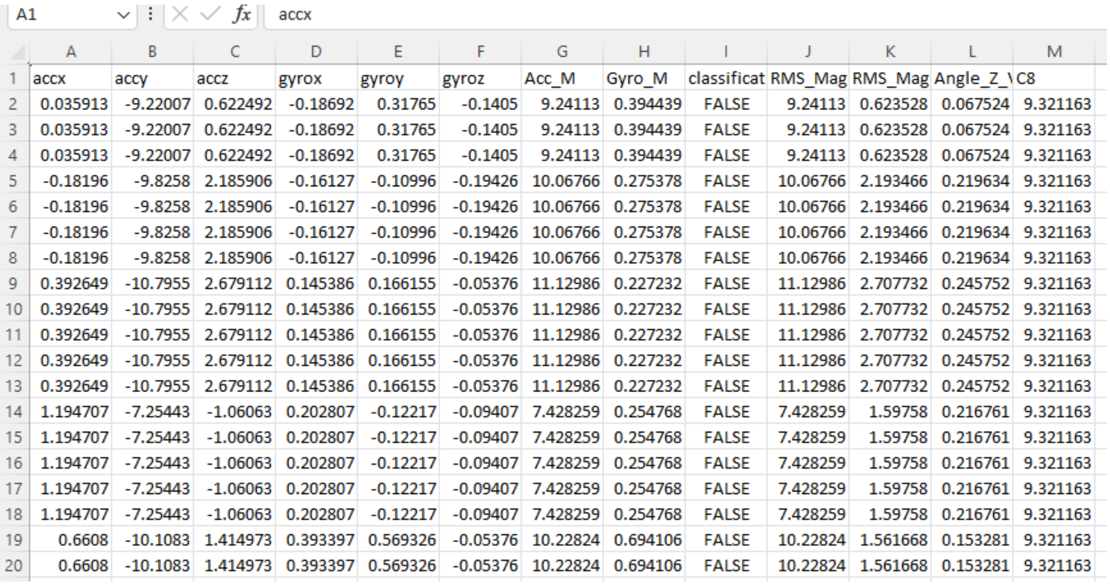
3.7.1.1 Sensor Data

There are 32 sub folders named by subject ID (e.g., SA06). Under each sub folder, it includes all the motion files from the corresponding subject. The naming rule of the motion file is explained here. Take ”SA06T01R01.csv” as an example, ”SA06” means Subject ID is 06; ”T01” means Task ID is 01; ”R01” means Trial ID is 01. Except Task 01, 11, 12 and 17, they are required for 1 trial while other tasks normally have 5 trials. However, due to Bluetooth disconnection, signal sychronization and miscounting issue, some tasks could have 4 trials or 6 trials For each motion file, it contains 11 columns which are TimeStamp(s), FrameCounter, acceleration; gyroscope and euler angle(°) along three axes.

3.7.1.2 Label Data

there are 32 label files named by subject ID. For each label file, there are 6 columns: Task Code (Task ID), Description, Trial ID, Fall onset frame and Fall Impact frame. The last two columns are based on sychronized video reference. Take the first record in SA06_label.xlsx (header is not considered) as an example, it means the fall onset frame and fall impact frame are 130 and 208 in the corresponding data file (SA06T20R01.csv), respectively.

We are collecting data in a real time environment where a subject wears a Samsung watch on his wrist. Watch has inbuilt accelerometer and gyroscope sensors. Both the sensors are recording values at x,y and z-axis. The activities that we are recording are “Walking” “Still” and “Fall”.



