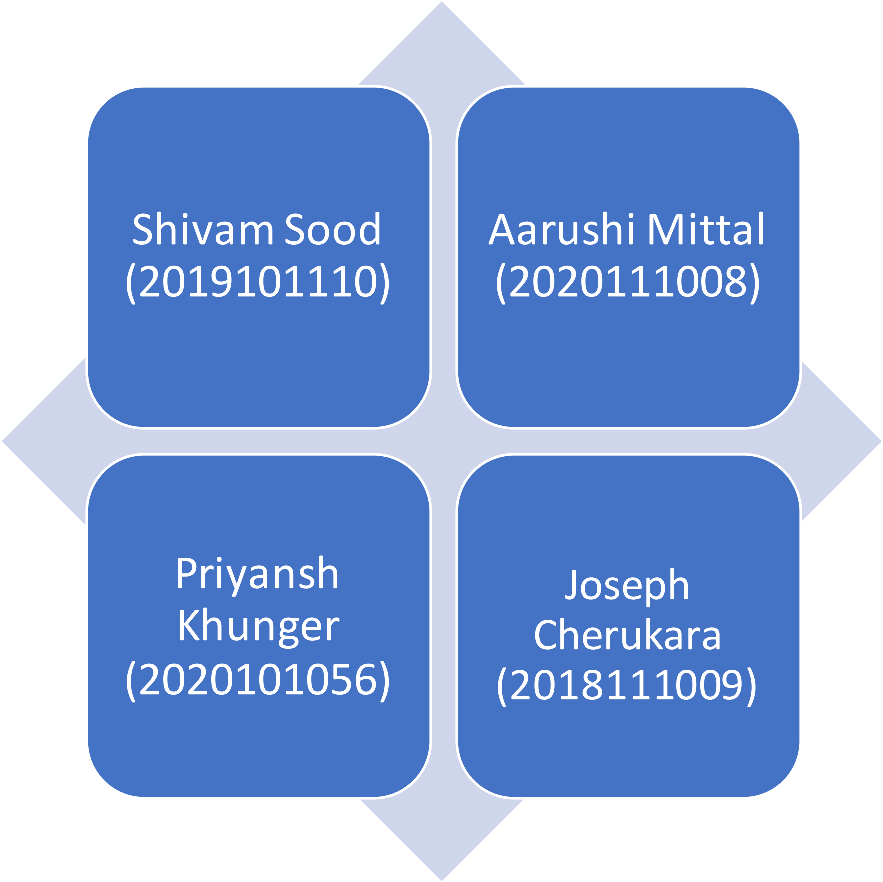
ESW Project
Fall Detection

Team -39

Team



Shivam Sood (2019101110)

Aarushi Mittal (2020111008)

Priyansh Khunger (2020101056)

Joseph Cherukara (2018111009)

Motivation

Construction workers and such at times during their work may fall from heights leading to injuries and accidents which may be treated properly if given care at early stages itself. But this does not happen most of the times since the injured can not move and others may notice them only at a later stage.

These kind of accidents can be dealt with earlier itself if we get to implement a fall detection system for these workers and thus lead to safer environments where in case a worker does fall, he can be administered care or taken to the hospital very early itself due to the notice

Project Idea

- Have a hardware setup that collects the accelerometer and gyroscope data in real time and uses that data to predict if a fall has happened or not in real time.
- This prediction model should implement an ML approach instead of a traditional deterministic approach
- The ML model should be lightweight to run on the microcontroller setup we use
- The prediction should be communicated to a central server which is visible at all times by the respective authorities

Setup

Components:

