Threshold Based Fall Detection using Mobile Sensors

December 12, 2024

CSE - 518 Final Course Project

Dhanush Chellamuthu
Computer Science,
Stony Brook University
dchellamuthu@cs.stonybrook.edu

Abstract

Falls are a serious health hazard, especially for the elderly and physically challenged. This project offers a threshold based fall detection that makes use of mobile phone. We compute Signal Vector Magnitude (SVM) and vertical acceleration using the magnetometer, accelerometer, and gyroscope sensors found in smartphones to detect abrupt movements that could signal a fall. Real-time fall detection is achieved by analyzing these metrics using a threshold-based method. When the mobile application detects a fall, it instantly notifies select emergency contacts; users can choose to ignore false alarms.

The system prioritizes user-friendly interaction to make it easier for senior citizens to utilize. It also stores fall activity, giving medical specialists useful data to evaluate physical health, fall frequency, and mobility patterns. The application may significantly reduce the number of fall-related injuries in vulnerable populations by ensuring prompt medical intervention, increased safety for those who are at risk, and improved inclusivity in human-computer interaction.

1 Introduction

Falls are a serious health hazard, especially for the elderly and people with physical ailments. According to the Centers for Disease Control and Prevention (CDC), one in four older adults aged 65 and above experience fall every year, with more than 14 million falls reported yearly [4]. The impact of falls extends beyond acute physical impairment, usually leading to decreased mobility, fear of falling, social isolation, and a decline in quality of life [14]. The consequences of falls highlight the vital necessity for prompt action and quick reactions. It can minimize injury severity, be life saving and reduce then significant healthcare expenses associated with fall-related occurrences. Hence, early detection of falls is essential for timely medical attention, which could prevent fatalities and serious injuries.

Conventional fall detection approaches, use ambient sensors or visual signals, that has drawbacks such as privacy issues, environmental restrictions, expensive installation and maintenance costs. In this paper, I have designed and implemented a prototype that offers the adoption of smartphones and implement a threshold-based fall detection. I developed an application in a smartphone as it takes advantage of their widespread use to offer an affordable and easily accessible alternative to detect falls. Current algorithms for threshold-based fall detection usually use