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	accx	accy	accz	gyrox	gyroy	gyroz	Acc_M	Gyro_M	classificat	RMS_Mag	RMS_Mag	Angle_Z_\	C8
2	0.035913	-9.22007	0.622492	-0.18692	0.31765	-0.1405	9.24113	0.394439	FALSE	9.24113	0.623528	0.067524	9.321163
3	0.035913	-9.22007	0.622492	-0.18692	0.31765	-0.1405	9.24113	0.394439	FALSE	9.24113	0.623528	0.067524	9.321163
4	0.035913	-9.22007	0.622492	-0.18692	0.31765	-0.1405	9.24113	0.394439	FALSE	9.24113	0.623528	0.067524	9.321163
5	-0.18196	-9.8258	2.185906	-0.16127	-0.10996	-0.19426	10.06766	0.275378	FALSE	10.06766	2.193466	0.219634	9.321163
6	-0.18196	-9.8258	2.185906	-0.16127	-0.10996	-0.19426	10.06766	0.275378	FALSE	10.06766	2.193466	0.219634	9.321163
7	-0.18196	-9.8258	2.185906	-0.16127	-0.10996	-0.19426	10.06766	0.275378	FALSE	10.06766	2.193466	0.219634	9.321163
8	-0.18196	-9.8258	2.185906	-0.16127	-0.10996	-0.19426	10.06766	0.275378	FALSE	10.06766	2.193466	0.219634	9.321163
9	0.392649	-10.7955	2.679112	0.145386	0.166155	-0.05376	11.12986	0.227232	FALSE	11.12986	2.707732	0.245752	9.321163
10	0.392649	-10.7955	2.679112	0.145386	0.166155	-0.05376	11.12986	0.227232	FALSE	11.12986	2.707732	0.245752	9.321163
11	0.392649	-10.7955	2.679112	0.145386	0.166155	-0.05376	11.12986	0.227232	FALSE	11.12986	2.707732	0.245752	9.321163
12	0.392649	-10.7955	2.679112	0.145386	0.166155	-0.05376	11.12986	0.227232	FALSE	11.12986	2.707732	0.245752	9.321163
13	0.392649	-10.7955	2.679112	0.145386	0.166155	-0.05376	11.12986	0.227232	FALSE	11.12986	2.707732	0.245752	9.321163
14	1.194707	-7.25443	-1.06063	0.202807	-0.12217	-0.09407	7.428259	0.254768	FALSE	7.428259	1.59758	0.216761	9.321163
15	1.194707	-7.25443	-1.06063	0.202807	-0.12217	-0.09407	7.428259	0.254768	FALSE	7.428259	1.59758	0.216761	9.321163
16	1.194707	-7.25443	-1.06063	0.202807	-0.12217	-0.09407	7.428259	0.254768	FALSE	7.428259	1.59758	0.216761	9.321163
17	1.194707	-7.25443	-1.06063	0.202807	-0.12217	-0.09407	7.428259	0.254768	FALSE	7.428259	1.59758	0.216761	9.321163
18	1.194707	-7.25443	-1.06063	0.202807	-0.12217	-0.09407	7.428259	0.254768	FALSE	7.428259	1.59758	0.216761	9.321163
19	0.6608	-10.1083	1.414973	0.393397	0.569326	-0.05376	10.22824	0.694106	FALSE	10.22824	1.561668	0.153281	9.321163
20	0.6608	-10.1083	1.414973	0.393397	0.569326	-0.05376	10.22824	0.694106	FALSE	10.22824	1.561668	0.153281	9.321163

Figure 3.8: Dataset For Wearable Devices

50 subjects performed the activities and we have collected the dataset. Our dataset has these columns “accx” “accy” “accz” “gyrox” “gyroy” “gyroz” We are extracting features and our feature columns are “AccM ” GyroM ” classification” RMSMagnitude” C8”. The values in the first row represents the three axis of accelerometer and gyroscope. The value at the first column acX is 0.035913 which indicates the change in x-axis of accelerometer. In a real-time it can be observed that which activity was being performed when this value was recorded. It is not a significant change so it is not considered as fall. The value recorded for the y-axis is -9.8258 is very close to the reading of gravity. This value is showing the value of acceleration downwards. Similarly, the value in -acX which is 0.622492 it shows the change in z-axis. The value of gyrox is 0.3175 which shows the minor rotation towards x-axis. The value at gyroy is 0.3175 which shows the minor rotation towards y-axis. The value at ‘gyroz’ is -0.1405 which shows the slight twist during z-axis. MS-Mag shows the total over all magnitude of all axis which is 9.24113.

It is not a large value so it can be identified that it is not a fall. We have selected LSTM model for our dataset because LSTM is good for time-series data. We are also using real-time time series data that we are collecting from watch and also published dataset for non-wearable devices which is aslo time-series data.

3.8 Long Short-Term Memory Model Detailed Overview

Long Short-Term Memory (LSTM) is a specialized type of recurrent neural network (RNN) that is designed to capture long-term dependencies in sequential data.We are using LSTM for our approach because we are dealing with time series data and LSTM is better for that as it has the ability to memorize the previous event. While traditional RNNs struggle with maintaining information over long sequences due to vanishing or exploding gradients, LSTMs solve this problem using a sophisticated gating mechanism that allows them to selectively retain or discard information[18]. Here's a detailed breakdown of how LSTMs work, their architecture, and applications.

3.8.1 LSTM Architecture

The architecture of an LSTM is unique because it incorporates several key components:

3.8.1.1 Memory Cell

The core idea of an LSTM is its memory cell, which can store information over long periods. The cell remembers values that are important for prediction, and its content can be modified through various gates.

3.8.1.2 Gates

LSTMs have three main gates—input, forget, and output gates—that regulate the flow of information into, out of, and within the cell.