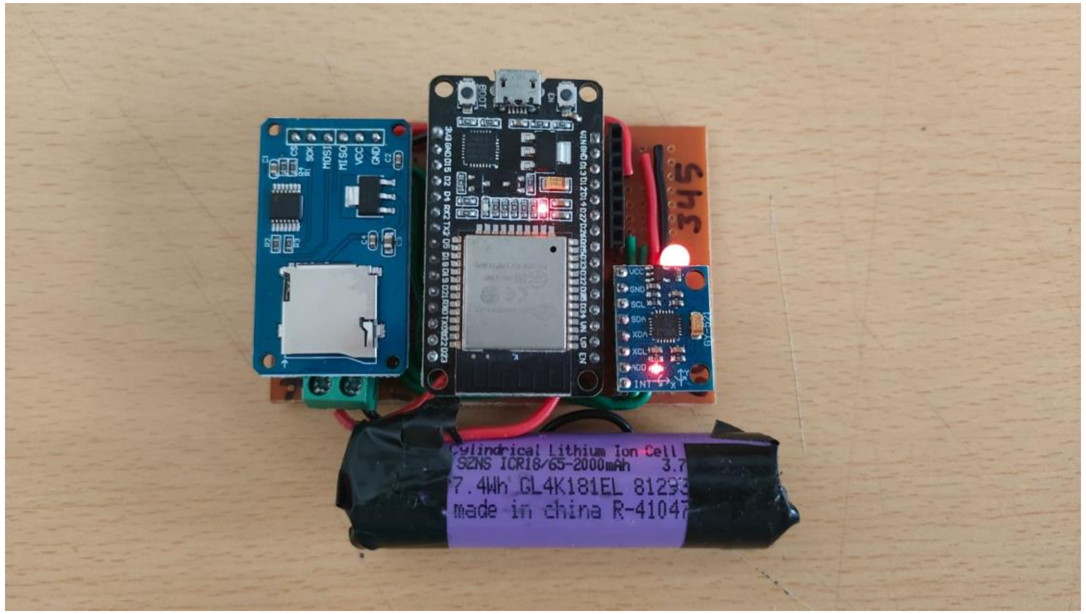
- ESP32 board
- Accelerometer and gyroscope module MPU-6050
- SD card storage
- Battery
- LoRa module

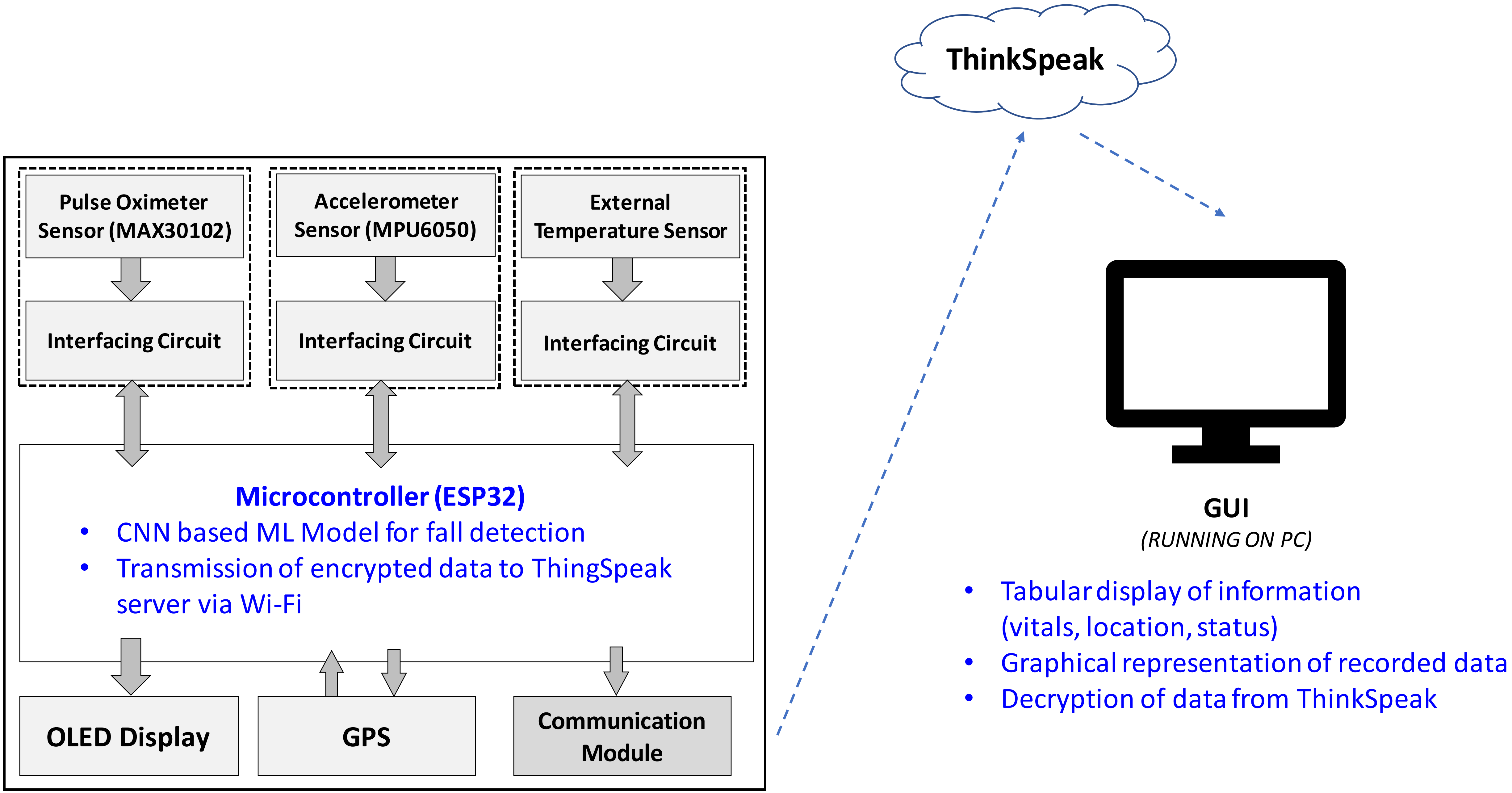
The accelerometer and gyroscope module collects the respective values at 250Hz

