fallenPersonDetector

By

Jacopo Carlon , Nicola Riccardi

Problem Introduction

- In the last decades, there has been a huge increase in life expectancy, meaning that in the future the absolute amount of elderly people will increase.

- Sadly, fall accidents are the leading cause of injury for adults over 65 years old, as well as the leading cause of injury-related death for the same group of people :
- Age-adjusted fall death rate has increased by 42% between years 2012 and 2021.

Proposed approach

- It should also be noted that the future generations will most likely be using their **mobile phones** a lot, considering that they will have seen such technologies for almost all their lives.
- Focusing on an Android mobile device, we will use data from its accelerometer in order to promptly detect a fall event, alerting nearby people in an ask for help.

Our choices

Some real life situations :

- Phone falls from hand during call/use … nothing bad
- Phone falls from table … nothing bad
- Phone is in pocket and is falling ?? THE PERSON IS FALLING !!!!! (bad)

In front of us :

- *Many (many) false positives* : jumping, running, phone falls from hand …
  -> … We don't want to alarm the authorities over everything … (in some places it is also illegal) … read "the boy who cried wolf"
- *Need to minimize false negatives* : if it might be a fall, we should take action just in case

->

We will not call emergency number for every fall detected, but we will **make a lot of noise** (== call *nearby* help) in any such occasion .

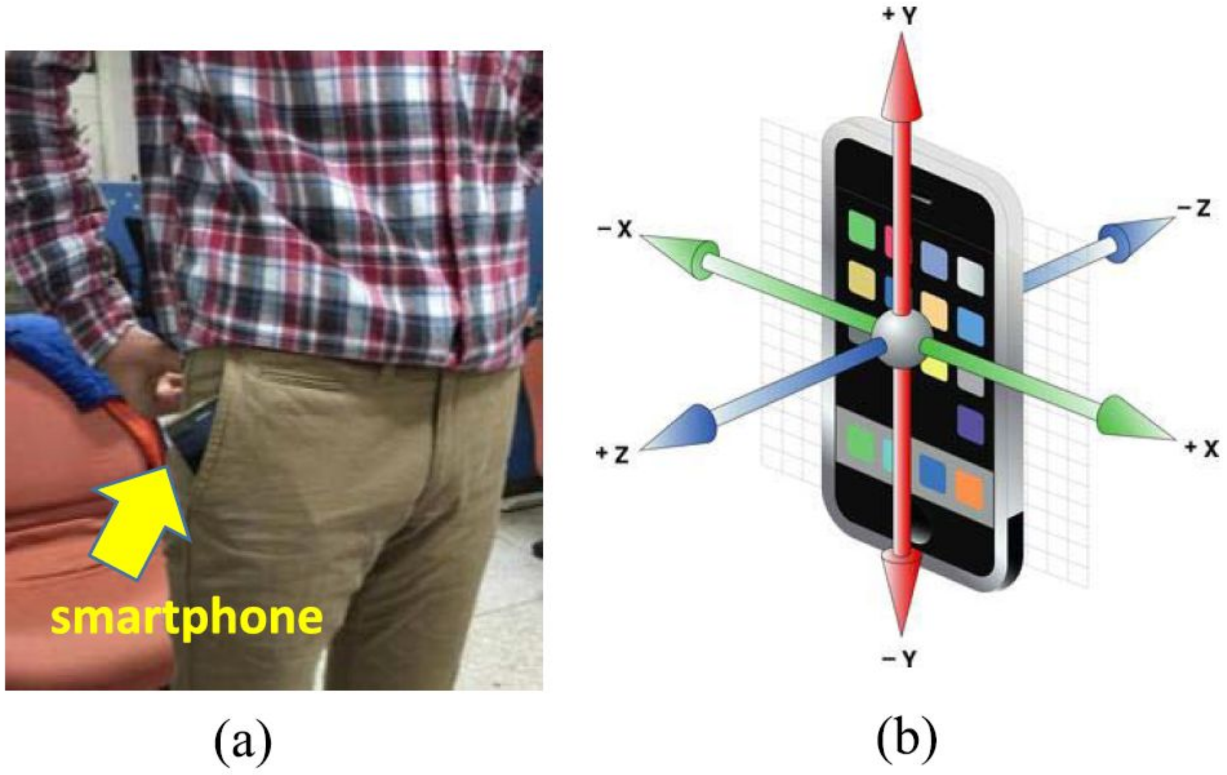
Most of all, **NO FALSE NEGATIVE** are allowed.