3.8.1.3 Forget Gate

Decides what information to discard from the cell state. This gate takes the current input and the previous hidden state as inputs and outputs a value between 0 and 1 for each number in the cell state, determining how much information to keep (1) or forget (0).

3.8.1.4 Input Gate

Controls what information to update in the cell state. It also takes the current input and the hidden state and decides which values to update.

3.8.1.5 Output Gate

Determines the output of the current cell and what information is sent to the next hidden state. It helps decide which parts of the cell state should be output and passed to the next time step.

3.8.2 Advantages of LSTM

LSTMs (Long Short-Term Memory networks) function through a series of gated mechanisms to manage and update their memory over time[22]. Initially, the forget gate decides which information from the previous time step should be discarded or retained. Next, the input gate determines which new data will be incorporated into the memory cell. The cell state is then updated by combining retained information with the new input. Finally, the output gate controls what information is passed on to the next hidden state, influencing both the final prediction and the input to subsequent LSTM units. This systematic approach allows LSTMs to capture long-term dependencies and manage sequential data effectively.

3.8.3 Limitations of LSTM

While LSTMs offer a number of advantages over traditional RNNs, they still have some limitations:

i. Resource-Intensive: LSTMs require significant computational power and memory, especially for large datasets or long sequences.

ii. Training Complexity: Training LSTMs can be slow compared to simpler architectures, and they may require careful hyperparameter tuning for optimal performance.

3.9 Proposed Methodology

Different approaches have been proposed for fall detection using wearable and non-wearable devices however wearable devices stand most for the fall detection because they continuously track the movement of the person. We are comparing both wearable and non-wearable devices for fall detection and evaluating the results of machine learning model on the datasets. Following steps are followed by the proposed methodology:

i. Data collection from wearable devices

ii. ML models training and testing after preprocessing

iii. Model Deployment on Pi

iv. Passing data to the model

v. Feature Selection of Dataset

vi. Testing and Training of ML model

vii. Evaluation of results

viii. Generating Alarm

3.10 Data Collection

For wearable devices we have made an experimental setup using Pi, Server, and Samsung Watch. Subject wears a watch on his wrist and starts performing the activities. We have recorded Walk, Still and Fall activities. Accelerometer and sensor are reading values of their x,y and z axis sending it to the mode that is deployed on Pi.Firstly we used wrist watch on waist but it was not giving acccurate results because little movement was detected as fall which was not correct also wearable devices on waist is not feasible.

3.10.1 Data Collection for Training the Model

We are collecting dataset from Watch sensors and each sensor has three axes, x, y and z. Sensors are generating data continuously. We have programmed the Samsung watch that captures the data from these sensors, attaches timestamp with the reading and transfers it to the Pi through MQTT protocol. The sensors are generating approximately 32 readings in very 5 seconds. Another program running at Pi receives the data and labels the data with “Walk”, “Still” or “Fall” values. Here is the major issue that the data is being generated at watch and it is being labeled at Pi. The main issue was this if the fall occurs in this time, we cannot identify the exact time where fall starts and ends. We do not have the exact value where we need to start and stop the timer. So, we tackled this issue by starting the timer 2 steps before the fall occurs and off the timer after the person falls and stays still. Because when a person falls, he goes into the still situation. When we collect the data from watch because we are collecting data on Pi. Our timer app is running on Pi. So, there is a possibility that Watch has different time and Pi has different time. Now it can be possible that data that is going from watch its current time from watch is also recording. For example, if watch has time one minute and 1 seconds it is recording on csv file on PI and timer app is opening file. Now the time that is coming from watch is matching with the time of Pi. If we start the timer at one minute and 1 second and off it at one minute and 6 seconds but Pi has one minute and 4 seconds so it is possible model is not giving the data of one minute and last 2 seconds of watch time. So it is very important to have the time synch of watch and Pi. Model will match the time of the watch and Pi and save the record of the activity only at matching time duration of Pi and watch. Also there was a issue that we cannot set time on raspberry Pi at microseconds. We were setting it with code but every time we on the Pi it gets updated every time so we have to synch it again. We have to make RTP configuration from Pi. We set time in microseconds on Pi manually. If a subject is performing activity and there is a jerk in activity and time has even 2 seconds of difference, we get 5 to 6 records in every second there may be possibility that fall is occurring in that 2 second so it was also an issue.

3.10.2 ML Models Training for Wearable Devices

In the model training process, we have collected the data in the real time and we have given labels to the data. For collection of dataset, 50 subjects performed activities. The activities that were recorded during the data collection process were sitting, walking and fall. We have sent this labelled data to the model. Model learns from the dataset, as it is already labelled, this process is called model training.For improving the accuracy of the model we are using feature extraction method so we extracted some features which are the magnitude of the data, Magnitude with respect to horizontal plane, Angle between z-axis and horizontal. Time ("Jerk" rate of change of acceleration) and statistic (Standard deviation magnitude on horizontal plane) which are explained in section. After extracting features when we apply (LSTM) model.