This collected data is sent and stored in the Sd card

LoRa module is used to communicate with the base transmitter

Setup


CS
SCK
MOSI
MISO
VCC
GND
345
VCC
GND
SCL
SDA
XDA
XCL
ADD
INT
Cylindrical Lithium Ion Cell
SZNS ICR18/65-2000mAh 3.7
7.4Wh GL4K181EL 81293
made in china R-41047



ThinkSpeak

Pulse Oximeter Sensor (MAX30102)

Interfacing Circuit

Accelerometer Sensor (MPU6050)

Interfacing Circuit

External Temperature Sensor

Interfacing Circuit

Microcontroller (ESP32)

- CNN based ML Model for fall detection
- Transmission of encrypted data to ThingSpeak server via Wi-Fi

OLED Display

GPS

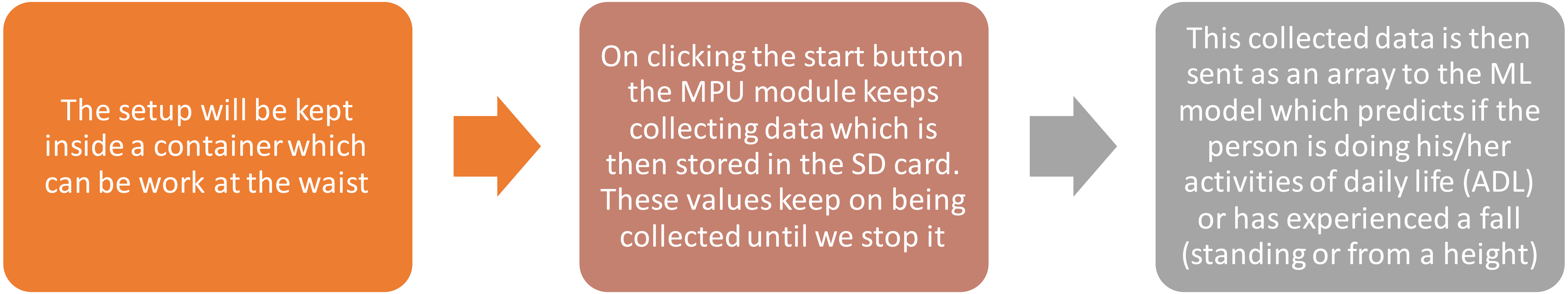
Communication Module

GUI

(RUNNING ON PC)

- Tabular display of information (vitals, location, status)
- Graphical representation of recorded data
- Decryption of data from ThinkSpeak

Working of Setup



The setup will be kept inside a container which can be work at the waist

On clicking the start button the MPU module keeps collecting data which is then stored in the SD card. These values keep on being collected until we stop it

This collected data is then sent as an array to the ML model which predicts if the person is doing his/her activities of daily life (ADL) or has experienced a fall (standing or from a height)

