Journal of Emerging Technologies and Industrial Applications
Vol. 1 No. 2 (2022) pp. 1-13
e-ISSN: 2948-507X
© Malaysia Board of Technologists (MBOT)
Journal homepage: https://jetia.mbot.org.my/index.php/jetia/index

Development of a Wearable Human Fall Detection System

Chua Kean Hong, Sai Siong Jun, Chan Chi Yen, Noor Ain Kamsani, Azura Che Soh*, Raja Mohd Kamil Raja Ahmad

Department of Electrical & Electronic Engineering, Faculty of Engineering,
Universiti Putra Malaysia, 43400, Serdang, Selangor, MALAYSIA

Email: *azuracs@upm.edu.my

Abstract: Fall contributes for over 80% of injury-related hospitalisation, especially amongst the elderly. A long lie due to a fall may lead to other health complications if medical intervention does not take place in a timely manner. Hence, a fall detection system that employs a wearable detector is important to detect the fall automatically. In this study, an accelerometer and gyroscope sensors were installed in a wearable fall detector. A fall detection algorithm was developed using MATLAB. This algorithm will extract features from the signal sent by the sensors and conduct a series of decision-making and classification process to determine whether a fall has actually occurred. The accuracy of the fall detection algorithm was determined to be at 98.41%. The detected fall will be notified to the elderly person's smart phone via Bluetooth and the smart phone will send an emergency message to the caregiver's smart phone via the Google Cloud Messaging (GCM) system. The smart phone will also update the database, with regards to the fall event. This system embraces the Internet of Things (IoT), big data, and data analytics concepts as well as allows data-driven healthcare to be developed.

Copyright © 2022 MBOT Publishing.
All right reserved.

Received 15 June 2022; Accepted 10 October 2022; Available online 15 December 2022

Keywords: Fall Detection, Wearable Device, Threshold-based Method, Internet of Things (IoT), Database, Big Data, Data Analytics, Google Cloud Messaging (GCM).

*Corresponding Author:

Azura Che Soh
Department of Electrical & Electronic Engineering, Faculty of Engineering,
Universiti Putra Malaysia, 43400, Serdang, Selangor, MALAYSIA.
Email: azuracs@upm.edu.my

1. Introduction

By the year 2040, Malaysia is expected to have a significant number of aged populations. In less than 20 years, there will be three senior citizens for every 20 Malaysians [1]. According to the World Health Organization, about 28-35% of people aged 65 and above experience fall each year [2]. Falls are the leading cause of injury and accidental death in those 75 years of age and older, accounting for 70% of accidental deaths [3]. Falls contribute for over 80% of injury-related hospitalisation of people older than 65 years [4]. Common injuries sustained from falls include soft and connective tissue damages, bone fractures, and head injuries. Other effects of falling at this age include a loss of self-confidence and physical fitness, joint limits, muscular weakness, limitations in everyday tasks, lack of balance while walking, social isolation, increasing dependency on others, and fear of falling [5].

Researchers have indicated that balance disorder and leg weakness are the major causes of falling [6]. Researches have also revealed that one-half of the older adults who fall are unable to get back up [7]. When an elderly person remains on the ground or "long-lie" for longer than an hour, it may lead to dehydration, rhabdomyolysis, pressure sores, and pneumonia [8]. Each year, about 3% of them are found helpless or dead at home by paramedics [9].

This raises a need for a "distress call" system to reduce the implications of the "long-lie". An existing Personal Emergency Response System in the form of a push-button pendant is not an effective fall-detection method for "long-lie". Unconscious wearer is unable to activate a pendant with push buttons [9]. Research revealed that disorientation among elderly people could also hinder them from activating their pendant [10]. Additionally, other similar systems are limited to the distance range of the house due to the dependence on a central base station [11]. A phone system is also required to contact the call centre for help, instead of sending alert messages to family members of the user. The intermediary call service also charges monthly fees [12].

A detection system is proposed to enable the elderly to alert their kith and kin directly when they fall down. A wearable detector is linked with a smart phone application, which cuts the cost for call services. It needs to be portable so that smart phone users can carry it around and work beyond a fixed area. The smart phone will update fall events to the smart phone of the caregiver. The data from the detector provides health-care information. Unreported falls can therefore be identified via recorded data.

The proposed system embraces the concept of the Internet of Things (IoT). IoT is the interconnection of uniquely identifiable embedded computing devices within the existing Internet infrastructure [13]. The "thing" is the wearable detector worn by an elderly person. It receives inputs from the wearer and transfers those data to the Internet for collection and processing. The "thing" produces outputs to the physical world with an "actuator". In this case, it is a smart phone application. This IoT concept shall improve health care system.

A shared database that has more than one user will enable the patterns of the fall to be analysed to build statistical modelling of fall events for local or global population. This data analytic and big data approach will enhance the predictive ability of fall events to be used by the health care providers.

2. Methodology

The proposed system is consisting of three vital components, namely, the wearable device, the feature extraction and fall detection subsystems, and the notification subsystems.

The wearable device, which is mounted to the wearer's belt, is responsible to collect movement data for fall and non-fall recognition processing. The movement data is obtained using an accelerometer and a gyroscope of which a variable g-force could be pre-set by the user to indicate an inertial impact value when a fall event occurs. Data that exceed the threshold value will trigger a Bluetooth module to transmit a set of data to a smart phone for processing.

The signals obtained are filtered to remove noise. Then, the filtered signals undergo feature extraction to identify important information. Such information may include the co-acceleration amplitude of the sample that describes the changes in human movements. From the extracted features, classification of the signals is conducted. Hence, fall and non-fall events can be detected.

The notification subsystem consists of the wearer's Android application, a third-party application server, database, Google Cloud Messaging (GCM) connection server, and the caregiver's Android application. The Android applications serve as the notification medium, with the support of the third-party application server, and the GCM connection server. The caregiver will receive notification when the wearer's application submits a fall message to the application server. At present, these notification subsystems were developed using the open source.

2.1 Development of Wearable Device

The block diagram of the fall detection device is shown in Fig. 1, and the developed system is shown in Fig. 2. The size of this device is approximately 8.3 cm long, 5.1 cm wide, and 3.8 cm tall, which is the equivalent of the size of an adult's palm. It comprises of three boards, which are the Freescale FRDM-K20D50M board, Freescale FRDM-FXS-9AXIS sensor board, and a custom circuit on a strip board.