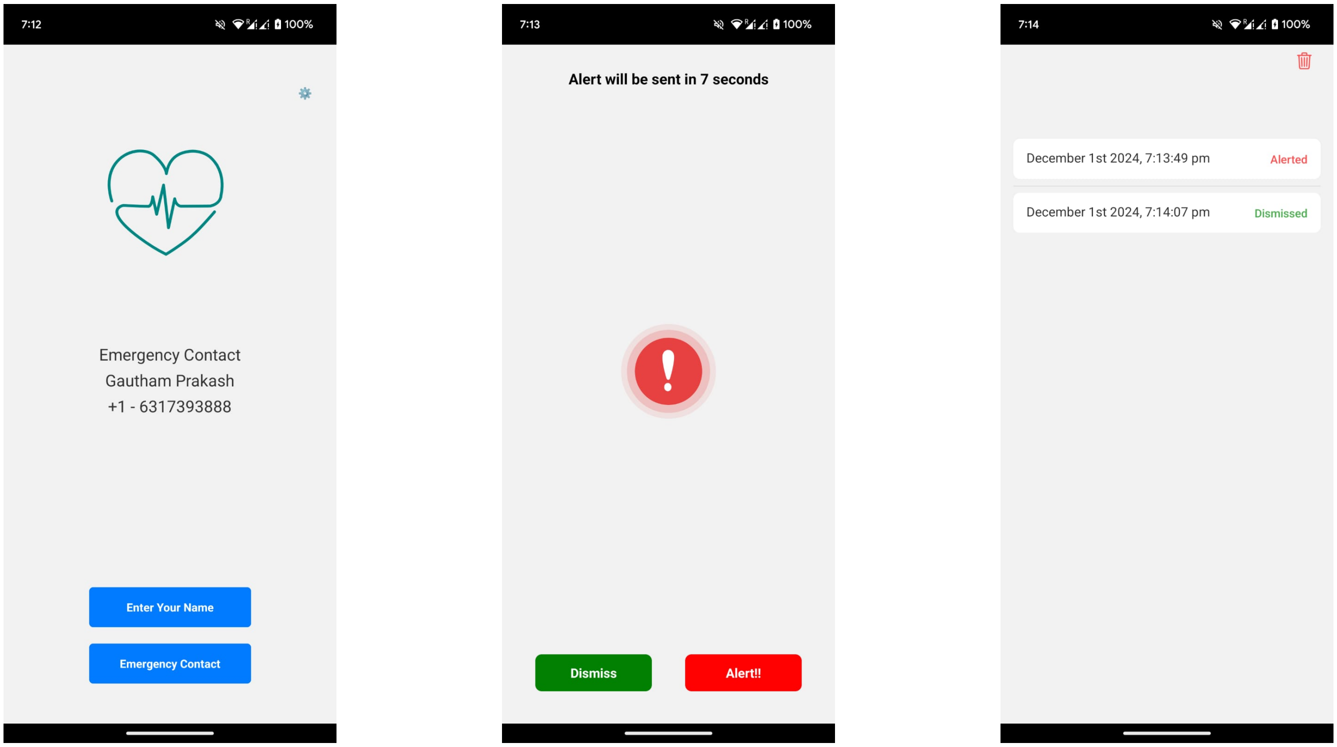


Figure 1: Screens from the mobile application

accelerometer data to spot abrupt shifts in movement patterns that could be signs of a fall. But this study utilizes both accelerometer data and magnetometer data which can be used calculate vertical acceleration that can be used for fall detection. This paper involved a user study where I have collected data for common activities such as standing up, sitting down, running, transitioning from running to walking, and falling to develop a threshold-based algorithm that recognizes falls. The collected data are subsequently used to analyze and find threshold value to be used for the fall detection algorithm.

The gadget acts as a body-attached sensor that records motion data while a person engages in different activities and it should be placed in the left or right pocket. The program leverages the Root Mean Square Error (RMSE) between vertical acceleration and overall acceleration magnitude, which are computed using the sensor data. The RMSE value helps us to detect possible falls. This method offers a simple and reliable solution for the management of fall risk by enabling precise fall detection while reducing false positives from daily movements.

2 Related Work

For fall detection, the earlier studies mostly focused on visual cue-based systems. This approach utilize cameras and computer vision algorithms to track and interpret human movements. They mostly use 3-D modeling approaches to detect falls [13]. For example, some papers have investigated the use of depth cameras such as the Microsoft Kinect to record depth information and recognize falls [9]. But these visual cue-based systems frequently face obstacles due to the requirement for specialized equipment such as depth cameras, it may have privacy concerns, and is limited in a variety of scenarios.

Environment cue-based system is a different technique that is used to detect falls [11]. This approach uses a variety of ambient sensors, which involve vibration and audio detectors, to collect movement data and analyze fall patterns [11, 3]. Cheffena [6] presents a fall detection system based on analyzing the acoustic elements acquired by a smartphone in a household setting. Even-though, it may be less obtrusive, environment-based systems usually have limits in responding to changing surroundings, the system has limited coverage, and can be expensive to set up and operate [9].

The use of smartphones is widespread, and various research work have focused on investigating the possibility of mobile sensors to recognize falls [7]. Smartphone wearable device-based fall detection systems provide an accessible and customizable solutions [12]. Smartphones' built-in accelerometers and magnetometers can be used to collect motion data, which gives insights on movement patterns and probable fall incidents [1].

The accuracy with which these mobile sensors can detect falls has been studied. Albert et al. [2] used machine learning methods utilizing mobile phone sensor data to efficiently classify falls. Another study, Medrano et al. [10], investigated a novel way to detect falls by classifying them as "novelties" in acceleration patterns recorded from smartphones, with promising results. The use of threshold-based algorithms has emerged as a popular strategy for mobile sensor-based fall detection [8]. These algorithms use specialized parameters retrieved from accelerometer and magnetometer data, such as Signal Vector Magnitude (SVM) and vertical acceleration, to distinguish falls from other everyday activities [5]. The selection of appropriate thresholds for these parameters is critical to ensuring accurate fall detection and reducing false alarms [15].

