ELDERLY FALL DETECTION SYSTEM

SUBMITTED TO,

Dr. Purnendu Pandey
SOET

SUBMITTED BY,

Hitesh Varma P [1600228C203]
N V Dhatrija Chowdary [1600367C203]
Pratyush Chandolu [1600198C203]
Dinesh Ega [1600104C203]
Sai Jawahar Reddy M [1600326C203]

Abstract

The project is focused entirely on the elder people. The falls due to accidents or loss of consciousness in older people can turn out to be fatal if the required assistance is not received with in time. The fear of falling puts constraints on the free movement of the elderly, and forces them to always be accompanied by a caretaker. People who don't have the opportunity of personal caretaker are forced to leave their comfort behind and move to the care facilities. Our idea was to seek help when required rather than being watched over at all times. We achieved this through two ways of communication ranges. One being the SMS to the desired person but when they are far away or not available the second way of alerting the nearby people with the help of a buzzer.

Introduction

Individuals everywhere throughout the world are facing much fall not only because of faint. People experiences different types of fall in public places. It can occur on playgrounds, streets, parks, work environment, schools, offices, etc.

In India, every day more than 30 people were murdered, and many are suffering austere mental and physical trauma. An individual is among the most under-announced wrongdoings overall in view of the social disgrace appended to the idea of the wrongdoing. These Android applications put those assets readily available rapidly, and a few of them have both free and premium forms. Regardless of whether you're stuck in a deplorable circumstance or get isolated from companions amid a night out and don't have the thought how to return home, having these applications on your telephone can diminish your hazard and bring help when you need it.

Falling among the old is viewed as a major cause for death. Lately, encompassing and remote sensor stages have been broadly utilized in created nations for the discovery of falls in the old. Be that as it may, we trust additional endeavors are required to address this issue in creating nations.

Falls are characterized as the incidental settling down of a body on the ground, floor or other lower level. The predominance of falls is extremely normal among the older and increments with age. The World Health Organization (WHO) revealed that 28%– 35% of individuals matured 65 years or more fall every year and the rate increments to 32%– 42% for those more than 70 years old
This gives two major point:

1. Falls need immediate medical attention and there is only one way that can establish is by conceiving a sensing method.
2. Falls greatly limit the independence of an individual. He or she will not have the confidence to venture or stay alone for extended periods of time. This is primarily due to the fear of medical complications surfacing.

Since the root issue has been set up, we will recognize a strategy for detecting.

i. Non-invasive
ii. Wearable system
iii. No false-positives

iv. No delay time
v. Common route of communication
vi. Interactivity

Literature survey

There have been different approaches to solve the problem of fall detection. In those, many of them focus on elderly mainly because of the fact that they are in most need of such helping system. We will discuss many approaches that have been implemented ranging from specialized and more advanced sensor based to use of Machine Learning methods to detect elderly fall. Most of the approaches we researched fail to see a false positive which is a challenge. Even the F1 scores and other evaluation metrics provide only a slight information on accuracy in various and diverse situations. We will discuss approaches based on placement of sensors or device used for detecting falls. There are many ways we can integrate a fall detection system to monitor falls of elderly: wearable devices, camera-based, ambience device, motion device, motion analysis, shape change analysis among many others. Evaluation metrics relevant for fall detection systems are accuracy, sensitivity and specificity.

[1] Uses specialized fall sensor component comprising of accelerometer sensor, battery, microprocessor and Bluetooth module. The algorithm is a very simple one which has around 90-95% sensitivity and 91-93% specificity.

$$\text{Specificity} = (TN/(TN+FP)) \text{ and sensitivity} = (TP/(TP+FN))$$

[2] Uses sophisticated machine learning algorithms like DAGSVM, ellipse fitting techniques on input from many types of sensors like gyroscope, accelerometer, etc. but it is only deployable in smart home environment. Also it requires a lot of power to run different mining algorithms on data every second. Also, it doesn't fare well for outside environment. Apart from fall, it detects other types of postures and activity of a person in that smart home environment.

[3] Uses accelerometer, gyroscope, magnetometer, MEMS and MARG sensor. It does have 89-91% accuracy, 60-90% sensitivity and 98-100% specificity. It performed well on a large scale deployment runs.

[4] Uses video feed to detect fall in home environments. This method too uses ellipse approximation techniques, projection histograms and sophisticated foreground segmentation along with other techniques to detect fall in a video feed.

[5] Uses RGBD cameras to detect fall. The RGBD is a 3D camera which produces a 3D model of the environment and algorithms crunch the video feed every second to detect various types of feeds. It uses special computer vision algorithms on video data.