3.11 Passing Data to the Pi

To send the data to raspberry Pi using Samsung watch6 through WIFI. For that both devices should be connected to the same network. Transferring data between a Samsung galaxy watch S6 and a raspberry Pi involves several underlying technologies and protocols that enable data transfer between these devices. By default, the Raspberry Pi will be given a dynamic IP address when connected to a (local) network, which makes it easy to get connected. Although such a dynamic address should be fine for most uses, in some cases it is recommended to create a static IP address to ensure a consistent connection. WIFI this technology allows two devices to connect to each other directly without needing a router. It simplifies the process of connecting the watch to a Raspberry Pi. Standard protocols like TCP/ICP are used for reliable data transfer over mo- bile. When working with arrays of Raspberry Pi's, it is crucial to establish a dedicated local area network (LAN) with a specific subnet to which individual Raspberry Pi's can be easily added. Another option is to configure the Rasp- berry Pi as a self-contained Wi-Fi access point, generating its own network for the host device to connect to. This setup can be particularly useful for remote field systems, although it does require some terminal-based configuration. In situations where wireless or wired network options are unavailable, the Raspberry Pi offers several alternative connectivity solutions to directly interface with the device. One approach is to establish a direct computer-to-computer connection using an Ethernet cable.

3.12 Passing Data to the Model

To send the data to raspberry pi using Samsung watch6 through WIFI. For that both devices should be connected to the same network. Transferring data between a Samsung galaxy watch S6 and a raspberry Pi involves several underlying technologies and protocols that enable data transfer between these devices. By default, the Raspberry Pi will be given a dynamic IP address when connected to a (local) network, which makes it easy to get connected. Although such a dynamic address should be fine for most uses, in some cases it is recommended to create a static IP address to ensure a consistent connection. WIFI this technology allows two devices to connect to each other directly without needing a router. It simplifies the process of connecting the watch to a Raspberry Pi. Standard protocols like TCP/ICP are used for reliable data transfer over mobile. When working with arrays of Raspberry Pi's, it is crucial to establish a dedicated local area network (LAN) with a specific subnet to which individual Raspberry Pi's can be easily added[29].

Another option is to configure the Raspberry Pi as a self-contained Wi-Fi access point, generating its own network for the host device to connect to. This setup can be particularly useful for remote field systems, although it does require some terminal-based configuration. In situations where wireless or wired network options are unavailable, the Raspberry Pi offers several alternative connectivity solutions to directly interface with the device. One approach is to establish a direct computer-to-computer connection using an Ethernet cable.

3.12.1 Model Testing For Wearable Devices

Now for testing of model different subjects performed different activities in a real-time environment. The real-time data that is collected during activities is sent to the Pi where our model is deployed and running. We are loading trained model with a single line of code and passing real-time data from the model. We have evaluated the model by checking its accuracy. Model's accuracy is evaluated when the model predicting the same results that it was predicting when we have trained the model.

3.12.2 Model Implementation

In the context of fall detection using wearable and non-wearable devices, the workflow presented in the code is well-suited for building machine learning models to predict falls based on sensor data. Whether using accelerometers, gyroscopes, or other motion sensors, this process can be adapted to handle sequential data, particularly when using Long Short-Term Memory (LSTM) networks, which are effective at capturing temporal dependencies.

3.12.3 Scaling Features

The first step involves standardizing the input data using StandardScaler. This step ensures that the features from the wearable and non-wearable devices, such as accelerometer readings (accX, accY, accZ) and gyroscope measurements (gyrox, gyroy, gyroz), are scaled to have a mean of 0 and a standard deviation of 1. In the fall detection dataset, the readings can vary widely, depending on the intensity of movement or the orientation of the devices. Standardizing these features is essential, as it helps the LSTM network interpret the data more effectively and accelerates the convergence of the model during training.

3.12.4 Splitting Data

After scaling, the data is split into training and testing sets using train test split. In the fall detection dataset, this means dividing the sensor data collected during fall events (labeled as Fall) and non-fall events (labeled as No Fall) into two sets: one for training the LSTM model and one for evaluating its performance. This split ensures that the model can generalize to new, unseen data, which is critical for a real-time fall detection system. For wearable devices, such as those worn on the wrist or waist, the time series data captured during daily activities is included in the training set, while other segments are reserved for testing. For non-wearable devices like cameras or motion detectors, the corresponding sequences would be treated similarly.