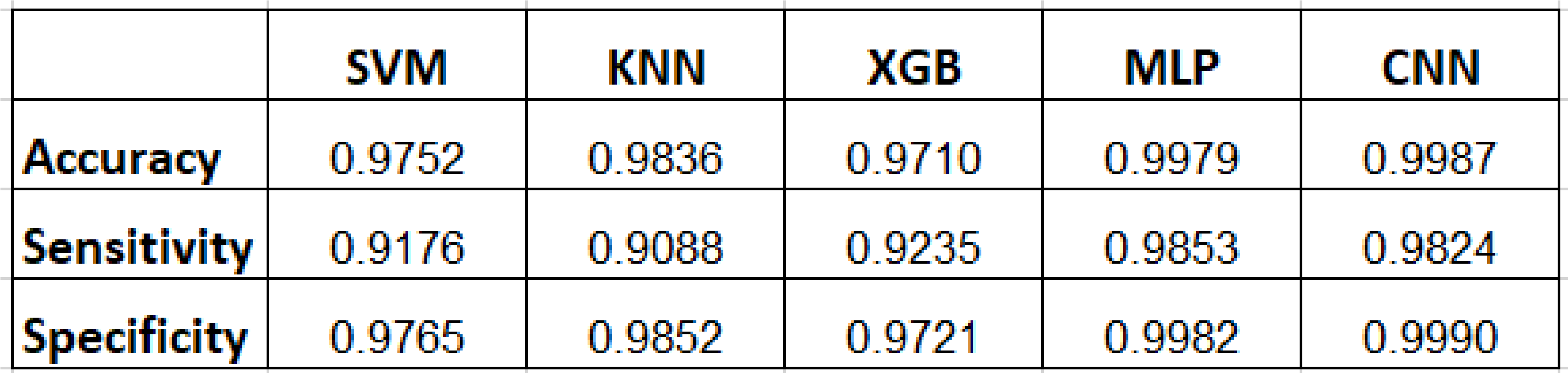
Data Collection

Fall: Using this setup, we first click on the button to start the data collection process and then drop/throw the setup kept inside a box with styrofoam, from a certain height.

ADL: Using this setup we click on the start button and then continued with the activity while keeping the setup in the pocket (eg: for running, walking, jumping, etc)

Once the button is clicked the MPU module keeps collecting data which is then stored in the SD card. These values keep on being collected until we stop it

ML Model accuracy comparison for Sisfall Dataset



	SVM	KNN	XGB	MLP	CNN
Accuracy	0.9752	0.9836	0.9710	0.9979	0.9987
Sensitivity	0.9176	0.9088	0.9235	0.9853	0.9824
Specificity	0.9765	0.9852	0.9721	0.9982	0.9990

- Since CNN was the highest we chose CNN as our ML model

CNN Model

The structure of the CNN model used is one convolutional layer of ten neurons, with relu activation, followed by one dense layer of two neurons with softmax activation. The CNN layer currently has kernel size of (50,6) and stride is 5. This gives approx 4500 parameters.

We use SMOTE [Synthetic Minority Oversampling Technique] for fixing the class imbalance. This is used along with random undersampling

[Old Values: 30 Fall, 9000 ADL] [Sensitivity was less than 0.5]

[New Value: 1000 Fall, 2000 ADL] [Sensitivity was 1.0]