[6] Discusses method of threshold in a tri-axis accelerometer for fall detection. It shows differences between values for fall zone and non-fall zone.

IoT Architecture

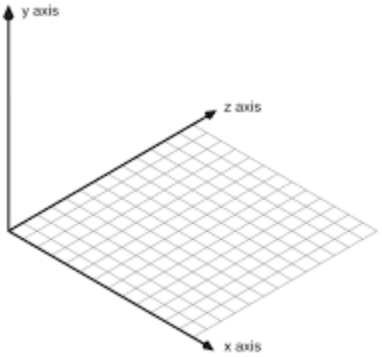
Sensors:

Accelerometer: It is a 3-axis sensor. The data we can get from the accelerometer are the acceleration and the angular position in each axis. The net acceleration can be calculated by the sum of the squares of the acceleration in each axis.

$$A_{net} = \sqrt{A_x{}^2 + A_y{}^2 + A_z{}^2}$$

The x and y axis being parallel to the ground are zero whereas the z which is perpendicular is 1 in a ideal state. In an ideal stationary position the Anet is its maximum i.e., 1g. During a free fall, the feeling of weightlessness is due to the net acceleration dropping to zero. The free fall is definitely different from the any ordinary fall. Hence the net acceleration is not zero but substantially less than 1. Along with the net acceleration we can calculate the angular position. The angle on the x, y and z is Roll, pitch and Yaw respectively. The angular position towards the z axis is not effective in the finding the required result. The roll and pitch are the relevant for our use case or say scenario.

$$\text{Pitch} = tan^{-1}\left(\frac{A_x}{\sqrt{A_y^2+A_z^2}}\right) \qquad \text{Roll} = tan^{-1}\left(\frac{A_y}{\sqrt{A_x^2+A_z^2}}\right)$$



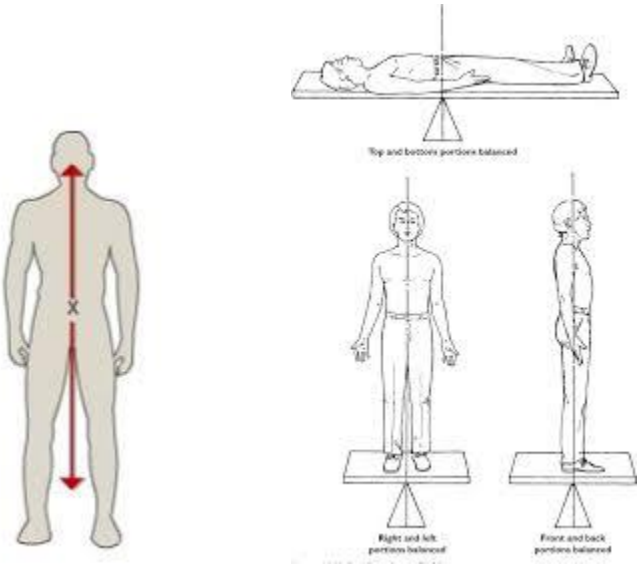
The sensor is working as a battery powered and mobile sensor.

Communication Network Layer:

Access Network sublayer: It defines the range and the topology of the communication between the "things" and later layers. The Bluetooth technology used by us has put a restriction on the range of our communications.
Access Technology: Wireless personal area network
Topology: Point to Point Topology, Star topology.
Gateways and Backhaul: The smart object often report to the Arduino Uno.
Network Transport sublayer: Bluetooth

Proposed algorithm:

The device has a better precision of the inclinations in both x and y axis when it is positioned with the center of mass and gravity of the object.



The center of gravity of the person is dependent on the position of the person but it gives a high sense of inclination in the X and Y axis.

The algorithm with a set of checkpoints to eliminate false alarms and increase the efficiency of the application. The application basically runs on the values received from the accelerometer itself now we have to extract all the parameters like Net Acceleration, Pitch and Roll. It is a 3-axis sensor. The data we can get from the accelerometer are the acceleration and the angular position in each axis. The net acceleration can be calculated by the sum of the squares of the acceleration in each axis.

$$A_{net} = \sqrt{A_x{}^2 + A_y{}^2 + A_z{}^2}$$

The x and y axis being parallel to the ground are zero whereas the z which is perpendicular is 1 in an ideal state. In an ideal stationary position the A net is its maximum i.e., 1g. During a free fall, the feeling of weightlessness is due to the net acceleration dropping to zero. The free fall is definitely different from the any ordinary fall. Hence the net acceleration is not zero but substantially less than 1. Along with the net acceleration we can calculate the angular position. The angle on the x, y and z is Roll, pitch and Yaw respectively. The angular position towards the z axis is not effective in the finding the required result. The roll and pitch are the relevant for our use case or say scenario.