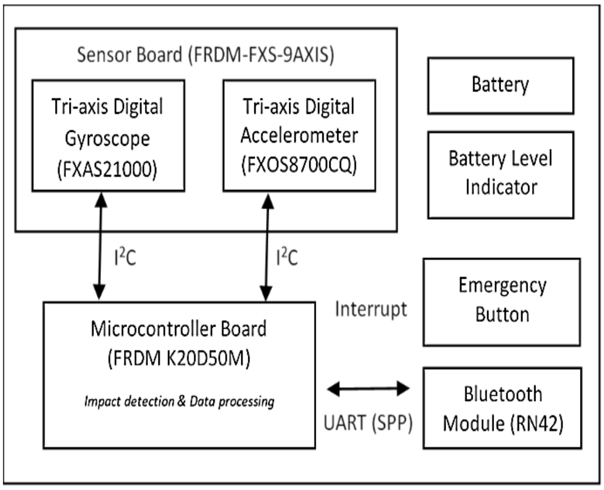


Fig. 1 - Block diagram of the wearable fall detection device

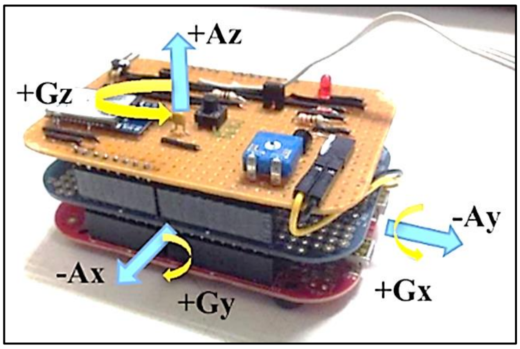


Fig. 2 - Orientation of the accelerometer and gyroscope of the wearable fall detector

The Freescale FRDM-K20D50M board is an ultra-low-cost development platform with a built-in ARM® Cortex^{TM}-M4 processor. The maximum operating frequency is 50 MHz, with 128 KB memory, and 16 KB

RAM. In this project, the board serves as a controller that detects movement data and analyses the data based on the preset threshold value to determine the impact of the movement or fall. Data is collected from the Freescale FRDM-FXS-9AXIS sensor board. This sensor expansion board features a 14-bit tri-axis digital accelerometer (FXOS8700CQ) and a 14-bit digital gyroscope (FXAS21000). The processors are interfaced with the embedded sensors through Inter-Integrated Circuit (I^2C) communication protocol.

The last board is a custom circuit on a strip board with components that include roving networks, RN-42 Bluetooth module, capacitive touch sensor, emergency button, and battery level indicator. The RN-42 Bluetooth module is a small form factor, low power, Class 2 Bluetooth radio. It was added to the system to add a wireless capability for the fall detection device. This feature may add flexibility in the market for this product where it can be utilised by any smart phone system. This module is also equipped with a printed circuit board (PCB), a trace antenna, and supports the Bluetooth Enhanced Data Rate (EDR). The capacitive touch sensor is able to detect the body of the wearer. It is designed to produce an alert signal when the belt is not worn. An emergency button is also implemented to allow the wearer to issue an alert to caregivers with request for help.

The battery level indicator circuit is an indicator circuit with LED to alert the wearer when the battery is low. It consists of a TL 431A voltage reference, a 3.3V Zener diode, a red LED, and a 10 K potentiometer. The red LED is made visible to the wearer to indicate low battery status, and is turned on when battery voltage drops below a value set in the voltage reference from the TL 431A.

2.1.1 Impact Detection Algorithm

The fall detection system in this wearable device is shown in Fig. 3, and the code was developed using the CodeWarrior Integrated Development Environment (IDE) suite. The wearable fall detector device will be initialised during start up or reset of the hardware. Bluetooth buffer size, sensor data buffer size, accelerometer, and gyroscope sensitivity threshold values will be set to default values during the initialisation. Then, a hardware timer event is set to a frequency of 200 Hz. It is set to precisely read sensor data at every 5 milliseconds. The two types of sensors used in this study are a tri-axis digital accelerometer and a digital gyroscope. Table 1 shows the full-scale range and sensitivity of the sensors used in the fall detection.

The acquired sensor data are inserted into a buffer, and the content is constantly maintained to contain 2 seconds of sensor data before impact is detected. The buffer is able to keep 4 seconds data collected from the sensors that sample at every 5 milliseconds.

In a nonlinear fall, a human fall would involve a rotation of the body. It is a fall with significant movements, and external forces are being applied. Meanwhile, a linear fall is a fall without external forces, which is a free-fall condition that involves only one axis of acceleration changes. Sum vector magnitude (SVM) method is used to indicate the impact value of the movement. This value is calculated using the acceleration signals from the accelerometer, as shown in Equation (1).

$$SVM = \sqrt{A_x^2 + A_y^2 + A_z^2} \quad (1)$$

where A_x, A_y, and A_z are the x-axis, y-axis, and z-axis of the tri-axis accelerometer, as shown in Fig. 3.

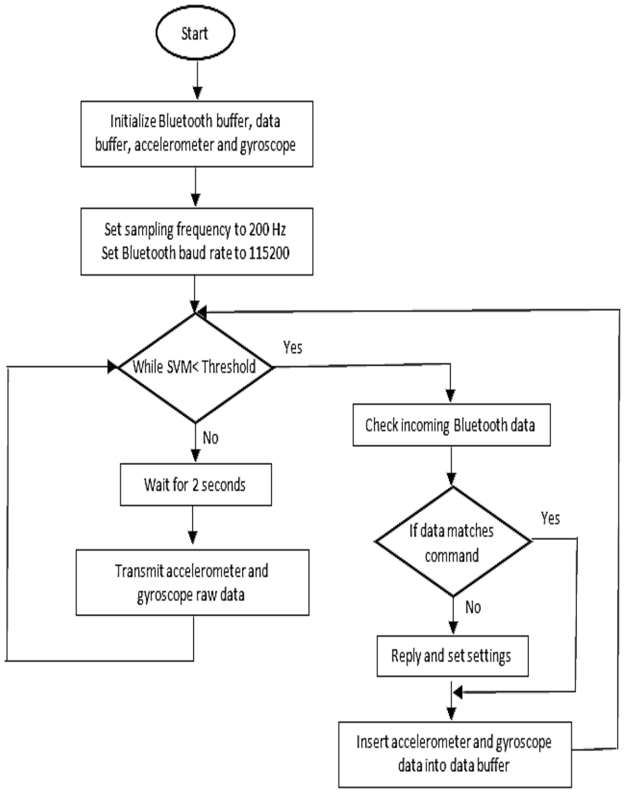


Fig. 3 -Impact detection flowchart

Table 1 - Setting of accelerometer and gyroscope

Setting	Accelerometer	Gyroscope
Data	14-bit signed number	14-bit signed number
Full Scale Range	±8g	±1600 dps
Sensitivity (per count)	1g/1020(0.98 mg)	0.2 dps

During the initialisation stage, the fall detector will be paired with a smart phone via a Bluetooth signal. A threshold value that determines the impact sensitivity is configurable from the smart phone application. To be able to set the threshold value, the fall detector will always listen for the incoming Bluetooth data from the smart phone application. Then, the system will constantly check for the SVM value obtained from the sensor and compares it with the threshold voltage value. If the SVM value exceeds the threshold value, the fall detector will wait 2 seconds to obtain another 2 seconds of sensor data after the impact detection. Each axis of the sensor data will consist of a window size of 4 seconds data. After 2 seconds of the impact, the fall detector will transmit the sensor data to the smart phone.

2.1.2 Smart phone and Wearable Detector Communication

A smart phone application was developed to configure the threshold value of the fall detector and to receive sensor data. This application was developed using the Android Studio, an IDE for developing an Android platform. The configurable threshold value is ranged from 0 to 9 g, where g is the acceleration measurement for gravity, which is equal to 9.81 m/s^2. When an impact is detected, the sensor data from the wearable fall detector is sent to the Android application for fall recognition processing.