How to measure accelerations

- H : Phone is in front pocket
- Acceleration sensor : sampling 50 Hz (once every 20 ms)
- Sometimes sampling pauses / has delays …

  -> will linearize it via software
- If phone is on table, records *g* …

  -> We use TYPE_LINEAR_ACCELERATOR, which removes g (projected on the 3 axes)



(a) (b)

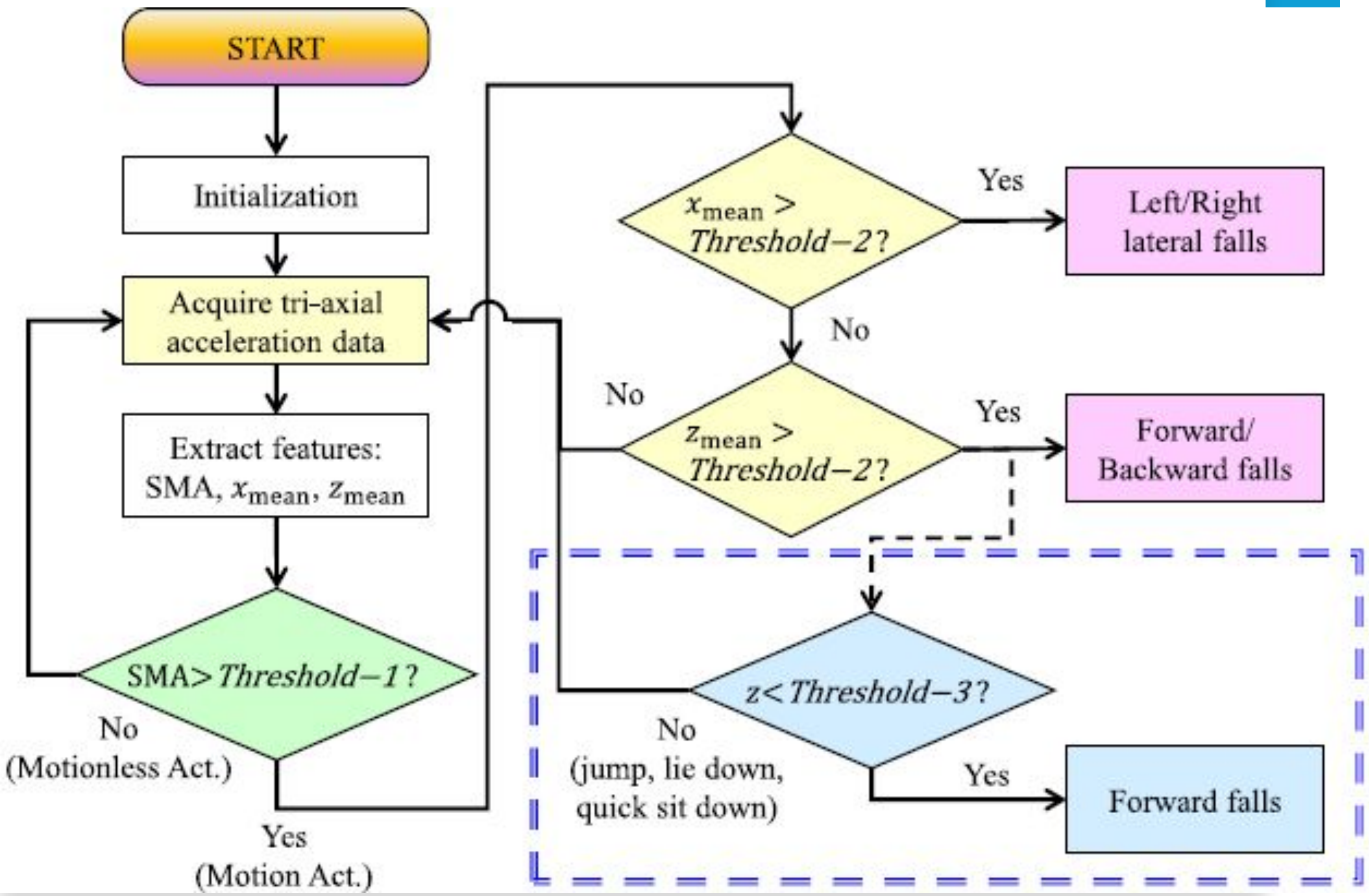
Detection Plan :

as per paper “Development of an Enhanced Threshold-Based Fall Detection System Using Smartphones With Built-In Accelerometers”
by Jin-Shyan Lee, and Hsuan-Han Tseng