$$\text{Pitch} = tan^{-1}\left(\frac{A_x}{\sqrt{A_y^2+A_z^2}}\right) \qquad \text{Roll} = tan^{-1}\left(\frac{A_y}{\sqrt{A_x^2+A_z^2}}\right)$$

Checkpoints:

1) The net acceleration starts decreasing as the fall starts. The net acceleration when the fall commences is less than 1g an average as suggested by few paper we have set the value to be around 0.8g
2) The net acceleration will have a sudden spike when the person hits the ground. this impact can be greater than 1 g and the time lag between the two checkpoint should be less than 2 sec
3) The value of pitch and roll should be calculated and evaluated in this case the threshold is set to more than 60 as the inclination during fall is confirmed with the inclination more and the time difference should be very less probably in milliseconds.
4) These are the functional of eliminating the false alarm and the probability of the user being falling is at peak, but not all falls need assistance. Now we see for 20 seconds for the person to get up by checking the inclination change, if the change in inclination doesn't occur we will confirm the fall.
5) This is the last resort of our project giving the person chance to disable the system and giving them this opportunity for nearly a minute.

The above checkpoints if met i.e., the net acceleration should be decrease to a threshold value to less than 0.8g when the fall commences and the net acceleration should result in a sudden spike of value which is slightly greater than 1 and this both should have a time gap of less than 2 seconds. Along with satisfying this the inclination values of the pitch and roll should be increased above 60 and in very short amount of time which is nearly a difference in milliseconds. This will result in detection of the fall and now waiting for the person to get by checking the inclination values along the X and Y axis for nearly 20 seconds and this will help us to evaluate whether the person needs some serious attention or not if the person gets up the alarm will not be triggered in other case the fall is confirmed and the signals are passed and piezo buzzer will change its frequency. Giving a grace period of sorts to the person to turn the buzzer manually to avoid sending the message it can either being able to receive help just from the buzzer . Even after the time given there is no response then the message is sent to the registered numbers.
The message passing is being done using a cloud service named Twilio and Temboo. When all the critical points are met the message passing is triggered and the values will be passed to the Temboo API which acts a framework for machine to machine communication. The Twilio acts a cloud based communication portal for sending a SMS to the registered user.

Experiment and Simulation Setup

Hardware:

Arduino UNO:

Arduino UNO is an open-source microcontroller developed by Arduino.cc. It is most used microcontroller in the Arduino range due to its easy usability and its capability of being interfaced to various expansion boards. It is developed on the basis of ATmega328P microchip. The board consists of both digital and analog pins and a programmable Arduino IDE. The power supply can

be given through a USB cable or an external 9 volt battery. It can accept the voltages between 7 and 20 volts. Some other specification include a 32 KB of flash memory with nearly 16 MHz clock speed.

Accelerometer:

We have used ADXL335

It is a 3 axis accelerometer. We have used it to find the and angular position and net acceleration. Some of the specifications are

- 3-axis sensing
- Small and compact packaging
- Works on low power
- Works on supply (1.8 to 3.6v)
- 10,000 g shock survival
- Good temperature resistance

Bluetooth Module:

We have used HC-05

It is used to transmit the data collected by the sensor to PC or device. It is easy to use and is encapsulated. It can work with any USB Bluetooth adapter. It works on very low power supply

3.3v.It is also known for being compatible with master mode, slave mode and both master-slave mode.