In order to classify an actual fall from activities of daily life (ADL) of the elderly, an intentional fall and a sequential ADL were carried out in a laboratory environment. Intentional forward falls were performed by 15 healthy volunteers: ten male and five female subjects, aged between 20 to 25 years old, height ranged between 165 to 190 cm, and weighted between 50 to 95 kg. These subjects were required to perform a forward fall action on a soft mattress with a thickness of 20 cm. Each subject performed three sets of testing activities. Seven types of testing activities were required to be performed (forward fall and six types of activities of daily life), as shown in the following Table 2.

Table 2 - The test activities and their respective descriptions

Activities	Descriptions
Falling	Stand for 2 seconds then fall forward
Running	Run at normal speed of 2.53 steps per second for 8 seconds
Jumping	Jump for 8 seconds
Walking	Walk at normal speed of 1.67 steps per second for 8 seconds

During this experiment, the subjects were asked to follow short instructions for performing intentional falls and ADLs. They were asked neither to fall directly on their hand nor to make any recovery steps. A total of 270 ADLs and 45 falls were recorded.

2.2 Feature Extraction and Fall Detection Subsystems

Data were collected from experiments and were processed in a virtual environment in a computer using the MATLAB software. The algorithm that discriminates the fall from ADLs consists of four important process stages: a) Signal filtering; b) Segmentation; c) Features extraction; and d) Threshold-based fall classification.

For every second (200 samples of data), a high-pass and a low-pass filter were applied to the accelerometer and gyroscope data for obtaining high-pass and low-pass filtered values. The high-pass filtered values correspond to acceleration values, which are due to the movements of the user; whereas the low-pass filtered values correspond to acceleration due to gravity [14]. The data received were low-pass (cut-off frequency = 25Hz) and high-pass filtered (cut-off frequency = 3Hz), with a digital second-order Butterworth filter for posture detection and dynamic analysis.

According to [15], activity classification errors are often the result of inappropriate segmentation due to poor window size selection. Hence, a suitable length for the window or frame of filtered signal must be selected during this step. Short time lengths will avoid delays from continuously affecting real-time processing, while providing a reasonable recognition rate [16]. Hence, the window size selected in this study was 2.5 seconds. The technique used was a non-overlapping window sizing, which depends on the peak of the signal. Timeframe of 1 second before and 1.5 seconds after the peak of the signal was selected.

The features were extracted from both the low-pass filtered signal and the high-pass filtered signal. Multiple potential features were tested and approximately only six optimised features were selected for achieving high accuracy results.

There were five features extracted from the accelerometer signal (SVM_a, SV_{yz}, θ_{ratio}, $NormAcc$, and $MaxMin$) and one feature from the gyroscope signal (SVM_w), as listed in Table 3. For acceleration, three features (SVM_a, SV_{yz}, and $NormAcc$) were selected from the high-pass filtered acceleration signal and two features ($MaxMin$ and θ_{ratio}) were extracted from the low-pass filtered acceleration signal. For angular velocity, the (SVM_w) feature was extracted from the high-pass filtered angular velocity signal.

Table 3 - Lists of features

Activities	Descriptions
SVM_a	Sum vector magnitude of acceleration
SV_{yz}	Acceleration on horizontal y-z plane
SVM_w	Sum vector magnitude of angular acceleration
$NormAcc$	L1 norm of acceleration, which is the total acceleration exerted on the device
$MaxMin$	Differences between maximum and minimum acceleration values
θ_{ratio}	Ratio of polar angle

The formula for feature extraction as shown below:

- Sum vector magnitude of acceleration

This value can be calculated from the high-pass filtered acceleration values, also known as the dynamic sum vector. This feature describes the spatial variation of acceleration during the falling interval [17]. The feature is obtained from Equation (1).

- Acceleration on the horizontal y-z plane

This parameter represents the body tilts during the fall [17]. Its value is larger than a certain threshold if there is a fall. It is calculated from the high-pass filtered acceleration values using the following Equation (2):

$$SV_{yz} = \sqrt{a_y^2 + a_z^2} \quad (2)$$

where a_y and a_z are the acceleration of y-axis and z-axis, respectively.

- Sum vector magnitude of angular velocity

This value is calculated from the high-pass filtered angular velocity values. It is used to detect fall-related impacts. The formula from [18] is as shown in Equation (3):

$$SVM_w = \sqrt{w_x^2 + w_y^2 + w_z^2} \quad (3)$$

where w_x, $w_y$and w_z are the angular velocity of x-axis, y-axis, and z-axis, respectively.

- L1 norm of acceleration

This feature is calculated for each second from the high-pass filtered acceleration values. It directly corresponds to the amount of acceleration that a user has exerted on the accelerometer. If this value is above a certain threshold, it indicates that the user is involved in a dynamic activity, such as running, jumping, or experiencing a fall [14]. Equation (4) is used to determine this value:

$$NormAcc = |a_x| + |a_y| + |a_z| \quad (4)$$

where $|a_x|$, $|a_y|$and $|a_z|$are the absolute acceleration of x-axis, y-axis, and z-axis, respectively.

- Differences between maximum and minimum acceleration values

These values can be obtained during the interval time of the last 0.5 second of the signal. It is used to test the orientation of the user by determining whether the user is in a static phase after the fall.

$$MaxMin = a_{max} - a_{min} \quad (5)$$

The a_{max} and a_{min} refer to the maximum and minimum values from the sum vector magnitude of the acceleration signal, which is filtered by the low-pass filter.

- Ratio of Polar angle

It describes the posture of the body when the body is in a static position. This angle reflects the body-tilt and a sudden change could be indicative of a fall [18].

$$\theta_{ratio} = \cos^{-1}\left(a_x / \sqrt{a_x^2 + a_y^2 + a_z^2}\right) \quad (6)$$

where a_x, a_y and a_z are the accelerometer values.

The threshold-based method was proposed because it has low power consumption and fast processing speed, which would be suitable for the wearable fall detector system. It also reduces the computational complexity of training and testing the classifiers. Threshold-based technique distinguishes the actions between the fall and ADLs when the peak values are above or below certain predefined threshold values.

A two-stage fall detection algorithm that can detect the fall of an elderly person was developed in this project. This algorithm consisted of two processing stages, in which the first stage was used to detect the impact of the fall and the second stage was used to determine the orientation of the user when the fall occurs. The flow of the algorithm is shown in Fig. 4.

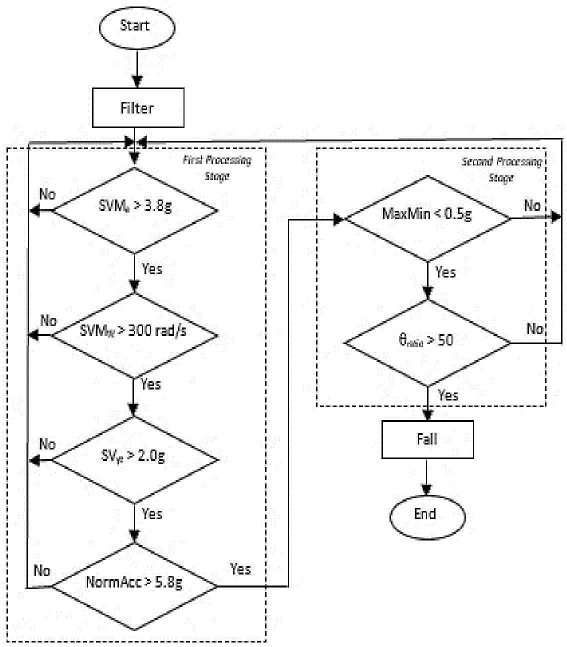


Fig. 4 -Flowchart of the fall detection algorithm

The implementation of the fall algorithm was tested in the android environment. Since the fall algorithm was developed in MATLAB, the MATLAB code was first converted into C++ language using MATLAB Coder. It was then integrated into the phone using the Qt software, which is a cross-platform application and user interface (UI) development framework for all major desktops, embedded and mobile operating systems. The MATLAB code was implemented as a C++ static library through the Coder. A test for compiling and running the converted code in Qt was conducted.