Porting to ESP32 and RPi

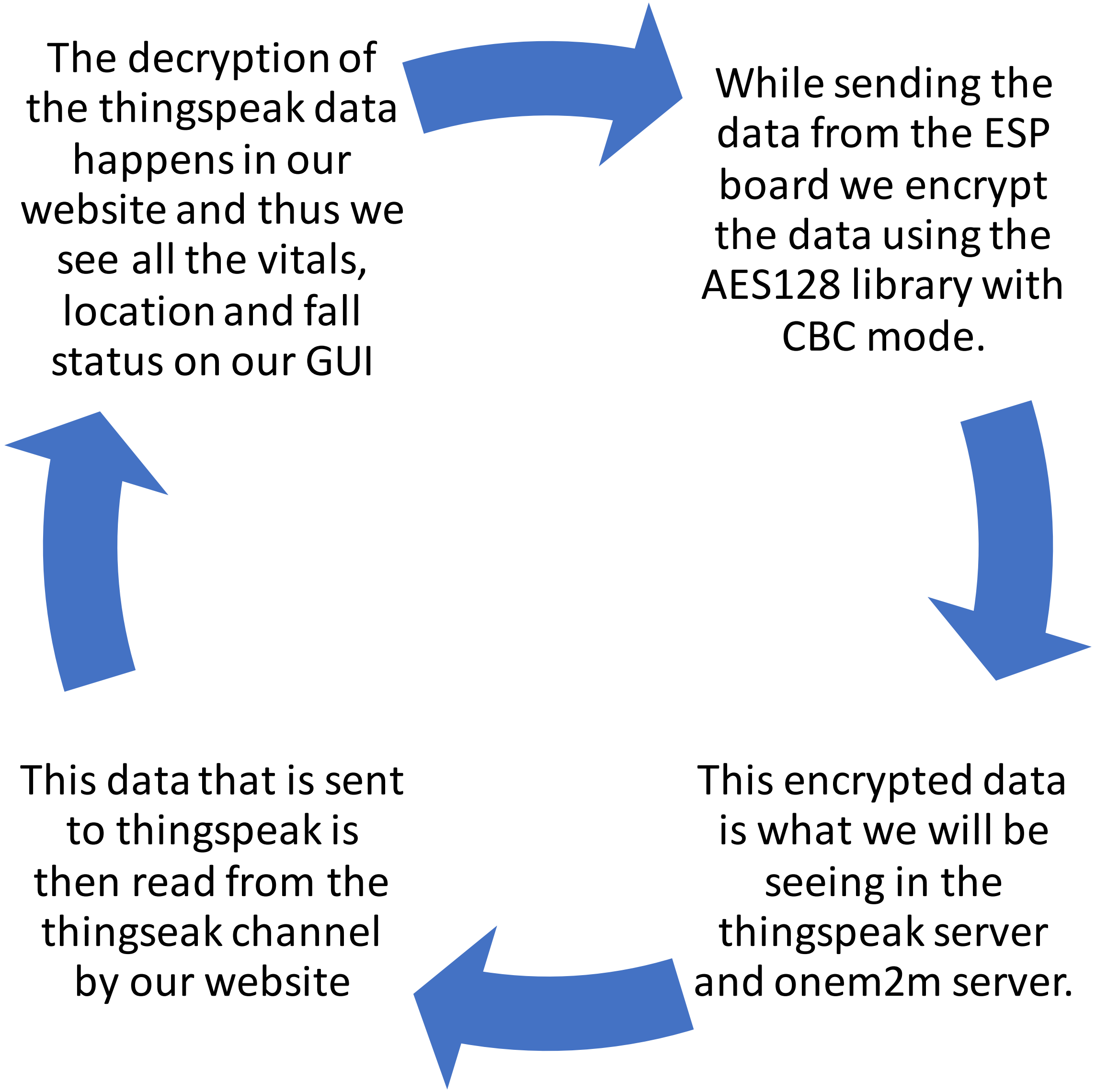
Since the code is done using python and the dependencies are readily available, we can run it straight on RPi after downloading the required dependencies

For ESP32 we need tensorflow lite to convert our tensorflow model to tensorflow lite model which we can readily run on ESP32

Convert the ML model to a C byte array using tinymlgen and store it in a .h file which we to store it in a read-only program memory on device

Model has been optimised by reducing no. of neuron, kernel size, stride to minimize complexity/computation cost so as to run on ESP32