- Initialize sensing service
- Get new acceleration data
- Check SMA > Th_1 else NOP
- Check avgAbsXacc > Th_2 -> ..Fall
- Check avgAbsZacc > Th_3 -> ..Fall
- If instZacc (negative) < Th_3 : ..Fall

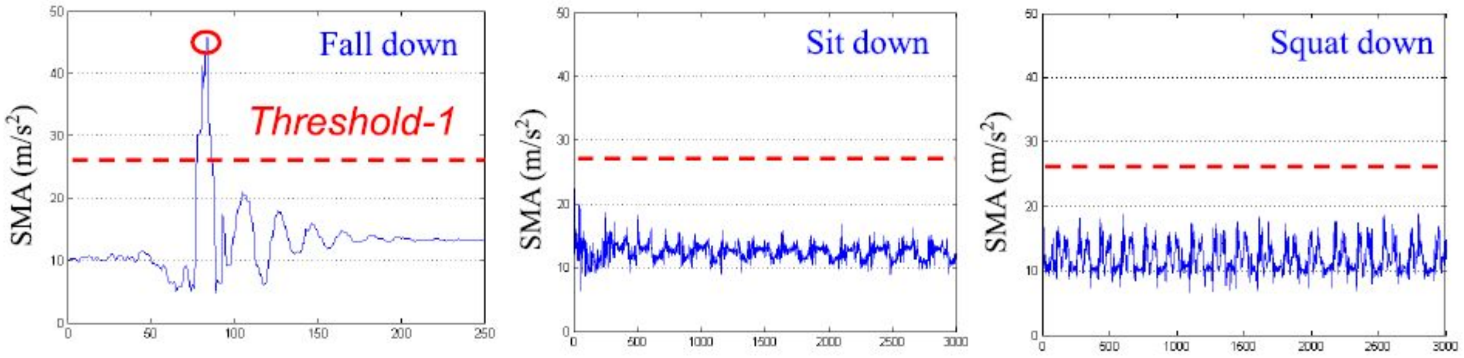
Fall -> make sound !!!

… with some twists, we’ll show


START
Initialization
Acquire tri-axial acceleration data
Extract features: SMA, x_{mean}, z_{mean}
SMA>*Threshold−1*?
No (Motionless Act.)
Yes (Motion Act.)
x_{mean} > *Threshold−2*?
Yes
Left/Right lateral falls
No
z_{mean} > *Threshold−2*?
No
Yes
Forward/ Backward falls
z<*Threshold−3*?
No (jump, lie down, quick sit down)
Yes
Forward falls

Is subject moving ("dangerously") ..?

We know any movement can kill you, but we need to consider the ones that are not "walking" or "sitting"



We extract Signal Magnitude Area as average of module of recent accelerations recorded :

$$\mathrm{SMA}[n] = \frac{1}{N} \sum_{i=n-N+1}^{n} \left(|x[i]| + |y[i]| + |z[i]|\right)$$

And use the experimental threshold Th_1 = 27.0 m/s^2 .

Lateral acceleration ? (x -axis)