2.3 Development of Notification Subsystem

The Android operating system was selected as the development platform. This is because Android is a free and open platform. Furthermore, Android has Google Cloud Messaging (GCM) service that allows you to send data from a third-party application server to Android-

Published by MBOT Publishing
https://jetia.mbot.org.my/index.php/jetia/index

powered devices. The GCM allows the delivery of messages to the targeted Android Application running on the target device [19].

Google Cloud Messaging (GCM) enables an Android application to receive downstream messages (from servers to GCM-enabled client apps). It also allows the GCM-enabled client apps to send upstream messages to a third-party application server. These messages can be a lightweight message that informs the client's phone application that there are new data to be fetched from the server or it could be a message containing up to 4 kb of payload data [19].

The design of this Android application was developed using the Android Studio. The selected platform was Android and above so as to support the GCM. A Google Account was created to obtain its project number based on a created-project ID. The project number was later used as the GCM sender ID. Inside the project account, the "Google Cloud Messaging for Android" option was enabled. Server API key was obtained for sending messages from the third-party application servers to GCM connection servers.

The Android application was also developed for receiving messages. The application consisted of various components, which include "MainActivity", "RegisterGcmService", "GcmBroadcastReceiver", and "GcmIntentService". The "MainActivity" checks the Google Play Services' APK and its version when the Android application is run. This is vital to ensure that the smart phone can run the Android application using GCM services. This step was followed by checking the availability of the registration ID for the device. If the Registration ID is not found, the device will need to be registered. This was done by obtaining its registration ID from the GCM cloud servers. The received registration ID was stored in the phone and was also submitted to the third-party application server via the "RegisterGcmService". The "GcmIntentService" will receive messages from the third-party servers and displays these messages as notification on the targeted caregiver's phone. The "GcmBroadcastReceiver" manages the partial wake lock on the application. It is vital to make sure that this device does not go back to sleep before the application activity is completed.

2.3.1 **Server-side System**

Server-side network programming was used to design and implement programs to be run on a server. The server-side applications run on a cloud infrastructure as a process and perform works requested by clients [20]. In this project, the server-side application has three main functions. Its first function was to receive the registration ID from the smart phone and to store it in the database. The second function was to send a message to the Android terminal through the GCM connection servers, and the final function was to update the database once a fall event has been determined. The server-side scripting language used in this project was the Hypertext Pre-Processor (PHP). PHP is a thoroughly functional language that can generate a variety of data types and make accessible to the database [20].

The server-side application in this project involved only two layers, namely, the business logic layer and the resource layer. The application developed in the server deals with user-related commands, processes the data, and organises the data storage and retrieval using a database management system. Hence, three PHP scripts were developed for the server-side system. These scripts are currently held in the 3rd-party application server. Currently, this project uses a free web hosting service provided by the www.000webhost.com. It is expected to have web hosting services provided by the Telekom Malaysia Berhad. This collaboration would allow the fall detection system to be enabled by the Telekom Malaysia Berhad.

The "AndroidRegister.php" was used to obtain the registration ID from the Android device and store the registration ID in the database. When a message needs to be sent to an Android device, the "MessageSend.php" receives the message from the wearer's Android application via HTTP protocol and fetches the registration ID from the database. These two data will be sent to the GCM connection servers using the "curl" method. The GCM connection server will respond by forwarding the messages to the Android device corresponding to the registration ID received.

Another "UpdataDatabase.php" was used to update the fall event as determined by the wearer's Android application in the online database. The script will receive the JavaScript Object Notation (JSON) encoding message from the Android application and decodes it accordingly. The decoded messages will put the data into their respective columns in the database. The JSON language is advantageous for data exchange, storage, and communication by any programming languages. JSON-formatted web service makes accessing the data in the database easy.

2.3.2 **Database**

Database is meant for persistent data storage [21]. The database is managed by an intermediary program, known as a database management system. This current project used the "MySQL" system. The database arranges the data in a relational approach. The table in the database has different columns that constitute different fields. Each row is one collection of fields and known as a record.

Generally, three methods are used for managing databases, namely, PhpMyAdmin, PHP script, and Structured Query Language (SQL) command line. This project employed the PhpMyAdmin and PHP script. The PhpMyAdmin was used to develop the database and to create the tables and their fields. Using PhpMyAdmin to create tables eliminates typographical errors when creating the attributes of the fields [22].

The database serves two main purposes in this development. It was used to store the registration ID received from the Android device and to store the fall

Published by MBOT Publishing
https://jetia.mbot.org.my/index.php/jetia/index

history of the wearer. Access to the database can be achieved by providing the user's name and the password to enter the web host.

Two tables were created in the database. One was for the GCM registration ID and the other was for the fall history. The creation of tables was done using PhpMyAdmin. The information in these tables were updated using PHP scripts that send SQL queries to the MySQL database management system. These tables were then updated accordingly. Databases require a server, a database program, and a PHP processor [22]. At the initial stage of the database development and testing, a single all-in-one package, known as WAMP (Windows, Apache, MySQL, and PHP) was employed in the local personal computer. The developed and tested database in the personal computer was then deployed to the remote host using the free hosting service provided by the www.000webhost.com. Similarly, the database is expected to be hosted by the Telekom Malaysia Berhad in the long-run as they will be the internet service provider for this fall detection system.

2.3.3 Implementation of Notification System

The system was then implemented using two Android smart phones. Each of them has the fall detection system's Android applications installed. One smart phone was for the wearer and the other was for the caregiver. The wearer's application was expected to transmit the fall event message to the notification application of the caregiver and at the same time, updates the event of the fall of the wearer in the database. The third-party application server was responsible for receiving the fall-related messages and registration ID to update the database and to send these messages to the GCM server, including the server ADI key (which was previously obtained from the Google account). The GCM server will then submit the messages to the caregiver's Android phone.

Hence, the fall event will be recorded in the database and messages related to the fall will be notified to the caregiver. This sequence produces the event-triggered fall notification system. Fig. 5 depicts the general block diagram of the notification subsystem.