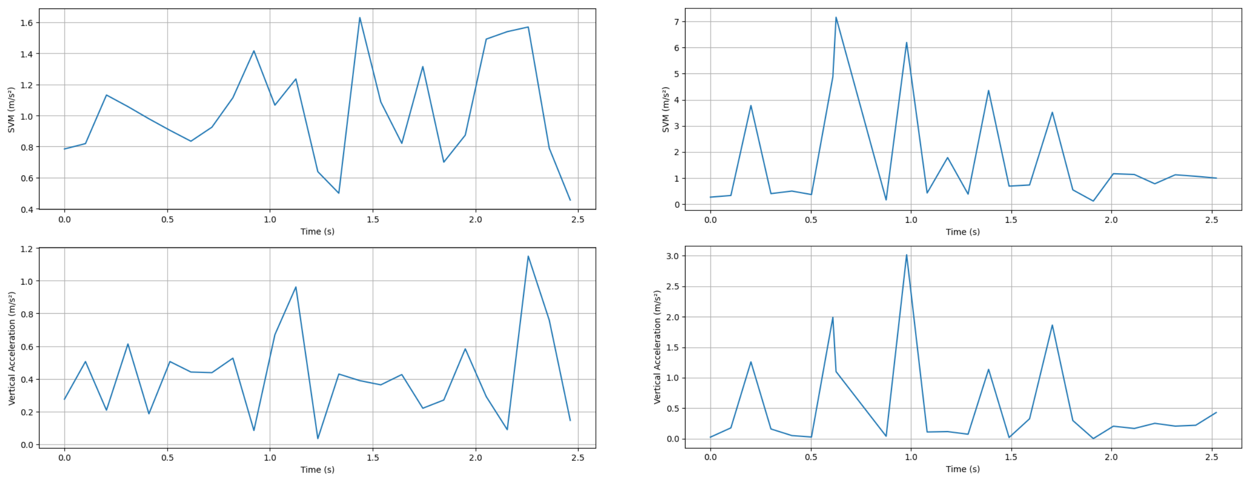
Despite various advancements in the mobile-based fall detection system, there still remain many challenges. Optimization of battery and having seamless integration with other smartphone functionalities are certain hurdles in developing a proper fall detection application. There are research efforts made towards developing energy-efficient algorithms, which utilize sensor data in an intelligent way to extend battery life.

3 User Study: Understanding Fall Detection Features from Mobile Sensors

This project aims to carry out a comprehensive study to examine and comprehend the features associated with various human activities, such as falling, running, walking, and transitioning from running to walking. Through this user study we aim to find features that could be extracted from the data gathered, establish suitable thresholds for fall detection, and differentiating fall incidents from other human activities. The study aims to provide important insights into the distinct traits and patterns connected to each movement type by examining the sensor data collected from mobile devices during these activities.

3.1 Experiment Setup, Apparatus and Participant

I am the sole participant in the experiment (24 years old), and I have used a Pixel 6a smartphone as the main data collection tool. I used React Native to create a custom mobile application that recorded accelerometer and magnetometer measurements for every timestamp in order to make data retrieval easier. Over the course of roughly 2.5 seconds, the application collected around 25 data points for each of the six different activities: falling, running, walking, switching from walking to running, sitting down, and standing up. A total of twelve recordings were available for analysis because each task was completed twice. This dataset is then used in next section to learn



(a) Walk Activity: Change in vertical acceleration and SVM w.r.t time

(b) Run Activity: Change in vertical acceleration and SVM w.r.t time

Figure 2: Walk and Run Activities

more about identifying falls and distinguishing them from other human activities. Each record has accelerometer data a_x, a_y, a_z and magnetometer data m_x, m_y, m_z and timestamp.

As part of the experimental setup, the subject carried a Pixel 6a smartphone in his left or right pocket. Sensor data was continuously recorded by mobile sensors, the react native application uses 'expo-sensors' package to obtain data. The activities included running and walking on a level surface for about 10 meters, and falling on a padded surface to detect fall.

3.2 Algorithm Features

1) Peak SVM value: SVM(Signal Vector Magnitude) is calculated as shown in equation (1). The mobile sensors provide A_x, A_y and A_z from which we calculate the total acceleration magnitude. The peak values help us determining if there is any sudden movement.

