*AL- Azhar University*

Faculty of Engineering, Computers and Systems Department



**HUMAN ACTIVITY RECOGNITION USING**

**SMART PHONES**

Project submitted in partial fulfilment of the requirements the Degree of Bachelor of Science in Systems and Computers Engineering

Submitted By

Asmaa Fathy Soliman

Esraa Hassan Abdelkhaleq

Esraa Mohamed Radwan

Nour Salem Alaoady

Heba Ahmed Hindi

Supervised by

Dr. Khaled Ahmed Alshafey

July, 2021

**Acknowledgement**

In the beginning, praise be to God for his beautiful benevolence and for the perfection of his grace and generosity, and he is the one who by his grace good deeds are completed and with the blessing of his help works and good deeds are complemented and he is the owner of majesty and honor and prayers and peace be upon the best of creation, Muhammad and his family and companions. And after, praise be to God, I extend my thanks and appreciation to our professor, Dr. Khaled El Shafeay, who would not be satisfied with any words, without his perseverance and continuous support, this work would not have been completed.

After that, thanks go to all our teachers and our parents who we were taught by their hands in all stages of our studies and who made strenuous efforts to be as we are now. You have all our love, respect and appreciation.

I

# Abstract

Human activity recognition (HAR) is the problem of automatic identification of physical activity performed by humans. There are several techniques to measure motion characteristics during these physical activities, such as accelerometer sensors which are built in smart phones, and are characterized by Reduced Cost, Low Latency, Security and Privacy. With the use of accelerometer data that sample some measures of the activity, we fed them to ML models that were able to learn from experience and correctly classify activities to their corresponding classes. In this project we use convolutional neural networks (CNN) to classify whether a person is performing the action of jogging, walking, walking upstairs, walking downstairs, sitting or standing only on the basis of mobile phone sensor data.

We build CNN models and optimize its result using hyperparameter tuning, performance measures such a confusion matrix and 98% on learning set and this result is very promising to be tested on smartphones which is our production environment. Lastly, we deploy our CNN model in smart phones with the help of tensorflow packages that are compatible with CNN.

Our model confuses between walking down stairs and upstairs and this happen due to the accelerometer data itself .

# Table of Content

Contents

[Abstract II](#_Toc76939851)

[Table of Content III](#_Toc76939852)

[List of Figures IV](#_Toc76939853)

[Abbreviations V](#_Toc76939854)

[Chapter 1: Introduction 7](#_Toc76939855)

[1.1 Motivation 8](#_Toc76939856)

[1.2 Objective 9](#_Toc76939857)

[1.3 Problem Description 10](#_Toc76939858)

[1.4 System Design 11](#_Toc76939859)

[1.5 Implementation 13](#_Toc76939860)

[1.6 Chapter Summary 15](#_Toc76939861)

Chapter 2 :[Literature review 16](#_Toc76939862)

[2.1 Background 16](#_Toc76939863)

[2.2 Chapter Summary 17](#_Toc76939864)

[Chapter 3 System design 18](#_Toc76939865)

[3.1 Process 18](#_Toc76939866)

[3.2 parameter 23](#_Toc76939867)

[3.3 Chapter summary 24](#_Toc76939868)

[Chapter 4 Implementation 25](#_Toc76939869)

[4.1 Configuration Code 25](#_Toc76939870)

[4.1.1 Data Analysis (EDA) 27](#_Toc76939871)

[4.1.1.1 Balance this data 28](#_Toc76939872)

[4.1.1.2 Data Visualization 29](#_Toc76939873)

[4.1.1.3 Standardization of data 35](#_Toc76939874)

[4.1.1.4 Frame Preparation 35](#_Toc76939875)

[4.1.2 Build 2D CNN Model 38](#_Toc76939876)

[4.1.2.1 Train Model 40](#_Toc76939877)

[4.2 Chapter Summary 41](#_Toc76939878)

[Chapter 5 Results, Analysis, and Discussion 42](#_Toc76939879)

[5.1 Experimental Consideration 42](#_Toc76939880)

[5.2 Experimental Results 43](#_Toc76939881)

[5.3 Results Analysis and Discussion 43](#_Toc76939882)

III

[Chapter 6 : 6.1 Conclusion 45](#_Toc76939883)

[6.2 Future Works 45](#_Toc76939884)