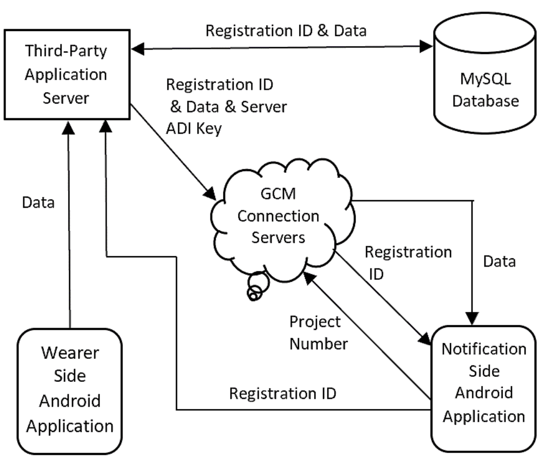


Fig. 5 - The notification subsystem architecture

3. Results and Discussions

A custom circuit was built on a strip board, as shown in Fig. 2. An emergency button was placed at the centre of this device and a red LED was placed at the top-right corner. The intention for creating custom circuits on strip boards was to allow these devices to be firmly attached, which would provide an easier path for interfacing components. A capacitive touch sensor was attached with a long wire onto a belt. A complete fall detector with orientation of sensors is labelled. There are three axes for accelerometer (A_x, A_y, A_z) and gyroscope (G_x, G_y, G_z) The accelerometer measures the rate of change of velocity.

3.1. Threshold Determination

Impact detection can be improved by providing a reasonable threshold for SVM. Determining a threshold value was a challenging part in this project. Low threshold values could increase false positives, where normal activities could be judged as an impact of a fall. Meanwhile, high threshold values could also cause false negatives that lead to failure of impact detection. In this project, threshold values were determined empirically, which include determining thresholds by taking minimum values of SVM during the fall experiments. In order to determine threshold values empirically, 15 subjects were instructed to wear the fall detector and perform a total of six activities: (a) fall, (b) jump, (c) run, (d) lie, (e) sit, (f) stand, and (g) walk. A box plot of the conducted experiments that represented different activities is shown in Fig. 6. The box plot shows the maximum, minimum, first quartile, second quartile, and mean values of each performed activity. The box plot shows that to effectively detect the impact of a fall; a minimum threshold value of 4.73 g can be used. This was the lowest peak value of SVM in the fall events. Normal activities, including jumping, running, and sittings rarely exceed this value. Although the threshold value was slightly higher than the maximum point of jumping, the threshold value was still valid to be used to ensure that every incidence of falls is being detected.

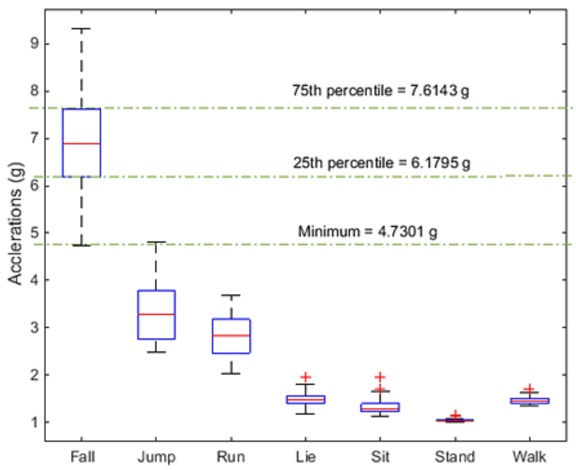


Fig. 6 - Box plot of SVM peak values using acceleration signals

Published by MBOT Publishing
https://jetia.mbot.org.my/index.php/jetia/index

3.2 **Power Consumption of Wearable Fall Detector**

Energy consumption is an important figure of merit for any mobile application, especially in this fall detection system. Tables 4 and 5 show the experimental results when the detector was under an idle state and when it was experiencing a fall. An idle condition is when no Bluetooth connection is established between the fall detector and the smart phone. The fall detector will constantly check for impacts of human movements. When an impact is detected, Bluetooth connection is established, and the fall detector will start to transmit sensor data to the smart phone. Several readings were recorded and the minimum, average, and maximum values of these readings are tabulated in the following tables.

From these tables, wireless communication (Bluetooth) is shown to consume the most power in the system. When an impact is detected, the Bluetooth is used to send sensor data and the current withdrawn power can go up to 24 mA. In an idle condition, assuming that battery capacity of 3000mAH is being used, the expected lifetime of the fall detector would be 3000 mAH/28 mA = 107.14 hours (equivalent to 4.46 days).

Table 4 – Power consumption of an idle fall detector

Symbol	Current Consumption		
	Avg	**Min**	**Max**
Overall	29mA	28mA	50mA
Microcontroller board	576.6μA	576.6μA	577.4μA
Bluetooth	2mA	1mA	23mA

Table 5 – Power consumption of fall detector when an impact is detected

Symbol	Current Consumption		
	Avg	**Min**	**Max**
Overall	50mA	49mA	510mA
Microcontroller board	577.7μA	577.5μA	577.8μA
Bluetooth	23mA	22mA	24mA

3.3 **Mobile Application**

An Android application was designed to connect to the wearable fall detector, to configure the threshold values of SVM, and to receive sensor data from the fall detector, as shown in Fig. 7. The user is allowed to change the default threshold value of 4.73g to another value.

The database developed in this system can receive details of a fall event from the wearer's Android application. The database will record the wearer's name, date added to the database, and the time and date of the fall event.

The smart phone of the caregiver has been pre-installed with the fall detection system's Android application. This app will display the notification when triggered by the wearer's Android application. The notification is as shown in Fig. 8. The details include the wearer's name, the date, and the time of the fall

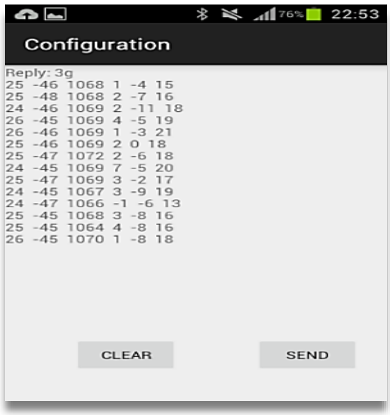


Fig. 7 - Android application for receiving sensor data (accelerometer and gyroscope) from fall detector

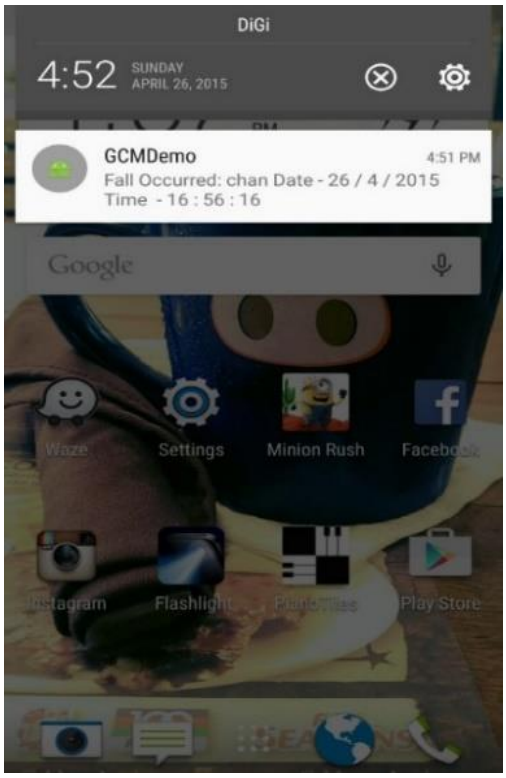
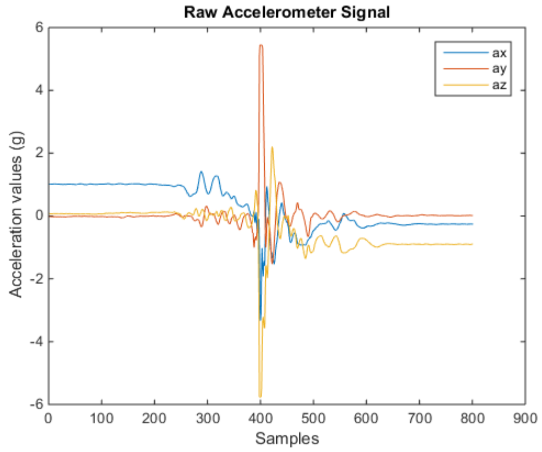