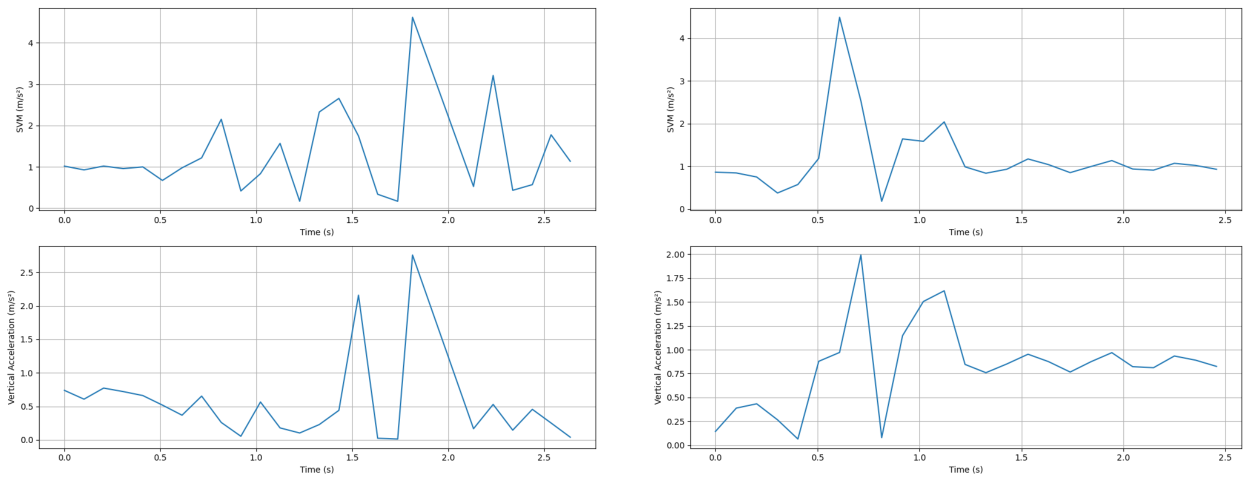
$$\text{SVM} = \sqrt{A_x^2 + A_y^2 + A_z^2} \tag{1}$$

2) Vertical Acceleration : Vertical acceleration is very similar compared to fall's SVM data which can be used to detect fall activities primarily. Vertical acceleration $|A_v|$ is calculated using sensor data from accelerometer and magnetometer data. Where θ_y and θ_z denote its pitch and roll values. The equation (2) shows how to calculate Vertical acceleration.

$$|A_v| = |A_x \sin\theta_z + A_y \sin\theta_y - A_z \cos\theta_y \cos\theta_z| \tag{2}$$

3.3 Activity Analysis

1) Walk: From the figure(2.a) we can see that the SVM peaks at $1.4th$ second with value $1.6m/s^2$ and vertical acceleration peaks at $2.25th$ second with value as 1.17. The peaks do not correspond. They are different because walk activity has both vertical and horizontal acceleration.



(a) Run to Walk Activity: Change in vertical acceleration and SVM w.r.t time

(b) Fall Activity: Change in vertical acceleration and SVM w.r.t time

Figure 3: Run to Walk and Fall Activities

2) Run: From the figure(2.b) we can see that the SVM peaks at $7m/s^2$ at 0.65 seconds and the vertical acceleration peaks at $3m/s^2$ at 0.95 seconds. From this we can see that the peaks don't corresponding. This could be because running involves both vertical and horizontal acceleration in disproportionate magnitudes.

3) Run to Walk: From the figure(3.a) we can see that the SVM peaks at $4.4m/s^2$ and there are multiple peaks. The vertical acceleration seems to be similar when comparing it to SVM data. But while comparing vertical acceleration and SVM data we can see that at certain peaks values differ.