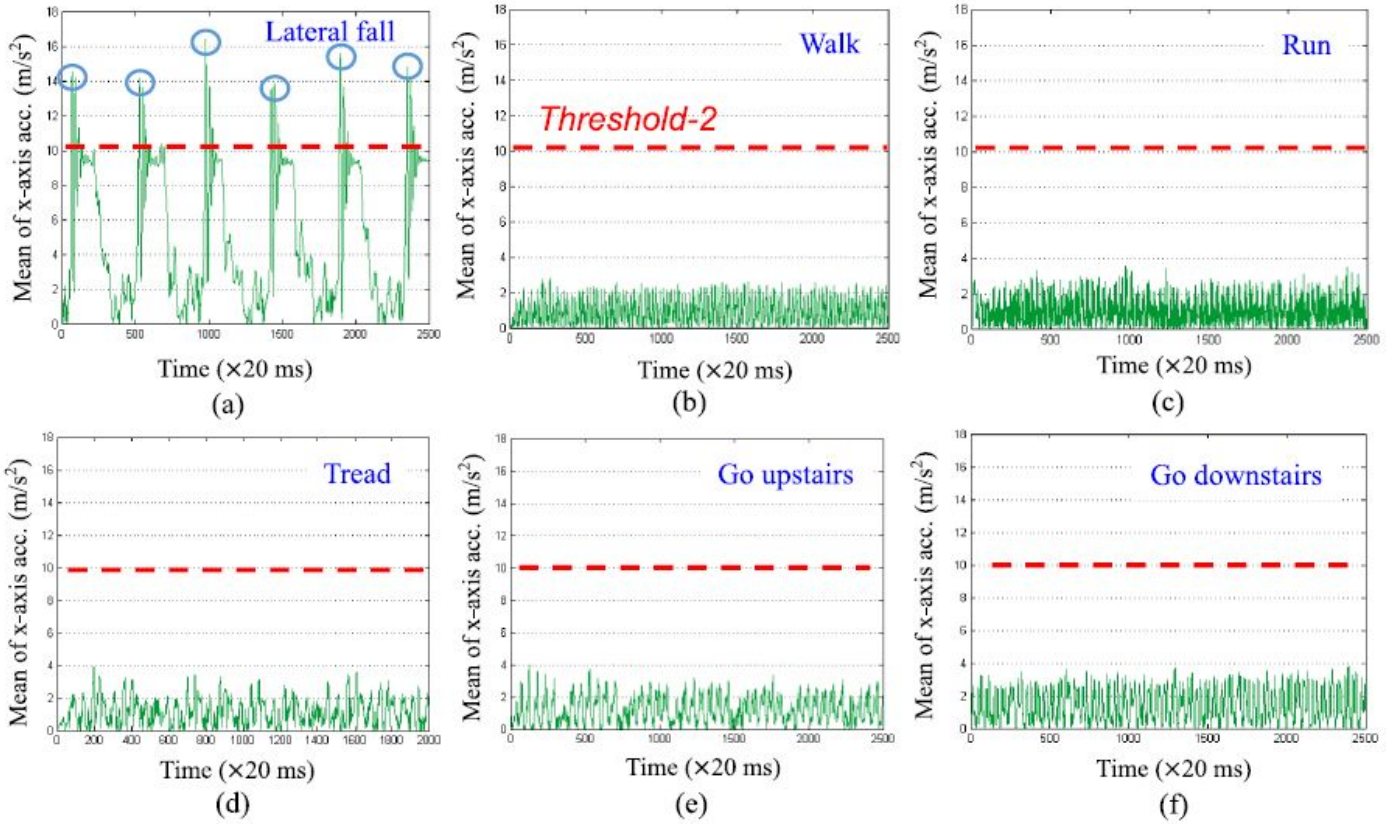
Extract average absolute acceleration on x-axis :

$$x_{\text{mean}}[n] = \frac{1}{N}\left(\sum_{i=n-N+1}^{n} |x[i]|\right)$$

And test it against experimental threshold Th_2 = 10.05 m/s^2

If xAbsMean > Th_2,

-> surely it is a **lateral fall** !!!



(a) (b) (c)

(d) (e) (f)