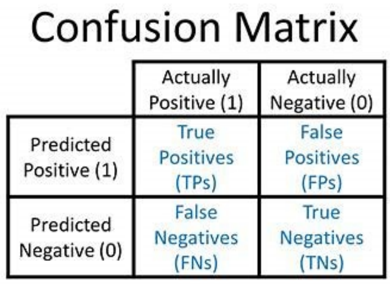
Chapter 4

Results and Evaluation

4.1 Introduction

In the methodology section we have proposed our approach for fall detection. We are comparing wearable devices with non-wearable devices. On the bases of results, we are concluding that which approach is better for Fall detection. We have dataset for both wearable and non-wearable. In case of wearable, we have made our experimental setup where we have Pi and server and a Samsung galaxy watch6. Subject is performing fall and non-fall activities. Data is saved by the model that is deployed in a Pi. At the end we are using Machine learning model i.e. Long short-term memory model for checking the accuracy of the model with wearable devices dataset on fall detection. In case of non-wearable, we are using base paper dataset. A dataset with 17O events is used. Each event has one of the following types human body models falling (40 records), chair falling (30 records) cellphone falling (30 records) water bottle falling (30 records) and noise environment (40 records). The dataset is recorded with a 100-millisecond period of readings with two accelerometers and one gyroscope. We have results for following parameters.

Confusion Matrix A confusion matrix is a tool used in machine learning and statistics to evaluate the performance of a classification model. It provides a detailed breakdown of how well the model predicts each class, helping to understand its accuracy beyond just a single performance metric.



Confusion Matrix

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)

Figure 4.1: Confusion Matrix

The matrix is typically a square table where:

True Positives (TP): The model correctly predicts the positive class. True Negatives (TN): The model correctly predicts the negative class. False Positives (FP): The model incorrectly predicts the positive class when it should be negative (also known as a "Type I error"). False Negatives (FN): The model incorrectly predicts the negative class when it should be positive (also known as a "Type II error").

4.1.1 Metrices Used for Evaluation

4.1.2 Accuracy:

Accuracy is a metric used to evaluate the overall performance of a classification model. It is defined as the ratio of correctly predicted instances (both true positive and true negatives) to the total number of instances in the dataset. Accuracy means how often model makes correct predictions

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

4.1.3 Precision

Precision is a metric used to evaluate the performance of a classification model, particularly in binary classification problems. It is defined as the ratio of true positive predictions to the total number of positive predictions made by the model

$$\text{Precision} = \frac{\text{TP}}{(\text{TP} + \text{FP})}$$

4.1.4 Recall

Recall is also known as sensitivity or true positive rate. It is a metric used to evaluate the performance of the classification model, particularly in binary classification problems. Recall measures the ability of the model to correctly identify all positive instances in the dataset.

$$\text{Recall} = \frac{\text{TP}}{(\text{TP} + \text{FN})}$$

4.1.4.1 F1-Score

The F1 score is an evaluation parameter that combines precision and recall into a single measure, balancing both accuracy and completeness, to evaluate a model's overall performance, especially in cases of imbalanced data.

$$\text{Recall} = \frac{\text{2xprecisionxRecall}}{\text{PrecisionxRecall}}$$

4.1.5 Results for Non-Wearable Devices

As mentioned earlier, for non-wearable devices, we took the base paper [1] results as such, and also used the same dataset with additional feature and tested the model's performance. The results of the base paper are given below:

i. The system achieved an accuracy of 92.44 percent, indicating a high proportion of correctly classified instances of falls and non-falls. ii. The precision of 94.6 percent reflects the proportion of true positive fall detections among all positive detections made by the system. iii.vThe F1 Score of 96.2 percent combines precision and recall into a single metric, showing a well-balanced performance between the two metrics.

iv. The recall of 97.8 percent shows the system's effectiveness in detecting actual fall events, capturing nearly 98 percent of the falls.

The paper demonstrates that the proposed IoT-based system performs well in detecting falls, leveraging sensor data and advanced algorithms to achieve these results. For a comprehensive understanding of the methodology and experimental setup, you can refer to the full text of the paper. We then performed experiments on the base paper dataset added with additional features as mentioned in the section (3.2.4). The results obtained with these additional features are given below:

i. Accuracy: 1.0: This means the model achieved a perfect accuracy score of 100 percent the test set, correctly classifying all test samples. ii. Precision: 1.0: Precision is also 100 percent, indicating that every positive prediction made by the model was correct, with no false positives. iii.Recall: 1.0: Recall is 100 percent, meaning the model identified all actual positive cases correctly, with no false negatives. iv Since the precision and recall both are 100 percent, as a result the F1 Score is also 100 percent.