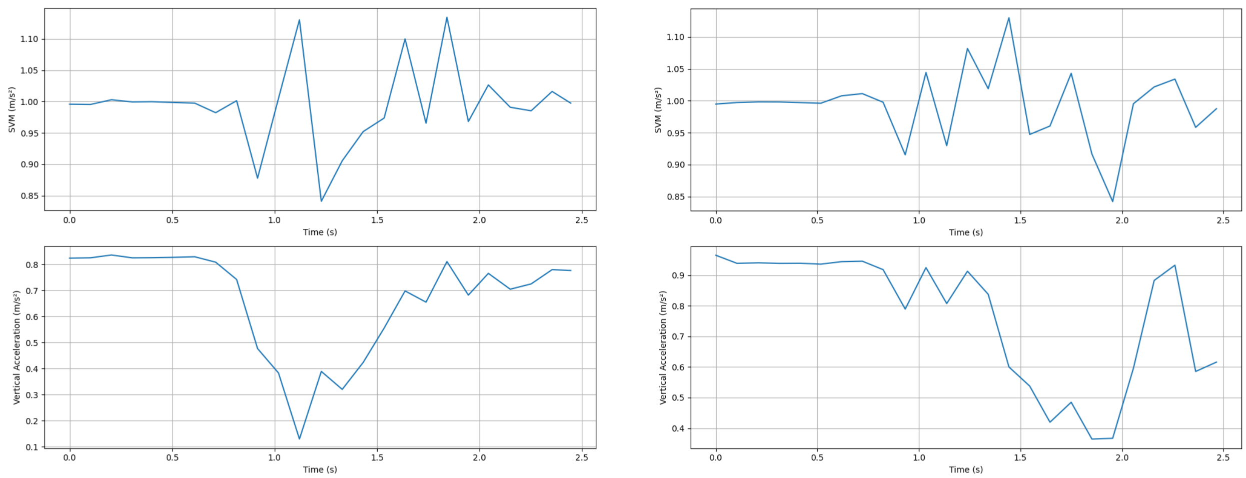
4) Fall: From the figure(3.b) we can see that the SVM peaks at $4.3m/s^2$ and there are multiple peaks. The vertical acceleration seems to be similar compared to SVM data. We observe that peak occurs at $0.6th$ second which corresponds to the fall event. The next peak at $1.15th$ second corresponds to the bounce back event. In real life the bounce back event will have much lesser acceleration compared to this data recorded. It is because this event was recorded for fall event on a padded surface.

5) Sit Down: From the figure(4.a) we can see that the SVM peaks only at $1.14m/s^2$ and there are multiple peaks. The vertical acceleration is not similar compared to SVM data. The graphs are in opposite direction

6) Stand up: From the figure(4.b) we can see that the SVM peaks only at $1.14m/s^2$ and there are multiple peaks. The vertical acceleration is not similar compared to SVM data.

In the examination of the SVM feature done above, we could observe that sitting down, standing up and walking do not exceed the limit of $4m/s^2$. Whereas other activities such as 'run to walk' and 'run' show that the SVM values exceed the threshold of $4m/s^2$, but a comparison of the similarity in vertical acceleration and SVM signal vector magnitude indicate that these activities appear to be less similar. From table 1 we can say that the RMSE for fall will be much smaller in real life as fall usually occurs on hard surface and the variation in bounce back event will small. So from the analysis I have used two threshold value: $3.5m/s^2$ to detect any abrupt fast motion and a RMSE threshold of 0.5 to detect fall from other activities. The RMSE is calculated as shown in equation(3).

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(SVM_i - Th_v[i])^2} \quad (3)$$



(a) Sit down Activity: Change in vertical acceleration and SVM w.r.t time

(b) Stand up Activity: Change in vertical acceleration and SVM w.r.t time

Figure 4: Sit down and Stand up Activities

Activity	MAX SVM (m/s^2)	RMSE(SVM, $\|A_v\|$) (m/s^2)
Walk	1.61	0.6
Run to Walk	4.50	0.76
Run	7.00	3.150
Stand Up	1.13	0.23
Sit Down	1.13	0.32
Fall	4.20	0.37

Table 1: table

3.4 Algorithm

The mobile application executes an algorithmic approach that combines mobile sensor data analysis with real-time fall detection capabilities. The application continuously operates in a monitoring cycle, by using magnetometer and accelerometer mobile sensors data to detect potential fall event while integrating user interaction and emergency response setup.

The application's home screen has the option for the user to enter their name and emergency contact details. After the user has entered the details the main fall algorithm begins to execute by tracking the accelerometer and magnetometer sensor data within precisely defined time frame of 2.5 seconds, which has window of 25 values of each sensor data. This window was selected to capture any significant detail needed for the human activities. Initially the algorithm calculates the acceleration magnitude or the Signal Vector Magnitude and monitors if it crosses a certain threshold. When the middle record crosses a threshold of $3.5m/s^2$, the algorithm moves to the next step of calculating the vertical acceleration. During this step, for the time frame when fall was detected the mobile sensor data is being used to calculate the RMSE between SVM and Vertical Acceleration. This dual step process help us to detect fall, while reducing false positives and maintaining high sensitivity to actual fall events.