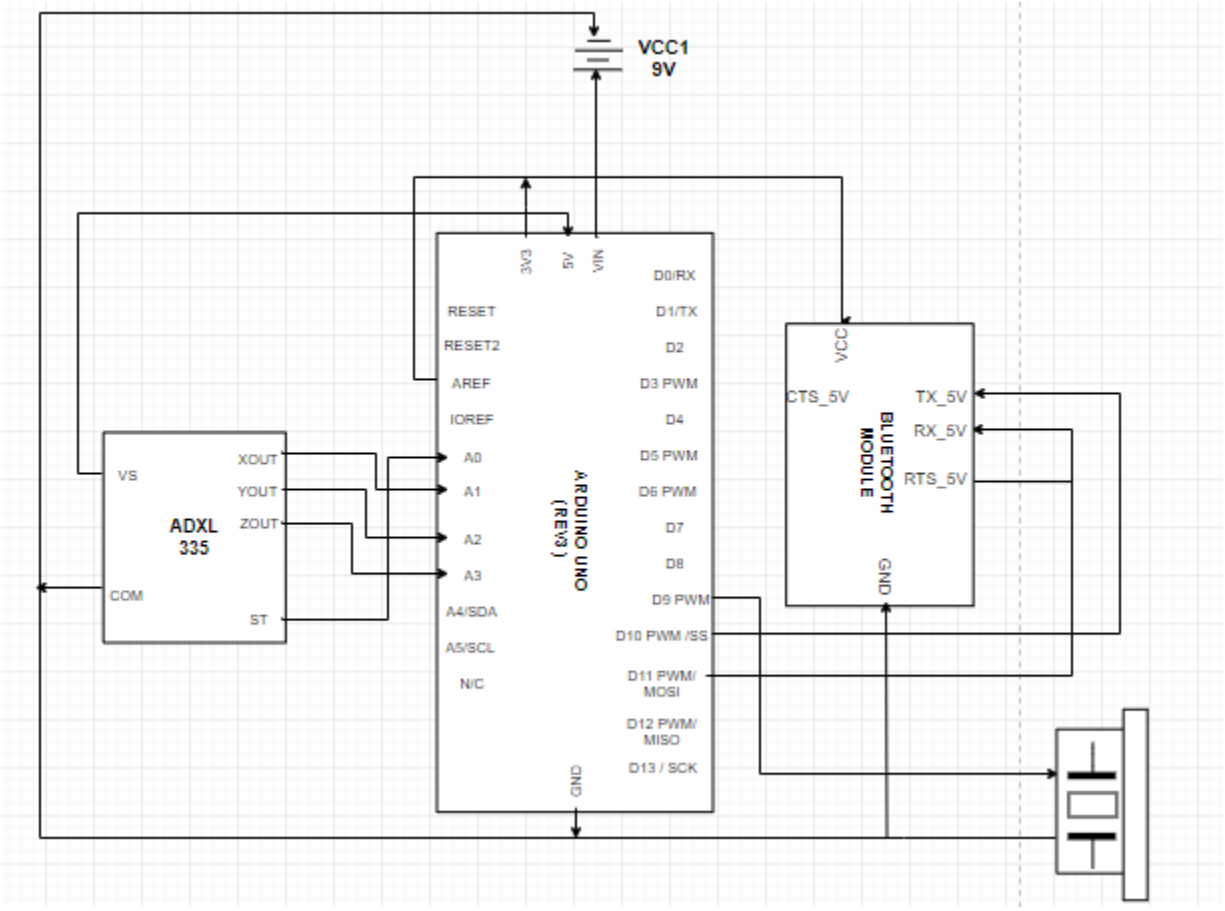
Piezo-buzzer:

It uses piezoelectricity to generate sound and the frequency can be given by the user in IDE.

Configuration:

Circuit diagram:

We have connected the Arduino board to the 9 volts battery and then the remaining hardware to the Arduino UNO. Below is the circuit diagram representing all the connections.



Code:

The project was coded in two different platforms i.e. on Arduino IDE and the other is Processing IDE. Arduino IDE uses basic C coding while Processing uses Java.