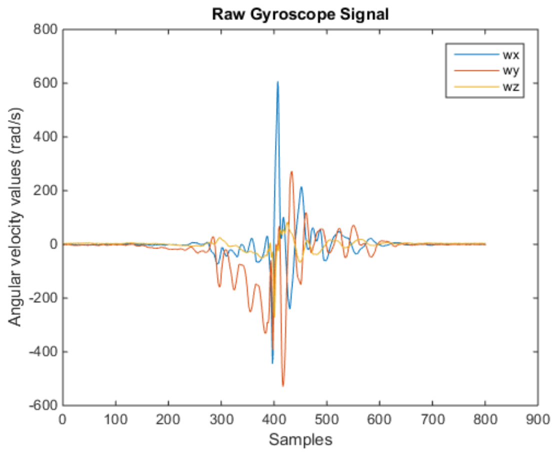
Fig. 8 - The notification received by the caregiver's Android phone

3.4 **Experimentation of Fall Detection Algorithm**

This algorithm was tested with 15 test subjects wearing the device at the waist (clipped onto the belt). They were instructed to perform falling and six kinds of ADLs, and the results are shown in Fig. 9. These results are shown as graphs of raw accelerometer signals and gyroscope signals when a fall has occurred. Since there is a fall, each of the axes has a large variation during the time interval of fall from both sensors.



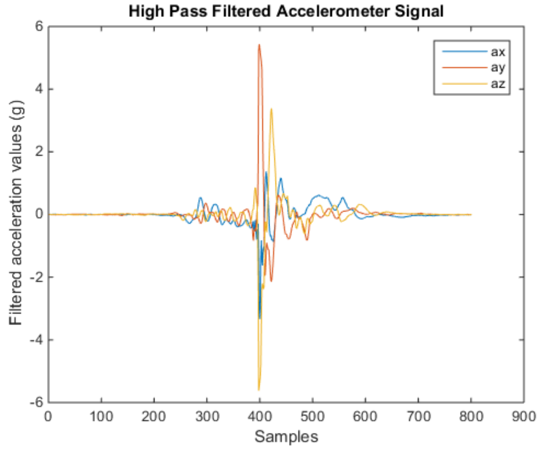
(a)



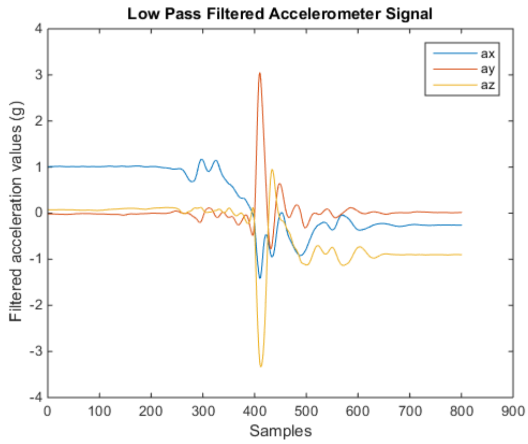
(b)

Fig. 9 - The raw sensor signals when a fall occurs (a) accelerometer and (b) gyroscope signal

Fig. 10 and Fig. 11 shows the high-pass filtered values as well as the low-pass filtered values for both sensors. The high-pass filtering was used for filtering out the low-frequency acceleration (gravity) that captured information related to static activities (standing, lying down, and sitting). Low-pass filter was used to filter out high–frequency components that exist in the signal in order to investigate the orientation of body performance.



(a)



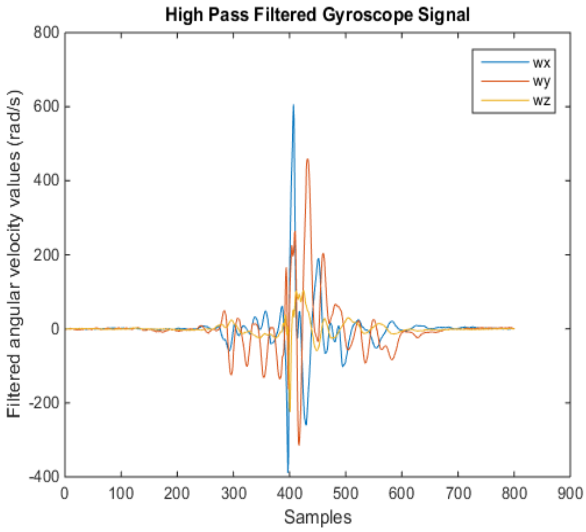
(b)

Fig. 10 - The filtered accelerometer signal when a fall occurs

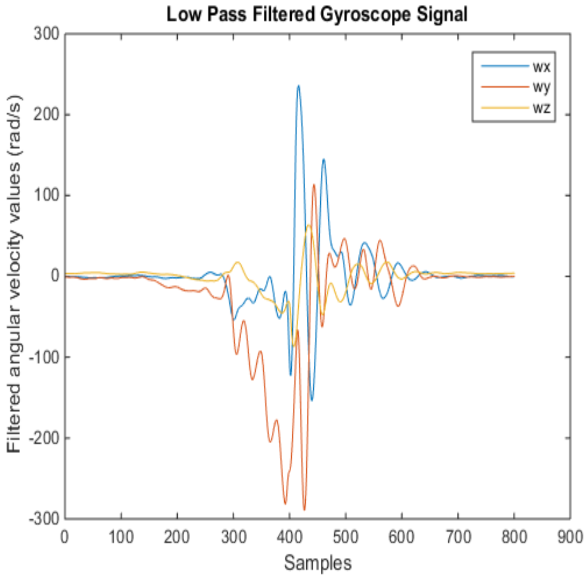
The filtered signal was then segmented into timeframes of 2.5 seconds, which were 1 second before and 1.5 seconds after the peak of the signal. Such short length frames were chosen due to the use of continuous signals. A long length timeframe will create difficulty when interpreting and analysing results.

The MATLAB algorithm for sensing a fall was explained in the flowchart shown in Fig. 4. The algorithm consisted of two processing stages for detecting a fall. The first processing stage was to detect the great impact of the fall and this stage uses four features that calculate the sensor values to determine whether there is a large impact during the processing stage. These four features include SVM_a, SVM_w, SV_{yz} and $NormAcc$. The first two features (SVM_a and SVM_w) were described previously and their threshold values were set to 3.8g and 300rad/s, respectively.

Published by MBOT Publishing
https://jetia.mbot.org.my/index.php/jetia/index



(a)



(b)

Fig. 11 - The filtered gyroscope signal when a fall occurs

However, these features (SVM_a and SVM_w) were not enough to detect an impact. Thus, two additional features (SV_{yz} and $NormAcc$) were introduced to increase the performance of the fall detection algorithm. SV_{yz} can be described as the acceleration on the horizontal y-z plane; whenever there is a fall, the value of acceleration over the horizontal plane would be higher than 2.0g, which became the threshold value. The $NormAcc$ represents the total acceleration exerted on the device (or user) when there is a fall. The value of the acceleration was higher than the threshold value of 5.8g.