z-axis acceleration

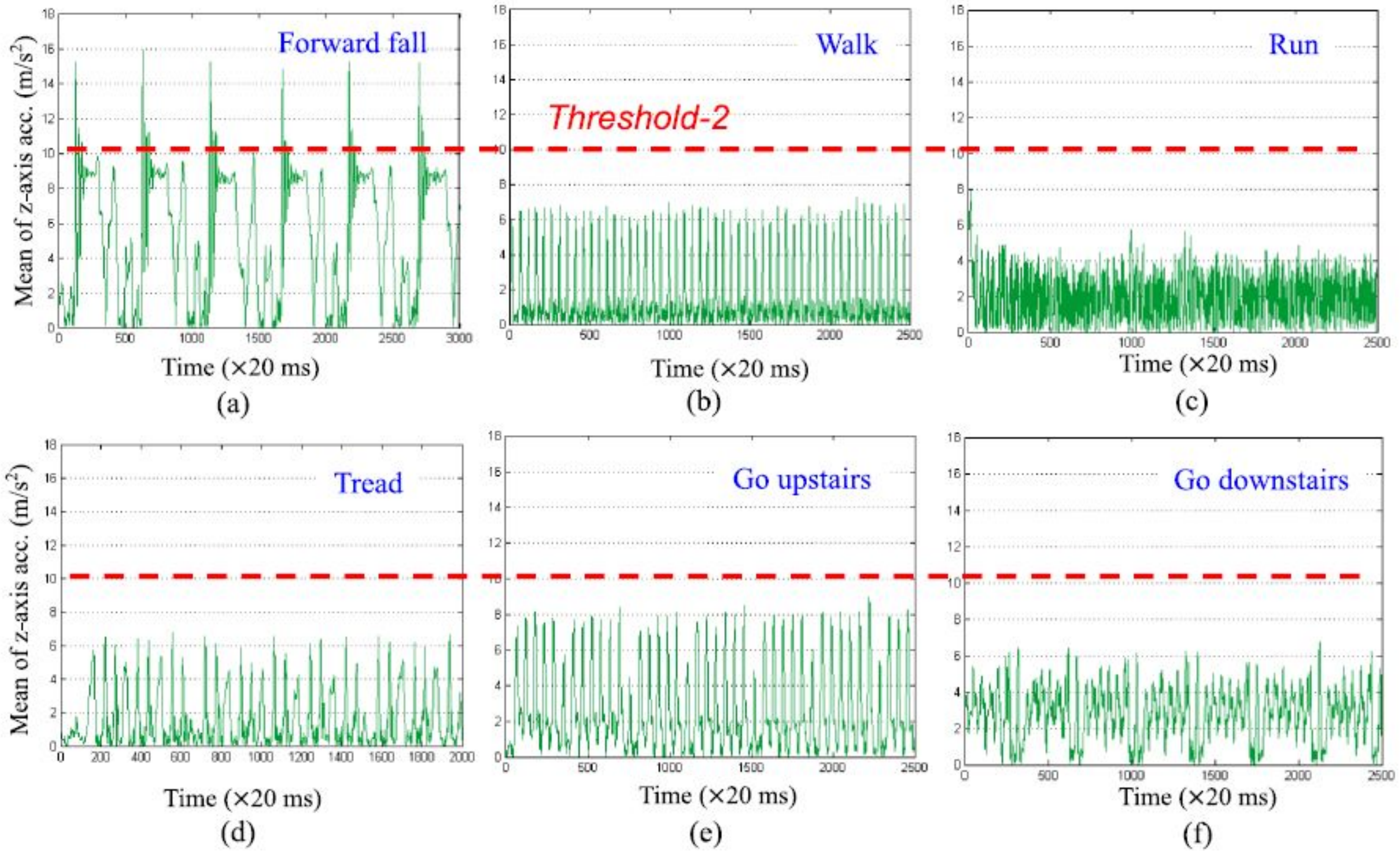
Same procedure as before, test :

If zAbsMean < Th_2

-> it is not a fall

Else could be :

- Forwards fall
- Backwards fall
- … other ?



(a) (b) (c)

(d) (e) (f)