Arduino Code:

```
/*
The circuit:

ADXL335 (Accelerometer)
-----------------------
analog 0: accelerometer self-test
analog 1: z-axis
analog 2: y-axis
analog 3: x-axis
GND: ground
+5V: vcc

HC-05 (Bluetooth Module)
------------------------
+3.3V: Vcc
GND: ground
Digital Pin 10(RX): TX
Digital Pin 11(TX): RX

Piezo buzzer
------------
Digital Pin 9(PWM): +ve
GND: -ve

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Arduino Uno @ COM16---- via Arduino IDE
*/

#include <SoftwareSerial.h>

// Accelerometer pin description
const int xpin = A3; // x-axis
const int ypin = A2; // y-axis
const int zpin = A1; // z-axis
```

```
float Xval, Yval, Zval; // acceleration in g's along each axis

float Anet, pitch, roll; // Net acceleration (1when stationary), pitch and
roll convey orientation of accelerometer

unsigned long time = 0; // variable to compute time differences

int state = 0;  // condition of subject (0 indicates stable and
5 indicates fallen)

int ALERT = 0; // fall, perhaps?

//below are the calibration values for the 3 axes, such that in
stable condition the values are (1g, 0g, 0g)

float xZero = 505;
float yZero = 510;
float zZero = 525;

SoftwareSerial mySerial (10, 11); //RX, TX

void setup ()
{

mySerial.begin(9600); // initialize the software serial
communication
pinMode (9, OUTPUT); // PWM output for piezo buzzer

}


void loop ()
{

analogReference (EXTERNAL); // Enable Aref @ 3.3V

// Compute acceleration values in g's
Xval = (analogRead(xpin)- xZero)/102;
Yval = (analogRead(ypin) - yZero)/102;
Zval = (analogRead(zpin) - zZero)/102;
Anet = sqrt(sq(Xval) + sq(Yval) + sq(Zval));
```

```
// Pitch and roll calculation
pitch = atan(Xval/sqrt(pow(Yval,2) + pow(Zval,2)));

roll = atan(Yval/sqrt(pow(Xval,2) + pow(Zval,2)));

//convert radians into degrees
pitch = pitch * (180.0/PI) - 90;
roll = roll * (180.0/PI);

// Serial transfer of required variables to Processing
mySerial.print(pitch);
mySerial.print(",");
mySerial.print(roll);
mySerial.print(",");
mySerial.print(state);
mySerial.print(",");
mySerial.print(Anet);
mySerial.print(",");
mySerial.print("\n");

fall_detect(); // FALL DETECTION FUNCTION
delay(50); // Delay between passing values

}

void fall_detect()
{

if((Anet<0.8)&&(state==0)&&((abs(pitch)<60)&&(abs(roll)<60)))
// Initial drop in acc. due to free-fall like condition
{
      state = 1;
      time = millis();
}

if(state == 1)
{
      if(Anet>1) // Increase in acceleration due to impact of fall
      {
            state = 2;
            time = millis();
      }
```

```
      if(((millis()-time)/1000)>2) // max.permissible time between states 1
and 2
      state = 0;
}

if(state == 2)
      {
            if((abs(pitch)>60)||(abs(roll)>60))  //Senses  a  fall  in  any
      orientation by factoring pitch and roll
            {
                  state = 3;
                  time = millis();
            }

            if((millis()-time)>100) // max. permissible time between
                  states 2 and 3
            state = 0;
      }
if(state == 3)
      {
            if((abs(pitch)<60)&&(abs(roll)<60)) // Detects getting-up
and prevents false alarm
            {
                  count++;
                  if(count>20)
                  {
                        state = 0;
                        count = 0;
                  }
            }

            if(((millis()-time)/1000)>3) // 3 second window after fall
between states 3 and 4
            {
                  state = 4;
                  ALERT = 1;
                  tone(9,250); // Fall, perhaps?
            }
      }
```

```
if((state ==4)&&(ALERT==1))
      {
      if(((millis()-time)/1000)>20) // if patient is unresponsive
      for more than 20 secs, fall is confirmed-- necessary action
      taken via processing
            {
            state = 5;
            tone(9,4000); // Generate piezo tone for distress
            }
      }
}
```

Processing Code:

```
/*

Processing sketch to process and interpret serial data
received via Bluetooth module. Serves as the central reference
for doctors/ hospitals giving alerts for possible falls of person. Also
requires active Internet service to send message via 3rd party server.

*/

// for serial data interpretation
import processing.serial.*;
Serial myPort;

/*

Temboo is a cloud-based platform with 2000+ processes for
APIs, databases, and more. The Processing + Temboo library
enables us to connect with TWILIO, a cloud communication company that allows
developers to send/receive SMS and phone calls programmatically.
This solution is cost efficient, effective and is an alternate to a bulky,
power hungry GSM module.

 */

import com.temboo.core.*;
import com.temboo.Library.Twilio.SMSMessages.*;
int Para = 2; // parameters -- pitch and roll
```

```
float[] values = new float[Para+2]; //total of 4 serial variables
String status = "Stable";
int flag = 0;

// Create a session with your Temboo account details

TembooSession session = new TembooSession("p05eidon",
"myFirstApp", "UOJZDUuaZfTCnt6zxnZ7h9QAUpDSU7YG");

void setup()
{

// List all the available serial ports:

printArray(Serial.list());

// Port [0] is generally found to be the incoming Bluetooth
serial line, if no USB COM is active:

myPort = new Serial(this, Serial.list()[0], 9600);

// don't generate a serialEvent() until you get a newline
character:

myPort.bufferUntil('\n');
background(0);

// initialize pitch:

values[0] = 0;

// initialize roll:

values[1] = 0;

// state variable

values[2] = 0;

}
```

```
// empty draw() loop

void draw()
{
}

// Everything occurs in the serialEvent() loop

void serialEvent(Serial myPort) // get the ASCII string:
{

// Send SMS if fall Confirmed/ Distress
if((values[2] ==5)||(values[2]==42))
      {
            if(flag==0)
                  runSendSMSChoreo();
            flag = 1;
      }

String inputString = myPort.readStringUntil('\n');
if (inputString != null)
{
      // trim off any whitespace:

      inputString = trim(inputString);

      // splits the serial string on the delimiters and assigns it to
      a float array, here the delimiter can be a ,--[comma] ,
      --[space] , \t--[tab]

      values = float(splitTokens(inputString, ", \t"));

      print(values[2]);

      if(values[2]==5)
            status = "FALL confirmed";

      else if(values[2] ==4)
            status = "FALL sensed";

      else if(values[2] == 0)
            status = "Stable";
  }
}
```

```
// Function to generate SMS
void runSendSMSChoreo()
{

// Create the Choreo object using your Temboo session

SendSMS sendSMSChoreo = new SendSMS(session);

// Set credential

sendSMSChoreo.setCredential("qwerty1");

// Set inputs
// Run the Choreo and store the results

SendSMSResultSet sendSMSResults = sendSMSChoreo.run();

// Print results

println(sendSMSResults.getResponse());

}
```