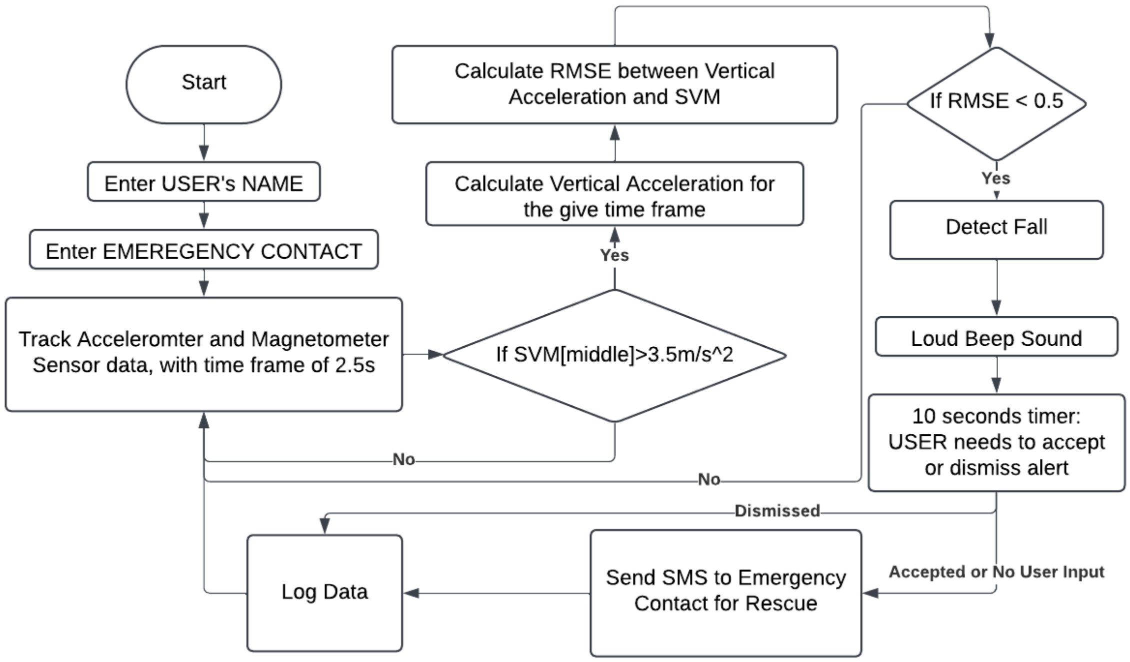


Figure 5: Fall Detection Algorithm

Once fall is detected, the application triggers a loud beep sound to notify the user. In this alert phase a pop-up with 10 seconds timer is initiated. During which the user has to either confirm or dismiss the fall detection. If the user clicks confirm or does not click dismiss within 10 seconds an alert is sent to the emergency contact. This feature is crucial to minimize false alarms while for unconscious users alert is sent automatically. The alert is sent as a SMS to emergency contact registered before. The fall event data is then finally recorded with timestamp and whether an alert was sent or not.

4 Future Work

There are certain drawback as the application is not 100% accurate and can falsely detect falls for activities that have similar SVM and vertical acceleration. The current threshold-based method is promising, but there are a number of directions that can be worked on to improve fall detection accuracy and user experience.

1) Machine Learning and Deep Learning Approaches: Algorithms based on machine learning (ML) or deep learning (DL) may be more accurate than threshold-based techniques. By automatically analyzing intricate patterns from sensor data, these methods may be able to identify subtle distinctions between falls and daily activities that could be overlooked by preset criteria. Techniques for supervised learning, such as Random Forests or Support Vector Machines (SVM) can be used for classifying falls. Convolutional neural networks (CNN) and long short-term memory (LSTM) networks are examples of deep learning architectures that can be used for temporal sequence analysis of sensor data.

2) Expanded Data Collection: Increasing the quantity and diversity of training data is critical for creating more reliable fall detection systems. Future work should focus on gathering more real-life fall data from elderly people in their natural settings. We can increase the number of participants to capture a broader range of fall patterns and activities of daily living, including data from people with various physical ailments or mobility disabilities.