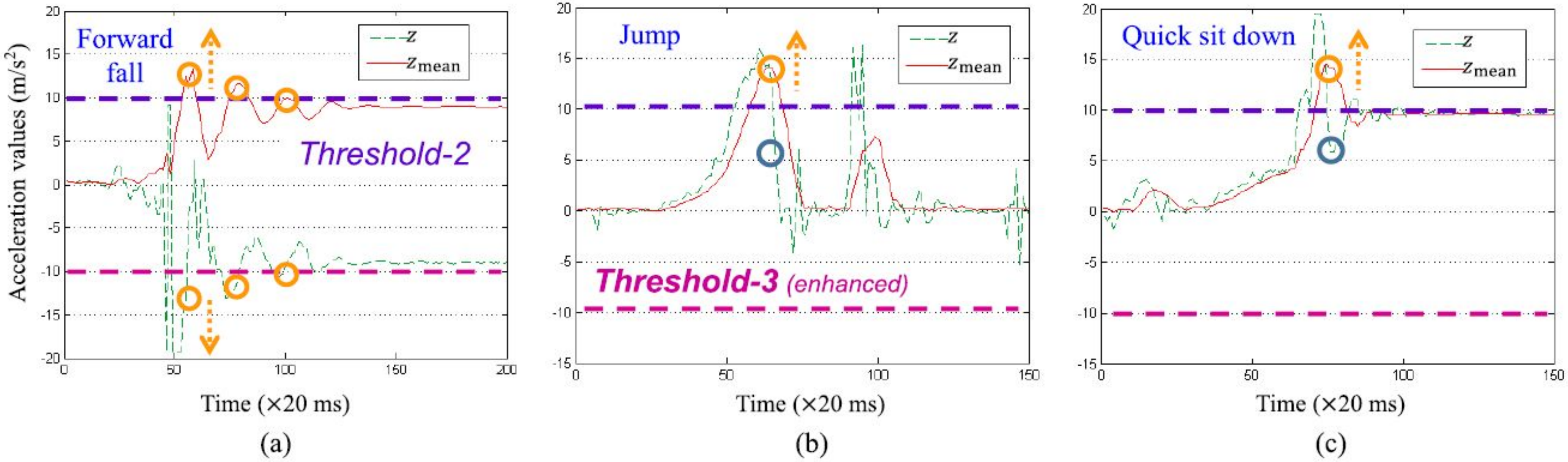
z-axis acceleration

Same procedure as before, test :

If zAbsMean < Th_2 -> it is not a fall .

Else could be :

- Forwards fall
- Backwards fall
- … other ?



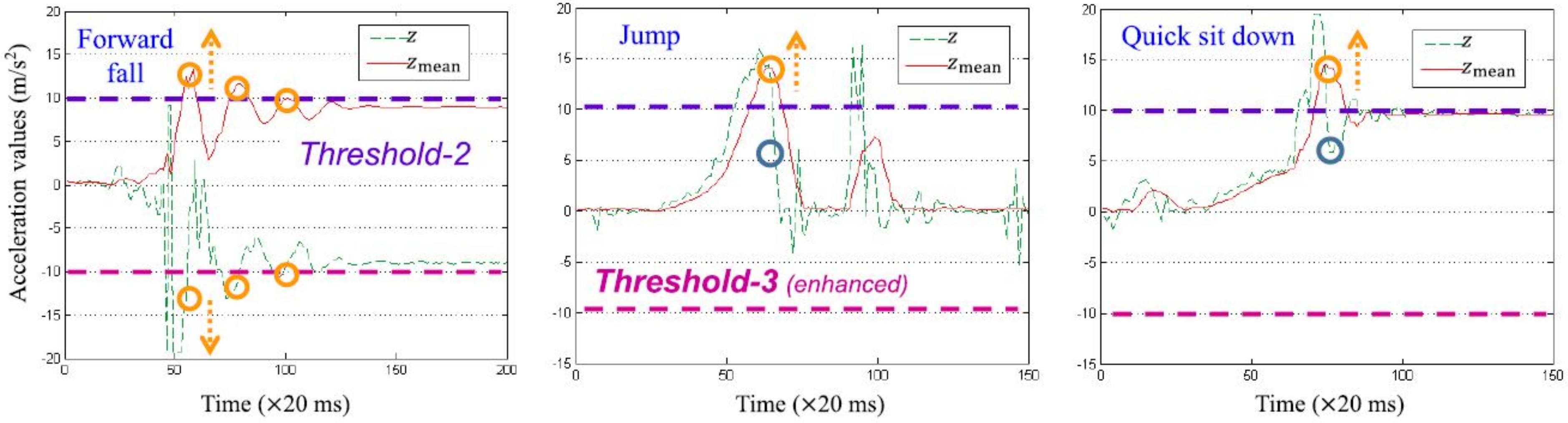
Z-axis instant acc observations :

During forwards fall, instant acceleration is negative, z_inst < -10m/s^2 (called Threshold Th_3)
That is a clear distinguisher of forwards falls.

However, what if only zAbsAvg > Th_2 ? Could actually be :

- Forwards fall -> bad, detected by Th_3
- Backwards fall -> bad
- Jump, quick sit down -> not bad (and quite unlikely for elderly people

-> Our priority is NO FALSE NEGATIVE -> any case of zAbsAvg > Th_2, we will consider it a fall !