For a user who is experiencing a fall-like event, such as highly dynamic activities (jumping and running), the features in the first phase were insufficient for classifying a fall since jumping and running have similar high impact values. Hence, the second processing stage was introduced to determine the orientation of the body; whether the wearer is in static phase after a fall.

The features used in this phase were the $MaxMin$ and θ_{ratio}. The $MaxMin$ is the differences between the maximum and minimum values of acceleration. This difference was measured during the final 0.5 second of time interval. If a wearer is falling, he or she will remain on the ground (static) after the fall. Hence, the value of acceleration during this static phase would be low, which was lower than 0.5 g. This feature has a threshold value of 0.5 g. The θ_{ratio} is the ratio of the polar angle. It indicates the orientation of the body. From the results, this feature exceeded 50˚ when a fall has occurred

Fig. 12 shows boxplots of all six features for the seven types of activities. Fig. 12(a) shows the boxplot for the sum vector magnitude of acceleration. It was found that the values of the fall were higher than for the other six types of ADLs, which was higher than 3.8g. Thus, the threshold value of this feature was set to 3.8g. As for the sum vector magnitude of angular velocity, the value of fall was higher than the values of other activities, which was higher than 300 rad/s. Hence, this value was selected as the threshold value for this feature

The acceleration on the horizontal y-z plane of a fall was on average greater than for the other activities. The highest value for the ADLs was approximately 1.8g (running). Hence, the threshold was set as 2.0g for this feature. For the next feature, the norm L1 of acceleration, the results showed that the value for falling was higher than 5.8g. Thus, this value was set as the threshold value. The value of differences between the maximum and minimum acceleration for the final 0.5 second for falling and other static activities (lying, sitting, and standing) was lower when compared to the value for dynamic activities (walking, running, and jumping). Low values of acceleration were determined for this feature, which showed that the movement of the user was in a static state. The difference was determined as 0.5g, which was set as the threshold value.

The boxplot of the last feature, which was the ratio of the polar angle, is shown in Fig.12(f). The angles of falling and lying have high values when compared with the values for other activities. This observation showed that falling and lying involve tilting the body. Based on the experimental results, the angle of the body when it is experiencing a fall will reach 50˚ and above. Hence, 50˚ was chosen as the threshold value for this feature.

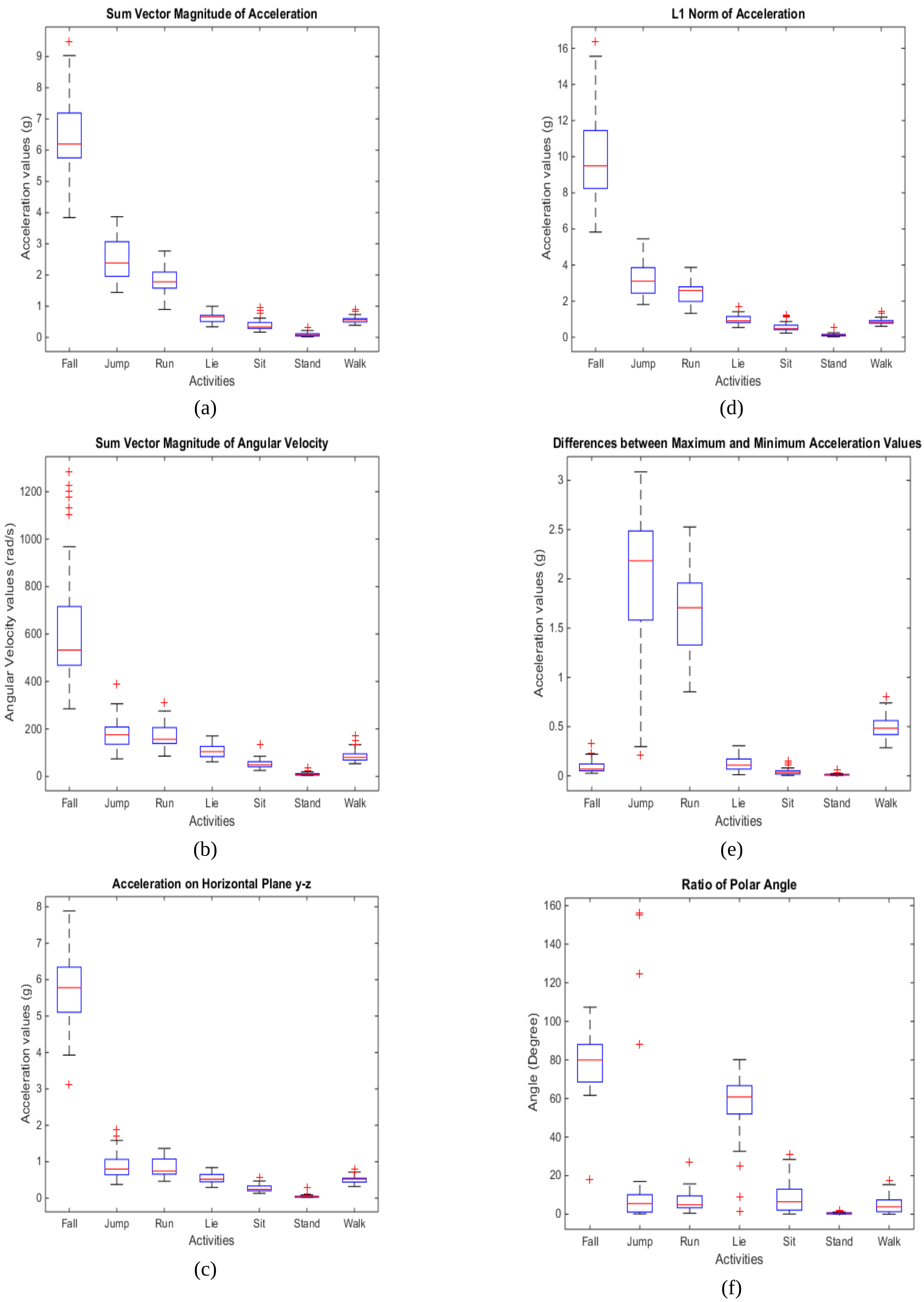


Fig. 12 -. The boxplot of the features

Published by MBOT Publishing
https://jetia.mbot.org.my/index.php/jetia/index

Since the output of the fall detection is binary, there were four possible results, as shown in Table 6. The algorithm performance was tested by collecting the fall and ADL test data, as previously mentioned. Its sensitivity reached 97.78%, specificity reached 98.52%, and accuracy reached 98.41%. Tables 7, 8, and 9 show detailed information considering the sensitivity, specificity, and accuracy of this algorithm.

Table 6 – Possible outcomes of the fall detection

Fall Outcome	Explanation
True Positive (TP)	A fall happens and the algorithm detects it.
False Positive (FP)	An ADL is performed and the algorithm detects a fall.
True Negative (TN)	An ADL is performed and the algorithm detects as non-fall.
False Negative (FN)	A fall occurs but the algorithm does not detect it.