3) Sensor Fusion Incorporating data from extra smartphone sensors may provide more contextual information for fall detection. Gyroscope data is used to better capture rotational movements. Barometer readings are used to identify altitude changes. To increase functionality. GPS data can be utilized to determine the user's position and surroundings.

4) Adaptive Algorithms It may be possible to increase accuracy and decrease false alarms by creating algorithms that can gradually adjust to the movement patterns of certain users. This might entail online learning strategies to continuously update the detection model and customized thresholds that change according to user behavior.

5 Conclusion

This course project showcases how smartphone capabilities can be leveraged to develop a threshold-based fall detection application. The project utilizes data collected from built-in accelerometer and magnetometer sensors to analyze key feature such as Signal Vector Magnitude(SVM) and vertical acceleration, to detect fall from other daily activities. In the application, we have seen that a dual-threshold approach to detect fall was implemented. Where, the SVM threshold is set at $3.5m/s^2$, which could help identify aburpt movement and RMSE threshold is set at 0.5 to distinguish fall from other rapid movement activities like run, run to walk etc. The application also provides a user- friendly interface for users from any background. The pop-up to dismiss fall event provides a way to remove unnecessary fall alerts to emergency contact as well. The app also monitors and keeps track of all fall activity, which can later be used by medical professionals to get an idea of the user's mobility patterns, frequency of falls, and overall physical health. This data can assist in diagnosing underlying conditions, adjusting treatment plans, and identifying trends that may indicate deteriorating mobility or increased fall risk.

This smartphone sensor based threshold system proves that it offers an accessible, practical, and potentially life-saving tool for elderly individuals and those with mobility limitations, enhancing their safety and independence in daily life.

References

[1] Stefano Abbate et al. "A smartphone-based fall detection system". In: *Pervasive and Mobile Computing* 8.6 (2012).

[2] Mark V Albert et al. "Fall classification by machine learning using mobile phones". In: *PloS one* 7.5 (2012).

[3] Majd Alwan et al. "A smart and passive floor-vibration based fall detector for elderly". In: *2006 2nd International Conference on Information & Communication Technologies* 1 (2006).

[4] Gwen Bergen, Mark R Stevens, and Elizabeth R Burns. "Falls and fall injuries among adults aged 65 years—United States, 2014". In: *Morbidity and Mortality Weekly Report* 65.37 (2016).

[5] Alan K Bourke, John V O'Brien, and Gerard M Lyons. "An evaluation of thresholds for detecting falls using acceleration data from a waist mounted tri-axial accelerometer". In: *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (2007).

[6] Michael Cheffena. "Fall detection using smartphone audio features". In: *IEEE journal of biomedical and health informatics* 20.4 (2016).

[7] Jiangpeng Dai et al. "Mobile phone-based pervasive fall detection". In: *Personal and ubiquitous computing* 14.7 (2010).

[8] Maarit Kangas et al. "Comparison of low-complexity fall detection algorithms for body attached accelerometers". In: *Gait & posture* 28.2 (2008).

[9] George Mastorakis and Dimitrios Makris. "Fall detection system using Kinect's infrared sensor". In: *Journal of Real-Time Image Processing* 9.4 (2014).

[10] Carlos Medrano et al. "Detecting falls as novelties in acceleration patterns acquired with smartphones". In: *PloS one* 9.4 (2014).

[11] Muhammad Mubashir, Ling Shao, and Luke Seed. "A survey on fall detection: Principles and approaches". In: *Neurocomputing* 100 (2013).

[12] Ahmet Turan Ozdemir and Billur Barshan. "Wearable device based fall detection using minimum instance based learning". In: *2014 22nd Signal Processing and Communications Applications Conference (SIU)* (2014).

[13] Caroline Rougier et al. "Robust fall detection system using a 3D head trajectory". In: *2011 5th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth) and Workshops* (2011).

[14] Laurence Z Rubenstein. "Falls in older people: epidemiology, risk factors and strategies for prevention". In: *Age and ageing* 35.suppl_2 (2006).

[15] Tal Shany et al. "Sensor-based fall detection systems: A review". In: *Journal of biomedical informatics* 45.3 (2012).