The 100 percent performance in all metrices is an alarming situation, as there are strong chances that the system is overfit. In order to remove this doubt, we performed our experiments multiple times using k-fold cross validation setting the value of k from 3, 5 and 10. Model repeated the same results which cannot be truly claimed unless the system is deployed in the real time. As mentioned already, that we could not perform these experiments in real time due to technical issue, so we quote these results with an element of doubt as the other approach has been tested in real time

4.1.6 Results on Wearable Dataset

When we deployed the model that is trained using the published dataset, that is, the k-fall dataset, then the system did not give very encouraging results, that is, the performance of the system was less than 70 percent We performed our experiments multiple times but the results were not improved. Our analysis of these results showed that the model trained on the k-fall dataset is not rightly compatible with the environment in which we were performing the experiments. Although the sensors being used were same, that is, accelerometer and gyroscope, however, there was a major different in the placement of the sensors, that is, the sensor in case of k-fall dataset were placed on the waste of subjects. This gives two major characteristics to the dataset, that is, the direction of the sensors remains static which means that the x, y and z coordinates of the sensors always give the same axes. Secondly, the movement of the waste generally depicts the movement of the body itself. On the other hand, our experiments were being performed when the watch containing the sensors was placed on the wrist which on one hand does not always represent the overall movement of the body. Secondly, wrist (or arm) moved much more frequently as compared to the waste and movement of the arm does not show always show the movement of body. So overall it means that data being generated while the sensors are placed on waste and while they are on the wrist are not much compatible. This led us to capture and label the data from the watch, train the model on this data and then use the same model in the real time. The performance of LSTM model on wearable dataset that we collected through our experiments is as follows:

i. The value of accuracy is 97.44 percent of all instances, showing overall effectiveness in distinguishing between falls and non-falls. ii. Precision reflects the accuracy of positive predictions. A precision of 95.12 percent indicates that when the model predicted a fall, it was correct 95.12 percent of the time, demonstrating strong performance in minimizing false positives.

iii. Recall measures the model's ability to identify actual falls. A recall of 75.97 percent means that the model successfully detected about 76 percent of the true fall events, though it missed some, as indicated by the false negatives.

iv F1 score reflects how well the model identifies falls while minimizing both false positives and false negatives. Model shows 84.4 F1 score which indicates that the model is making accurate predictions in both identifying falls and avoiding false alarms.

v.The overall performance of the system was around 50 percent which was not satisfactory. The main error being that the system was mixing the activities, specially, Walk and Fall. Such performance is obviously not acceptable, as the system has to generate alarm on the basis of Fall detection, that is, it would alert the attendant of the subject (elderly person in our case), and many of our Fall alerts were false.

vi By observing and plotting the values generated by the system, we realized that the sensors generate values showing abrupt or high movement in cases where the object moves fast and when it falls. At this point, system mixes Walk and Fall. So the model was unable to trace here the difference between the two activities.

vii We, in particular, observed the value of AccG, that is the magnitude of accelerometer values. We then realized following ranges of AccG values associated with different activities a. Still: 0 – 9.5 b. Walk: 8.5 – 14.6 c. Fall: 13.8 and higher The ranges observed show overlapping between different activities and that being the main reason for misjudgment of the system

viii Another reason for misjudgment of the model was observed that the subjects involved in the experimentation were mainly young. Their movement, in general, was fast specially while walking. This quick movement generated such values of the sensors that resemble the fall. ix This point was addressed by asking the subjects to move slow while performing the activities. This improved our results and the performance approached70x. Since the focus of our work is mainly the fall detection, so we shifted our system from activity recognition to fall detection only, that is, we let the system work as such, however, we combined the Still and Walk into 'Non-fall' and let the 'Fall' remain as such. We also asked subjects to move slow to actually depict the movement of an elderly. xi. These strategies improved the performance of the system a lot and overall performance of the system crossed 90xii With a casual movement of the subject, system generated some false alarms, however, it did not miss even a single Fall. That is, the true positive of the system was 100

This experimentation on one side gave us results that were reasonably good, on the other side it highlighted the issues associated with such a setup that will help us to improve this system and also help to implement in a better way in the real life where it is going to be actually implemented on elderly people

4.1.7 Comparison of the Base Paper Results with LSTM

This section presents an overall comparison of all our experiments that lead to our final conclusion, Table 4.1 below presents a comparison of all results. We present these results agreeing that the results are not really comparable because we could not create a common setup where they could be really comparable, still these results present an overall picture.