Result and Analysis:

After many tests and calibrations, we were successful in completing the system. A system which is capable of detecting a fall when it is occurred and send an alert message to kin. When we are analyzing the system number of graphs were generated about which we are mentioning below.

We have generated a dataset which is made up of values from during various daily life activities like sitting, sleeping, walking etc.

The parameters which we are considering are

- Net Acceleration

The initial net acceleration is generally 1g (in ideal conditions). But when a fall occurs the net acceleration is less than that 0 (in the case of free fall). Since when a person falls, it is not a free fall, so according to our algorithm the fall sense threshold is less than 0.8g.

After a fall commences, i.e. after the person hits the ground, there will be a rise in the net acceleration for an instant.

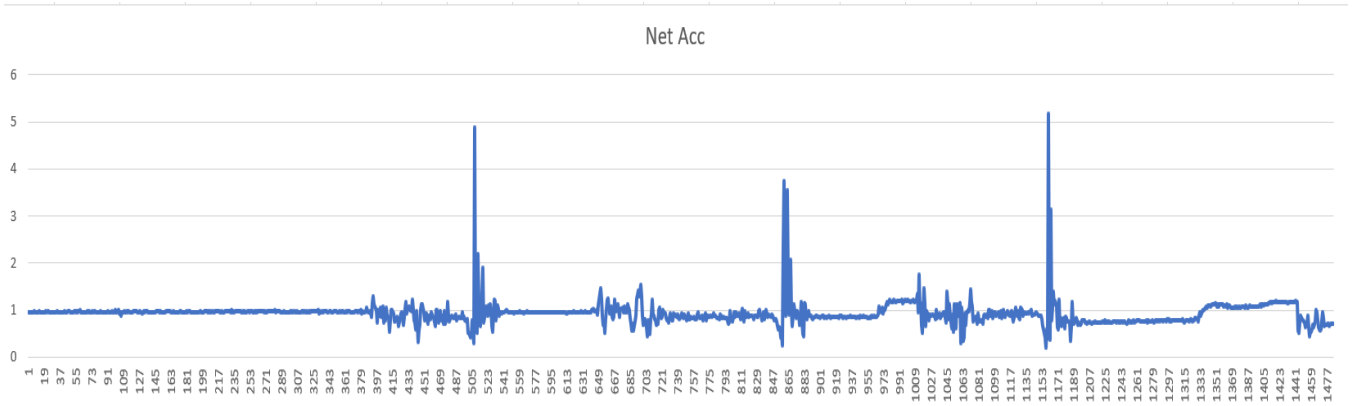


Fig 1.0 – graph of net acceleration over the generated dataset.

- Pitch

Pitch is the orientation (angle) along the x-axis. This is one of the most important factors in determining the fall. According the algorithm, If the pitch is greater than 60 degrees, there might be a chance of fall.