6.3 References …………………………………………………………..46

# 

|  |  |  |
| --- | --- | --- |
| **Figure No.** | **Title** | **Page No.** |
| 1.1 | Overview about HAR using accelerometer sensor | 7 |
| 1.2 | The young woman discovered that she had kidney failure with her apple watch. | 8 |
| 1.3 | Tracking for elders and people who has disabilities to prevent them | 9 |
| 1.4 | Classifying , Training , Testing | 10 |
| 1.5 | Human activity recognition mechanism using a Smartphone | 11 |
| 1.6 | First five row in data set in Jupiter by df.head | 11 |
| 1.7 | Raw signal of sitting | 11 |
| 1.8 | Raw signal of jogging | 11 |
| 1.9 | Spectrograms in 3D for one action | 12 |
| 1.10 | Predicted table | 13 |
| 1.11 | Deep conventional neural network | 15 |
| 2.1 | Lierature review of HAR | 17 |
| 3.1 | First five row in the data set in Jupyter by df. Head | 19 |
| 3.2 | Acceleration signals at 3D for one action. | 21 |
| 3.3 | Spectral analysis by cycles | 22 |
| 3.4 | User interface for activity classifier when I am sitting | 24 |
| 4.1 | Activity Signals | 25 |
| 4.2 | Signals of upstairs action in 3D | 29 |
| 4.3 | Signals of jogging action in 3D | 30 |
| 4.4 | Signals of downstairs action in 3D | 30 |
| 4.5 | Signals of sitting action in 3D | 31 |
| 4.6 | Signals of standing action in 3D | 31 |
| 4.7 | Signals of walking action in 3D | 31 |
| 5.1 | Input, output of HAR | 42 |
| 5.2 | Predicted table | 43 |
| 5.3 | Model accuracy | 44 |
| 5.4 | Model Loss | 44 |

# Abbreviations

HAR Human Activity Recognition

ML Machine Learning

DL Deep Learning

CNN Conventional Neural Network

RBM Restricted Boltzmann machines

PCA Principal Component Analysis.

FFT Fast Fourier Transform

ACC Accelerometer

# Chapter 1

# Introduction

* **Overview**

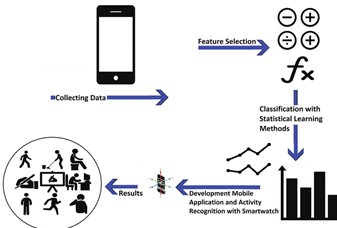
Human activity recognition is based on predicting what a person is doing depend on a trace of their movement using sensors, movements are often normal indoor activities such as standing, sitting, jumping, and going up and down stairs. Sensors are often located on the subject, such as a smartphone or vest, and often record accelerometer data in three dimensions (x, y, z).

Human Activity Recognition (HAR) aims to identify the actions carried out by a person given a set of bservations of him/herself and the surrounding environment. Recognition can be accomplished by exploiting the information retrieved from various sources such as environmental or body-worn sensors.

HAR includes two fundamental procedures: activity description and activity classification.

It mainly includes four steps: the collection of raw data, data preprocessing, model training and activity classification.

Deep learning methods have been demonstrated successfully on HAR problems given their ability to automatically learn higher-order features.



**Figure(1.1) :**Overview about HAR using accelerometer sensor

# 1.1 Motivation

In hospitals and with anyone who take care of older people or someone with disabilities where they need constant observation.

Classifying activity of a person is very useful to give reports to their supervised doctor and to know when and how they fall down which costs a number of deaths every year, this still in progress as an end to end system.

There are several use cases of human activity recognition, some examples as people who like to keep the track of their health and activities during the day and used in a more clinical context for example.

Disease diagnosis by monitoring the walking patterns of patients for early detection.

There is a great example that happened in 1999 on ABCaction news the young woman discovered that she had kidney failure by her apple watch.

Movement tracking for elders and people who has disabilities.

The Center For Disease Control And Prevention reports there are 29 million falls easily specially older people and it is leading to 31 billion dollar Medicare costs and 28000 deaths of American every year.



**(Figure 1.2):** The young woman discovered that she had kidney failure with her apple watch.



**(Figure 1.3**): Tracking for elders and people who has disabilities to prevent them.

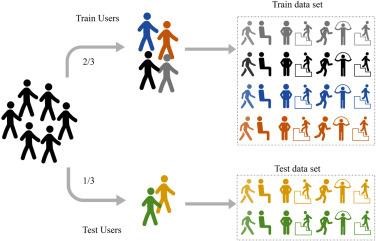
# 1.2 Objective

Main objective of our project is to recognize the human’s activities by analyzing the mobile phone’s sensor data, more specifically we have to make a model which can predict or accurately classifies whether a person is performing the action of jogging, walking, walking upstairs, walking downstairs, sitting or standing only on the basis of mobile phone sensor data.

Accelerometer is built in our smart phone to measure our position, movements and orientation and because of this sensor the improvements in daily life.

Using sensor data obtained from study performing six different activities (walking, walking upstairs, walking downstairs, sitting, standing and jogging), our objective is to build a model that accurately classifies which of these activities is being performed.