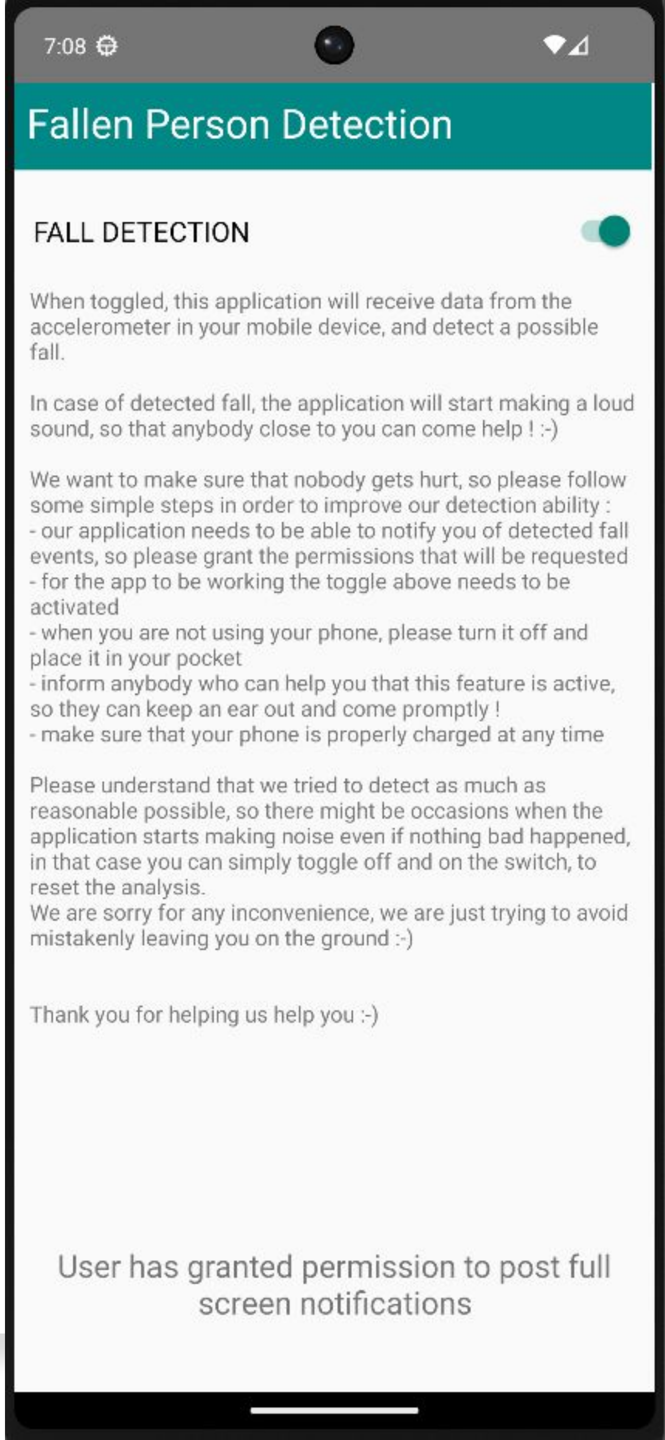
(a) (b) (c)

Our implementation

- Application **main activity** has *toggle to start/stop our **Service*** (which will run in background until stopped or the app is closed)

- The Service samples accelerometer every 20ms (==50Hz), calculate SMA, xAbsMean, zAbsMean, and checks if there is a fall event

- If a fall happened, make noise until stopped (hope we get some help for owner)



Fallen Person Detection

FALL DETECTION

When toggled, this application will receive data from the accelerometer in your mobile device, and detect a possible fall.

In case of detected fall, the application will start making a loud sound, so that anybody close to you can come help ! :-)

We want to make sure that nobody gets hurt, so please follow some simple steps in order to improve our detection ability :
- our application needs to be able to notify you of detected fall events, so please grant the permissions that will be requested
- for the app to be working the toggle above needs to be activated
- when you are not using your phone, please turn it off and place it in your pocket
- inform anybody who can help you that this feature is active, so they can keep an ear out and come promptly !
- make sure that your phone is properly charged at any time

Please understand that we tried to detect as much as reasonable possible, so there might be occasions when the application starts making noise even if nothing bad happened, in that case you can simply toggle off and on the switch, to reset the analysis.
We are sorry for any inconvenience, we are just trying to avoid mistakenly leaving you on the ground :-)

Thank you for helping us help you :-)

User has granted permission to post full screen notifications

Activities details

MainActivity

- Shows a description of the service, and has a switch button.
- The first time the user tries to start the activity, two permissions are required:
    - POST_NOTIFICATIONS: to send notifications
    - USE_FULL_SCREEN_INTENT: to associate a Full Screen Intent to an urgent notification
- Once all permissions are granted, checking the switch will start the *AccelerometerService*.