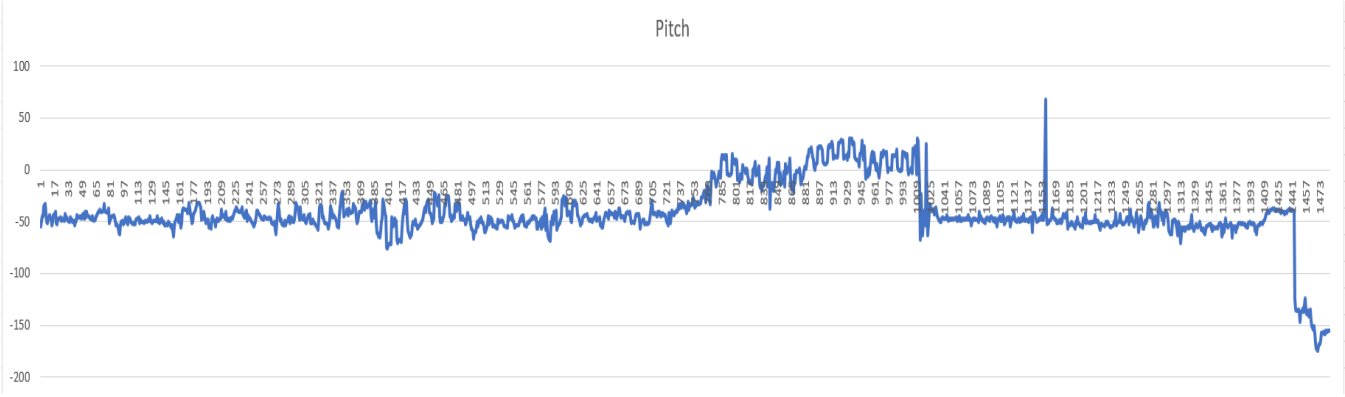


Fig 2.0 – graph of pitch over the generated dataset.

- Roll

Roll is the orientation (angle) along the Y-axis and is also an important parameter the finding/ detecting the fall. According to our algorithm, 60 degrees is the threshold.

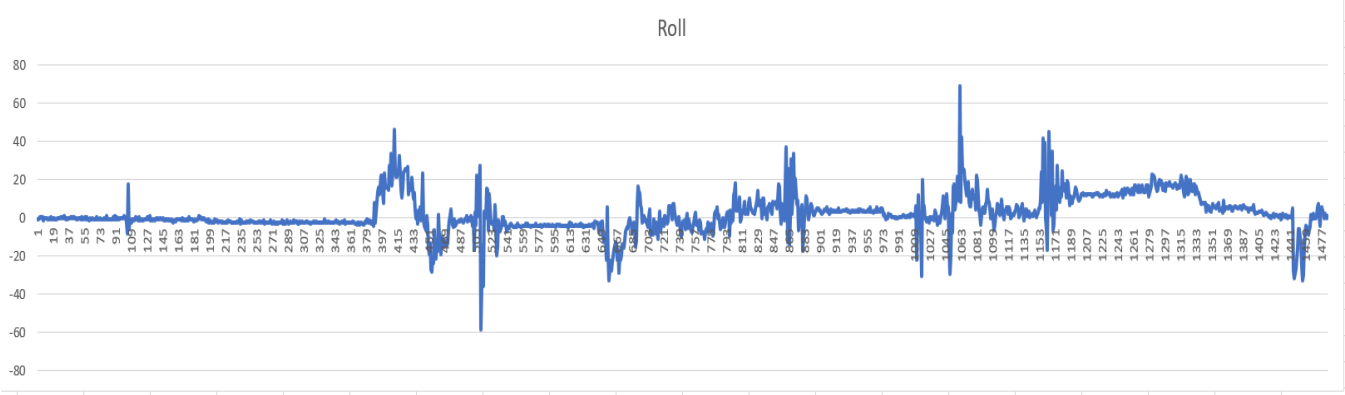


Fig 3.0 – graph of roll over the generated dataset.

When all the parameters show abnormality at the same time (there may be times when one or two of the parameters are crossing the threshold because of ADL's) then we can say that a fall is detected. We have created a unified parameter called the state which shows the state of fall.