Table 7 – Sensitivity of the Algorithm

Scenario	Performance
Falls (Number of Falls)	Forward Fall (45)
Fall detected (TP)	44
Fall not Detected (FN)	1
Sensitivity (TP)/(TP+FN)	97.78%

Table 8 – Specificity of the Algorithm

ADL (Number of tests)	Fall not detected (TN)	Fall Detected (FP)	Sensitivity (TN)/(TN+FP)
Running (45)	43	2	95.56%
Jumping (45)	43	2	95.56%
Walking (45)	45	0	100.00%
Standing (45)	45	0	100.00%
Sitting (45)	45	0	100.00%
Lying (45)	45	0	100.00%
Total (270)	266	4	98.52%

Table 9 – Accuracy of the Algorithm

Scenario	Performance
Number of tests (Fall and ADLs)	315
Fall Detected (TN+TP)	310
Fall not Detected (FN+FP)	5
Sensitivity (TN+TP)/(TN+TP+FN+FP)	98.41%

4. Conclusions

The Internet of Things (IoT) was implemented in this fall detection system project. In general, the IoT model was required to support three features, namely, the presentation, execution, and tagging. In this project, the wearable fall detector offers a personalised fall detection record (presentation). Next, the event of a fall will trigger the caregiver's mobile application, in which proper medical intervention can take place (execution). Lastly, the fall records collected from the user can be further analysed, fused, and interpreted with other health care data (tagging). This IoT model will produce a data-driven healthcare for the elderly.

The determined fall event was then sent to the Android mobile application. The mobile application, with the GCM messaging model, allows the emergency medical service of the fall detection system to be notified. The GCM utilises the push notification model, in which the data flow was initiated by the data sources, and was aperiodic as there is no predestined schedule for sending data. This protocol allows the individual uni-cast messages to be sent to multiple clients. This model also permits multiple caregivers to monitor the fall event of an elderly person.

Push notification forms the basis of most event-driven systems. It is a good solution for real-time systems in which it becomes particularly efficient when the system needs to notify a larger number of caregivers. Thus, for the occasional, time-critical message that must be received and given immediate attention by the receiving end, this messaging model becomes highly effective. The application does not need to poll the request to check whether there are any messages to be sent to the caregiver. This system works like a background task, which certainly reduces the power consumption of the caregiver's smart phone. Hence, this model of health monitoring application would be more acceptable by the caregiver.

In addition, when compared to notifications using the Global System for Mobile Communication (GSM), the system that uses GCM would be more cost-effective. This fall detection system is using the cloud services provided by Google to send messages to the intended subscribers. Hence, the service for GCM is free. Apart from that, the fall events of a particular wearer are recorded in the database. This health care database is invaluable to the user as well as to health care organisations, such as the hospital. This mobile application could enable ubiquitous access of personal health records by the elderly and the caregivers.

Nonetheless, this newly developed database needs to have a certain level of security. MySQL provides account security in which accessing the data requires user name and password. This will allow only authorised personnel access to the data. This database is also vital for fall risk assessments, prevention and detection.

Acknowledgement

The authors wish to convey their gratitude to the Department of Electrical and Electronic Engineering, UPM for funding this project.

References

[1] David, A. Malaysia's ageing population: Planning for the future, The New Straits Times, May 15, 2022.

[2] World Health Organization (2007). Global report on falls prevention in older age. WHO Technical Meeting.

[3] Ozcan, A., Donat, H., Gelecek, N., Ozdirenc, M. & Karadibak, D. (2005). The relationship between risk factors for falling and the quality of life in older adults. BMC Public Health, 5(90), 1-6.

[4] Kannus, P., Sievanen, H., Palvanen, M., Jarvinen, T. & Parkkari, J. (2005). Prevention of falls and consequent injuries in elderly people. The Lancet, 366(9500),1885–93.

[5] Giovannini, S., Brau, F., Galluzzo, V., Santagada, D.A., Loreti, C., Biscotti, L., Laudisio, A., Zuccalà, G. & Bernabei, R. (2022). Falls among older adults: screening, identification, rehabilitation, and management. Applied Sciences, 12(7934), 1-17.

[6] Rubenstein, L.Z. (2006). Falls in older people: epidemiology, risk factors and strategies for prevention. Age and Ageing, 35-S2, 37-41.

[7] Cummings, S.R., Nevitt, M.C. & & Kidd, S. (1988). Forgetting Falls. The limited accuracy of recall of falls in the elderly. Journal of the American Geriatrics Society, July 7, 613-616.

[8] Fleming, J. & Brayne, C. (2008). Inability to get up after falling, subsequent time on floor, and summoning help: prospective cohort study in people over 90. BMJ, 337, A2227.

[9] Gurley, R.J., Lum, N., Sande, M., Lo, B. & M. H. Katz, M.H. (1996). Persons found in their homes helpless or dead. The New England Journal of Medicine, 334(26), 1710–1716.

[10] Porter,E.J. (2005).Wearing and using personal emergency respone system buttons. Journal Gerontological Nursing, 31(10), 26–33.

[11] Chen, J., Kwong, K., Chang, J. Luk, J. & Bajcsy, R. (2005). Wearable sensors for reliable fall detection. Proceedings of the 2005 IEEE Engineering in Medicine and Biology 27th Annual Conference Shanghai, China.

[12] Harari, Y., Shawen, N., Mummidisetty, C.K., Albert, M.V., Konrad, K.K.P. & Jayaraman, A. (2021). A smartphone-based online system for fall detection with alert notifications and contextual information of real-life falls. Journal of NeuroEngineering and Rehabilitation, 18(124), 1-13.

[13] McEwen, A. & Cassimally. H. (2014). Designing the Internet of Things. United Kingdom. John Wiley and Sons, Ltd.

[14] Althobaiti, T. Katsigiannis, S. & Ramzan, N. (2020). Triaxial accelerometer-based falls and activities of daily life detection using machine learning. Sensors, 20(13), 3777.

[15] Jun, S.S., Ramli, H.R.R., Soh, A.C., Kamsani, N.A., Raja Ahmad, R.M.K., Ahmad, S. A. & Ishak, A.J. (2020). Development of fall detection and activity recognition using threshold-based method and neural network. Indonesian Journal of Electrical Engineering and Computer Science, 17(3), 1338-1347.

[16] Pham, C. & Phuong, T.M. (2013). Real-time fall detection and activity recognition using low-cost wearable sensors. in Computational Science and Its Applications – ICCSA 2013, Berlin Heidelberg: Springer. 673-682.

[17] Subramanium, S., Faisal, A.L. & Deen, M.J. (2022) Wearable sensor systems for fall risk assessment: A review. Frontiers in Digital Health, 4:921506. 1-20.

[18] Rakhecha, S. (2013). Reliable and secure body fall detection algorithm in a wireless mesh network. M.S. Thesis, Dept. Com. Eng., Kate Gleason College.,NY, United States.

[19] Android Open-Source Project. Overview Google Cloud Messaging. [Online]. Available: https://developer.android.com/google/gcm/gcm.html

[20] Ciubotaru, B. & Muntean, G.M. (2013). Advanced network programming. Principles and Techniques, Computer Communications and Networks, London: Springer-Verlag.

[21] Rochkind, M. (2013). The database, in Expert PHP and MySQL: Application Design and Development. Apress, .83-138.

[22] A. W. West, A.W. (2013). Create and test a MySQL database and table, in practical PHP and MySQL website databases: A simplified approach. Apress, 1-24.