Table 4.1: Parameters for Wearable and Non-Wearable Devices

Parameters	**Base Paper for NW**	**NW with extra features**	**Training Results**	**Real-time Multiclass**	**Real-time Binary**
Accuracy	92%	68%	97.44%	83%	94%
Precision	92%	36%	95.12%	83.5%	90%
Recall	92%	100%	75%	83.33%	90%
Specificity	-96.2	100%	84%	80%	90%

In the table above, second column shows the results of non-wearable (NW) devices taken from base paper as such. The third column gives results obtained during training the model with data from base paper along with 4 new features that were computed from the base data. Although we performed different measures to remove the doubt of overfitting, still the results remain the same. This model however has not be tested in real-time. The fourth column shows the performance of model during training on the data that we have captured through our own experiments. The second last column shows results of our experiments with multi-class prediction.Multi class refers to walking, still and fall activities. Here results are not that much encouraging since the model mixes different categories. The last column gives result of our experiment in real-time with a binary class prediction. These results are very promising that endorse the validity of our experiments for capturing data and for the fall detection.

Chapter 5

Conclusion and Future Work

Human fall detection has important role in life of the people especially for elderly people. Most of the devices using wearable devices for fall detection because they are considerable better approach as compared to non-wearable devices. We are comparing wearable and non-wearable devices for fall detection. Our approach is to collect our own dataset and test and train the machine learning model. For comparison we are also applying machine learning model to the base paper dataset for non-wearable devices. We have made our experimental setup for data collection in which we have raspberry Pi, server, Samsung galaxy watch6. Watch has inbuilt accelerometer and gyroscope sensors that are reading values from the subject's activities. We have recorded activities for walk, still and non-fall. We have extracted features like the magnitude of some vector, magnitude of sum vector on horizontal plane, angle between z-axis and horizontal, rate of change of acceleration and standard deviation magnitude on horizontal plane. We are giving all this data which is x-axis, y-axis and z-axis of accelerometer and gyroscope to the LSTM model and checking its accuracy, precision and Recall. We have base paper dataset we are extracting features on base paper dataset and checking the results. At the end we are comparing our results of wearable devices with the results we have checked for non-wearable devices data set. We are also comparing our results with the results of base paper and the results we have for non-wearable dataset. We have concluded that LSTM model is giving better accuracy in our collected dataset on wearable devices.

Bibliography

[1] O. Ribeiro, L. Gomes, and Z. Vale, "IoT-based human fall detection system," *Electronics*, vol. 11, no. 4, p. 592, 2022. [Online]. Available: https://doi.org/10.3390/electronics11040592.

[2] D. Yacchirema, J. S. De Puga, C. Palau, and M. Esteve, "Fall detection system for elderly people using IoT and big data," *Procedia computer science*, vol. 130, pp. 603–610, 2018. [Online]. Available: https://doi.org/10.1016/j.procs.2018.04.110

[3] . A. Purwar and I. Chawla, "A systematic review on fall detection systems for elderly healthcare," *Multimedia Tools and Applications*, vol. 83, pp. 433 277–43 302, 2019. [Online]. Available: https://doi.org/10.1109/ACCESS.2019.2906693

[4] F. K. K. A. A. J. I. C. Anwar, U. Hassan and H. Elmannai, "A Cost- Effective Fall-Detection Framework for the Elderly Using Sensor-Based Technologies," 2023.

[5] V. Kumar, N. Badal, and R. Mishra, "Elderly fall detection using IoT and image processing," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 24, no. 3, pp. 681–695, 2021. [Online]. Available: https://doi.org/10.1080/09720529.2019.1692451

[6] M. Al-Khafajiy, T. Baker, C. Chalmers, M. Asim, H. Kolivand, M. Fahim, and A. Waraich, "Remote health monitoring of elderly through wearable sensors," *Multimedia Tools and Applications*, vol. 78, no. 17, pp. 24 681–24 706, 2019. [Online]. Available: https://doi.org/10.1007/s11042-018-7134-7

[7] S. Nazir, Y. Ali, N. Ullah, and I. García-Magariño, "Internet of things for healthcare using effects of mobile computing: a systematic literature

review," *Wireless Communications and Mobile Computing*, vol. 2019, no. 1, p. 5931315, 2019. [Online]. Available: https://doi.org/10.1155/2019/5931315

[8] J. E. X. Wang and G. Azzopardi, "Elderly fall detection systems: A literature survey," *Frontiers in Robotics and AI*, vol. 7, 2020. [Online]. Available: https://doi.org/10.3389/frobt.2020.00071

[9] . X. Zhou, "Wearable health monitoring system based on human motion state recognition," *Computer Communications*, vol. 150, pp. 826–829, 2020. [Online]. Available: https://doi.org/10.1016/j.comcom.2019.11.008

[10] A. S. I. A. Faisal, "T. w. purboyo and r. ansori, "a review of accelerometer sensor and gyroscope sensor in imu sensors on motion capture," *Journal of Engineering and Applied Sciences*, vol. 15, pp. 826–829, 2020.