- State 1 – Stable
- State 3 – fall sensed
- State 4 – fall confirmed
- State 5 – SMS state

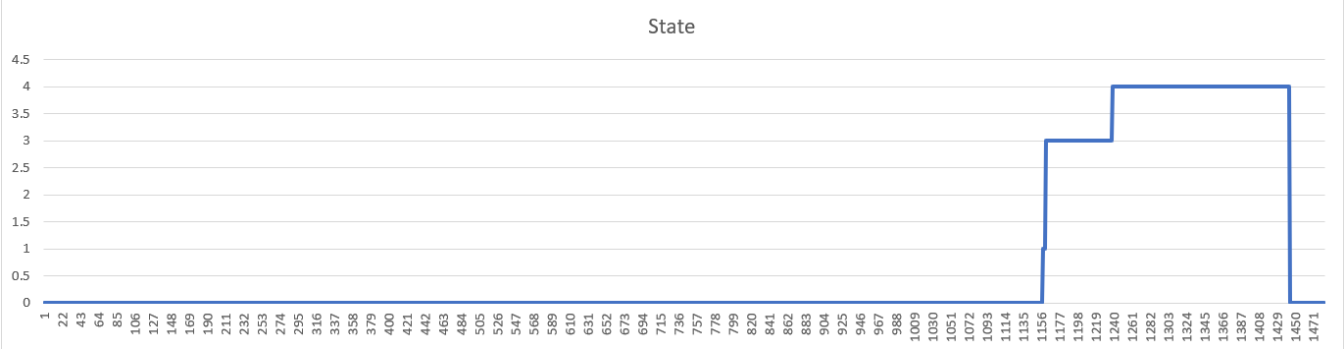


Fig 4.0 – graph of state over the generated dataset.

Excel Sheet Data:

Pitch	Roll	State	Net Acc
-54.52	-0.58	0	0.96
-49.44	-0.58	0	0.96
-44.83	0	0	0.96
-45.27	0.59	0	0.96
-35.45	0.6	0	0.94
-33.53	0	0	0.96
-43.2	0.6	0	0.94
-50.81	-1.14	0	0.98
-51.24	0	0	0.96
-47.43	0	0	0.96
-43.84	0	0	0.96
-47.04	-0.58	0	0.96
-46.93	-0.58	0	0.97
-53.03	0	0	0.96
-50.54	-0.58	0	0.96
-44.15	0.59	0	0.95
-43.1	-0.58	0	0.96
-41.16	0	0	0.96
-51.21	-0.58	0	0.96
-52.18	0	0	0.96
-48.6	-0.57	0	0.98
-47.27	-0.58	0	0.96
-47.38	-0.58	0	0.97
-46.3	0	0	0.96
-48.69	0	0	0.96
-47.27	0	0	0.96
-47.99	0	0	0.96
-48.94	0.6	0	0.94

Conclusion and Future work:

We have implemented a complete end to end solution of fall detection system for elderly. To improve evaluation metrics like sensitivity and specificity, we need to use more sensors and also apply various sophisticated machine learning techniques like CHMMs, SVM, random forests, XDABoost, MEMM etc. to provide resilient fall detection system.

Future fall detection system would include more than 20 sensors apart from accelerometer and gyroscope which will give the condition of the body. It would carry out same sophisticated process as in smart home deployment to outside world deployment. Since elderly prefer to have lightweight device which doesn't interfere with their daily activity, Nano sensors and computing should allow embedding of fall detection system into their clothes hidden from world view.

Bibliography:

1) Testing of a Long-Term Fall Detection System Incorporated into a Custom Vest for the Elderly. Alan K. Bourke, Pepijn W.J. van de Ven, Amy E. Chaya, Gearóid M. ÓLaighin, and John Nelson

2) A Posture Recognition-Based Fall Detection System for Monitoring an Elderly Person in a Smart Home Environment. Miao Yu, Adel Rhuma, Syed Mohsen Naqvi, Member, IEEE, Liang Wang, Senior Member, IEEE, and Jonathon Chambers, Fellow, IEEE.

3) A High Reliability Wearable Device for Elderly Fall Detection Paola Pierleoni, Alberto Belli, Lorenzo Palma, Marco Pellegrini, Member, IEEE, Luca Pernini, and Simone Valenti.

4) Intelligent Video Surveillance for Monitoring Fall Detection of Elderly in Home Environments Homa Foroughi1, Baharak Shakeri Aski2, and Hamidreza Pourreza11 Department of Computer Engineering, Ferdowsi University of Mashhad, Mashhad, Khorasan, Iran2 Islamic Azad University of Ramsar, Ramsar, Mazandaran, Iran.

5) Privacy Preserving Automatic Fall Detection for Elderly Using RGBD Cameras Chenyang Zhang1, Yingli Tian1, and Elizabeth Capezuti2.

6) Evaluation of a threshold-based tri-axial accelerometer fall detection algorithm A.K. Bourke a,*, J.V. O'Brien b, G.M. Lyons.

Plagiarism Report:

Total Words – 3730

The plagiarism checker takes only 1000 words as input, so we divided the doc into 4 parts with 1000 , 1000, 1000 and 730 respectively and below is the result.