Data Security



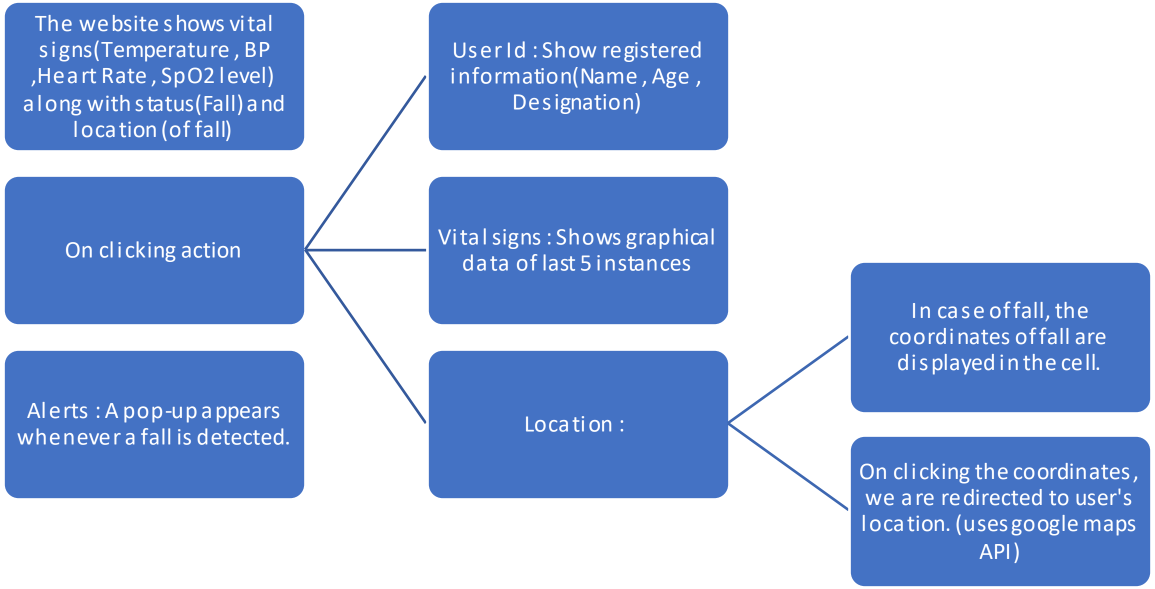
While sending the data from the ESP board we encrypt the data using the AES128 library with CBC mode.

This encrypted data is what we will be seeing in the thingspeak server and onem2m server.

This data that is sent to thingspeak is then read from the thingseak channel by our website

The decryption of the thingspeak data happens in our website and thus we see all the vitals, location and fall status on our GUI

GUI



The website shows vital signs(Temperature , BP ,Heart Rate , SpO2 level) along with status(Fall) and location (of fall)

On clicking action

Alerts : A pop-up appears whenever a fall is detected.

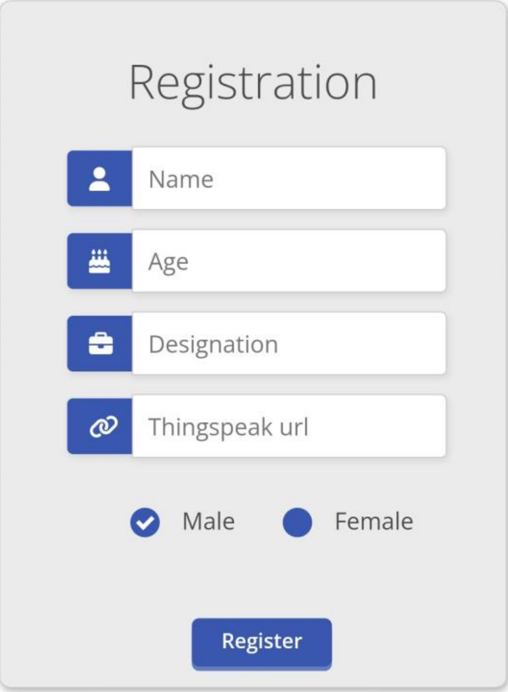
User Id : Show registered information(Name , Age , Designation)

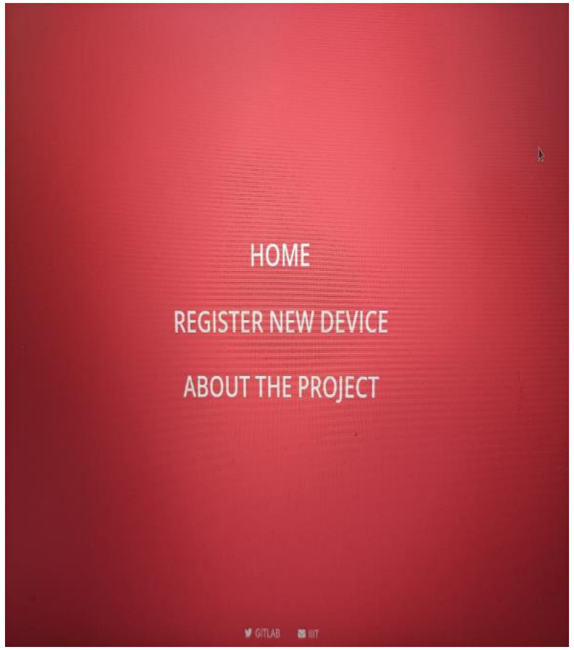
Vital signs : Shows graphical data of last 5 instances

Location :