[11] D. Yacchirema, "Fall detection system for elderly people using IoT and ensemble machine learning algorithm," *Procedia Computer Science*, 2019.

[12] R. F. S. Y. N. Mozaffari, J. Rezazadeh and K. Sandrasegaran, "Practical fall detection based on iot technologies: A survey," internet of things (netherlands)," *Internet of Things (Netherlands)*, 2019.

[13] H. Li, A. Shrestha, H. Heidari, J. Le Kernec, and F. Fioranelli, "Bi-LSTM network for multimodal continuous human activity recognition and fall detection," *IEEE Sensors Journal*, vol. 20, no. 3, pp. 1191–1201, 2019. [Online]. Available: https://doi.org/10.1109/JSEN.2019.2946095

[14] J. W. Jolles, "Broad-scale applications of the Raspberry Pi: A review and guide for biologists," *Methods in Ecology and Evolution*, vol. 12, no. 9, pp. 1562–1579, 2021. [Online]. Available: https://doi.org/10.1111/2041-210X.13652

[15] A. Choi et al, "Deep learning for fall detection: Three-dimensional cnn combined with lstm on video kinematic data," *IEEE*, 2022.

[16] A. Alarifi and A. Alwadain, "Killer heuristic optimized convolution neural network-based fall detection with wearable IoT sensor devices," *Measurement*, vol. 167, p. 108258, 2021. [Online]. Available: https://doi.org/10.1016/j.measurement.2020.108258

[17] V. H. F. K. A. Ovaska, S. J., "Fusion of soft computing and hard computing in industrial applications: an overview. ieee transactions on systems, man and cybernetics, part c (applications and reviews," *IEEE*, 2017.

[18] Li, X., Pang, T., Liu, W., Wang, T., "Fall detection for elderly person care using convolutional neural networks," *IEEE*, 2018.

[19] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "Deep learning for fall detection: Three-dimensional cnn combined with lstm on video kinematic data," *IEEE*, 2022.

[20] C. P. D. Yacchirema, J. S. De Puga and M. Esteve, ""practical fall detection based on iot technologies"," *Internt of Things*, p. 100124, 2019.

[21] J. D. L. A. Sucerquia and J. F. Vargas-Bonilla, "SisFall: A fall and movement dataset," *Electronics*, vol. 11, no. 4, p. 592, 2022. [Online]. Available: https://doi.org/10.3390/electronics11040592

[22] Bet, P., Castro, P. C., Ponti, M. A. (2019), "Fall detection and fall risk assessment in older person using wearable sensors: a systematic review. international journal of medical informatics," *IEEE*, 2019.

[23] G. L. Santos, P. T. Endo, K. H. d. C. Monteiro, E. d. S. Rocha, I. Silva, and T. Lynn, "Accelerometer-based human fall detection using convolutional neural networks," *Sensors*, vol. 19, no. 7, p. 1644, 2019. [Online]. Available: https://doi.org/10.3390/s19071644

[24] W. Fall and D. Systems, "Ss symmetry wearable fall detection systems," *Symmetry*, 202O.

[25] J. W. Jolles, "Broad-scale applications of the raspberry pi: A review and guide for biologists," *Methods in Ecology and Evolution*, vol. 12, pp. 1562–1579, 2019.

[26] C. W. Zhao, "Exploring iot application using raspberry pi," *International Journal of Computer Networks and Applications*, vol. 2, 2015.

[27] D. Kraft, K. Srinivasan, and G. Bieber, "Deep learning based fall detection algorithms for embedded systems , smartwatches , and iot devices using

accelerometers," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, pp. 2385–2394, 2022.

[28] E. D. Engineer, A. Solutions, and A. Solutions, "Tcp / ip based cruise control in traffic situation," *Journal of Engineering and Applied Sciences*, 2021.

[29] J. W. Jolles, "road-scale applications of the raspberry pi: A review and guide for biologists," *Methods in Ecology and Evolution*, pp. 1562–1579, 2021.

[30] M. Salman Khan, M. Yu, P. Feng, L. Wang, and J. Chambers, "An unsupervised acoustic fall detection system using source separation for sound interference suppression," *Signal Processing*, pp. 199–210, 2015.

[31] X. Yu, J. Jang, and S. Xiong, "A large-scale open motion dataset (kfall) and benchmark algorithms for detecting pre-impact fall of the elderly using wearable inertial sensors," *Frontiers in Aging Neuroscience*, vol. 13, pp. 1–14, 2021.