AlarmActivity

- Used as a Full Screen Intent
- Loops an alarm at full volume until stopped

AccelerometerService details

- At startup, registers an handler thread to *Linear Acceleration* sensor and creates a *Notification Channel*

- Data are evaluated using a support class (**StatusSelector**)

- If a fall is detected, sends an *high-priority Notification* associated with a Full Screen Intent to start **AlarmActivity**

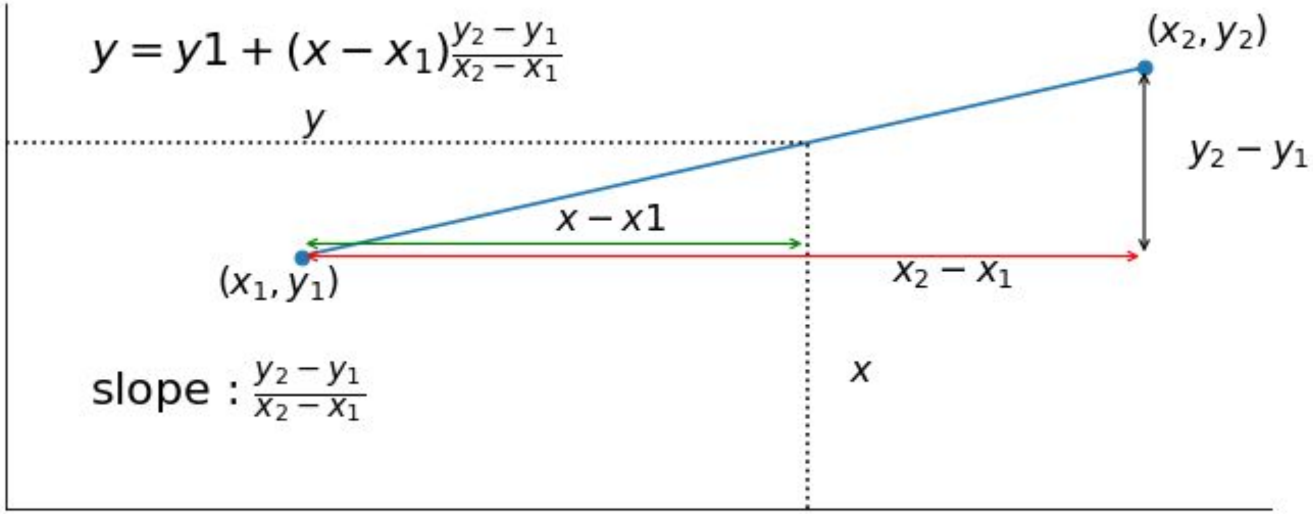
App behaviour with screen turned off :

- Detected fall is notified via an high-priority notification
- The notification is associated with a *FullScreenIntent*
- If the screen is off, the Intent starts automatically, firing up the alarm

Some specific details : linear interpolation

- We know hardware accelerometer can sometimes miss to sample, this could be important since we use averages (we would rather use properly populated arrays)
- -> when we receive a new value at a new time, we linearize and fill the missed values :

$$trgV = oldV + (newV - oldV) * \frac{trgT - OldT}{NewT - OldT}$$



$y = y1 + (x - x_1)\frac{y_2 - y_1}{x_2 - x_1}$
(x_2, y_2)
y
$y_2 - y_1$
$x - x1$
(x_1, y_1)
$x_2 - x_1$
slope : $\frac{y_2 - y_1}{x_2 - x_1}$
x

Conclusions

- Application works in target situation : background, in pocket, with phone turned off

Possible expansions :

- Add a call to some emergency number, this choice would require a reasonable balance between effective threat and FalsePositives (our minimizing FN risks an increase in FP)

Other notes:

Accelerometer uses 10 times **less** energy than any other sensor, so we consider the energy-safety trade-off worth it.