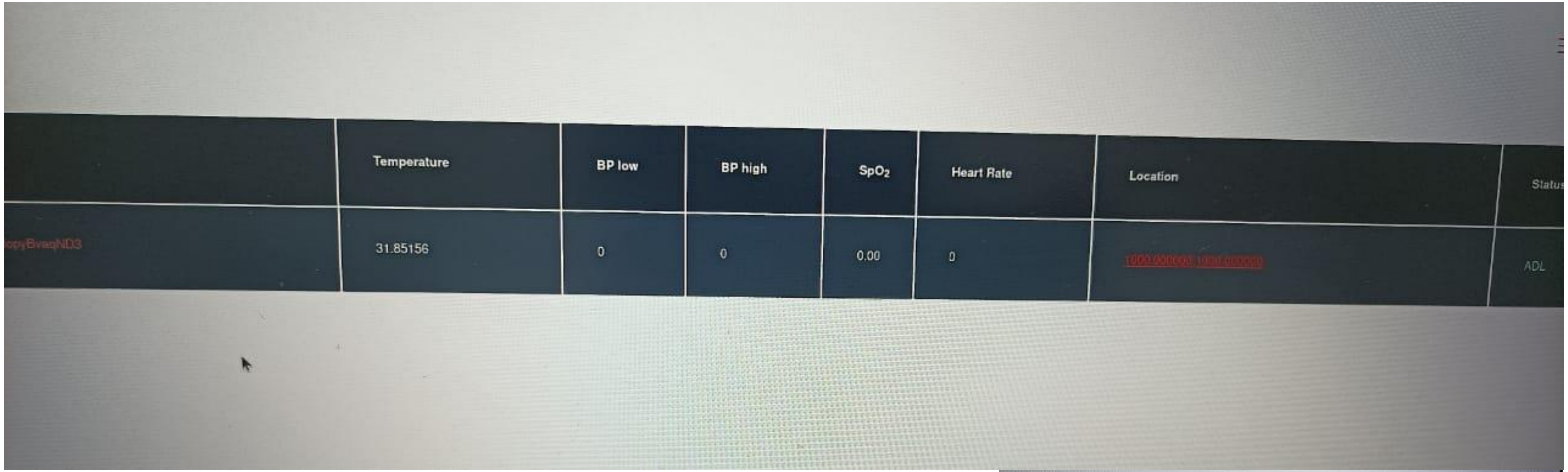
In case of fall, the coordinates of fall are displayed in the cell.

On clicking the coordinates , we are redirected to user's location. (uses google maps API)

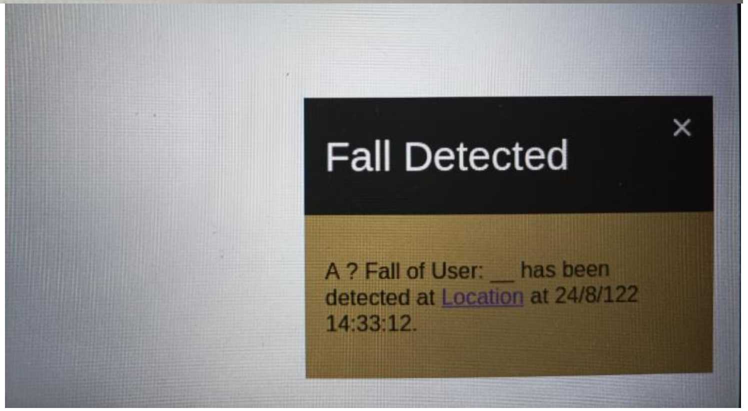




Data display



	Temperature	BP low	BP high	SpO_2	Heart Rate	Location	Statu
opyBvaqND3	31.85156	0	0	0.00	0	1000.000000,1000.000000	ADL



Fall Detected

A ? Fall of User: __ has been detected at Location at 24/8/122 14:33:12.

Analysis and Inferences

- From the fall status, type of fall, and vitals we can predict whether the user is unconscious or not
- By collecting the location and fall type we can also infer what all are the accident prone areas from where a person is likely to fall from a height and thus ensure more security measures are taken in those locations

ML Model GUI data

- The fall prediction CNN model outputs the probability values of it being a fall or an ADL
- From these probability values we predict if the user has fallen or not
- This prediction is then sent to thingspeak, and then shown on the GUI as the status of user