Building a simple android app as our case study to classify activity and used to track health and more clinical context (chart),for example if a company is doing a drug trial and wants to know if their drug makes study subjects and more or less active they can look at the activity classifier output and see if supjects are spending more time around or if they are mostly idle.



**Figure(1.4):** Classifying , Training , Testing

# 1.3 Problem Description

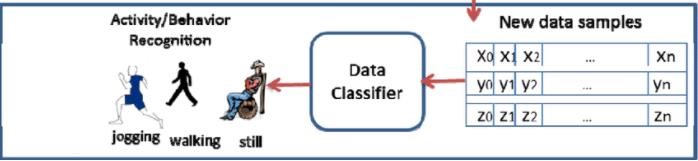
Human Activity Recognition (HAR) is the problem of identifying a physical activity carried out by an individual dependent on a trace of movement within a certain environment. Activities such as walking, jogging, sitting, standing, and walking up and down stairs are classified as regular physical movements and form our class of activity which is to be recognized.

To record movement, sensors such as triaxial accelerometers capture data while the activity is being performed, a triaxial accelerometer data detects acceleration or movement along the three axes orthogonal.

Data recorded is along three dimensions of the X, Y and Z axis at the specified frequency, for example, a frequency of 20 Hz would indicate that 20 data points are recorded each second of the action.

Activity recognition can be achieved by exploiting the information retrieved from this sensor and deep learning models have the capabilities to learn features of the higher order.

Advancement in such models makes it ability to learn and improve the performance of the predictive models and find deeper knowledge from human activities.



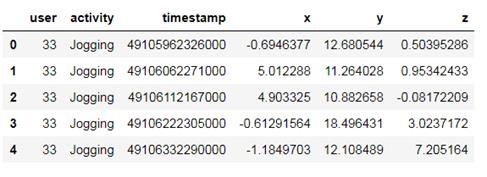
**Figure (1.5)** Human activity recognition mechanism using a Smartphone accelerometer

# 1.4 System Design

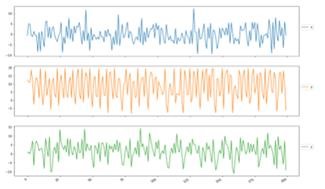
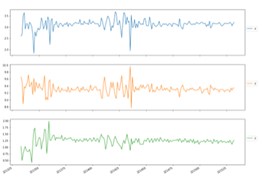
Data set is collected through controlled laboratory conditions the total number of examples is 1.098.207 and it contains six labels which are ( setting, walking, standing, jogging, downstairs, upstairs) the data used available at [WISDM: Wirless Sensor Data Mining](https://www.cis.fordham.edu/wisdm/dataset.php)

● Input to Model is time series data which means data changes over the time

(Unstructured data) so the best way to deal with it by using deep learning algorithms



**Figure (1.6**): First five row in data set in Jupiter by df.head



**Fig (1.7):** Raw signal of sitting **Fig (1.8):** Raw signal of jogging

The time series data are different in fig (7.) and fig (8.) it will be useful to make predictions.

Feature extraction is one of the key components of activity recognition.

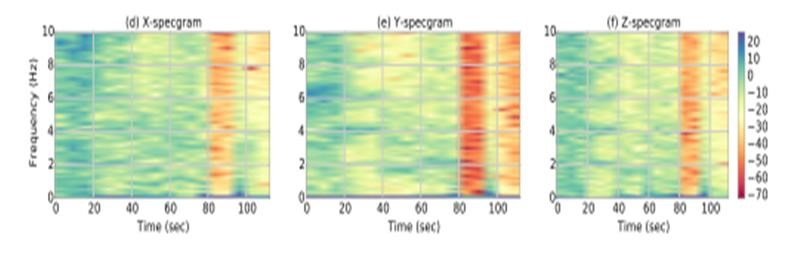
In this project we will focus on one of this features that is CNN: Conventional neural network.

In the classification phase, we first extract features from unseen raw data and then use the trained prediction model to predict an activity label.

Model design, we choose CNN because it is supported by tenserflowlite and would be compatible with smartphones and best suitable type of deep learning for recognition system is CNN

CNN-based model has three kinds of layers:

1. Input layers whose values are fixed by the input data
2. Hidden layers whose values are derived from previous layer
3. Output layer whose values are derived from the last hidden layer



**Fig (1.9):** Spectrograms in 3D for one action

# 1.5 Implementation

There are basic steps to have code that can predict and give good accuracy Firstly, we import data our data is collected through controlled laboratory conditions the total number of examples is 1.098.207 and it contains six labels which are ( setting, walking, standing, jogging, downstairs ,upstairs) the data used available at [WISDM: Wirless Sensor Data Mining .](https://www.cis.fordham.edu/wisdm/dataset.php)

Second step is cleaning the data that can be encoded, scaled, dropped if we not need it, our text file will be scaled.

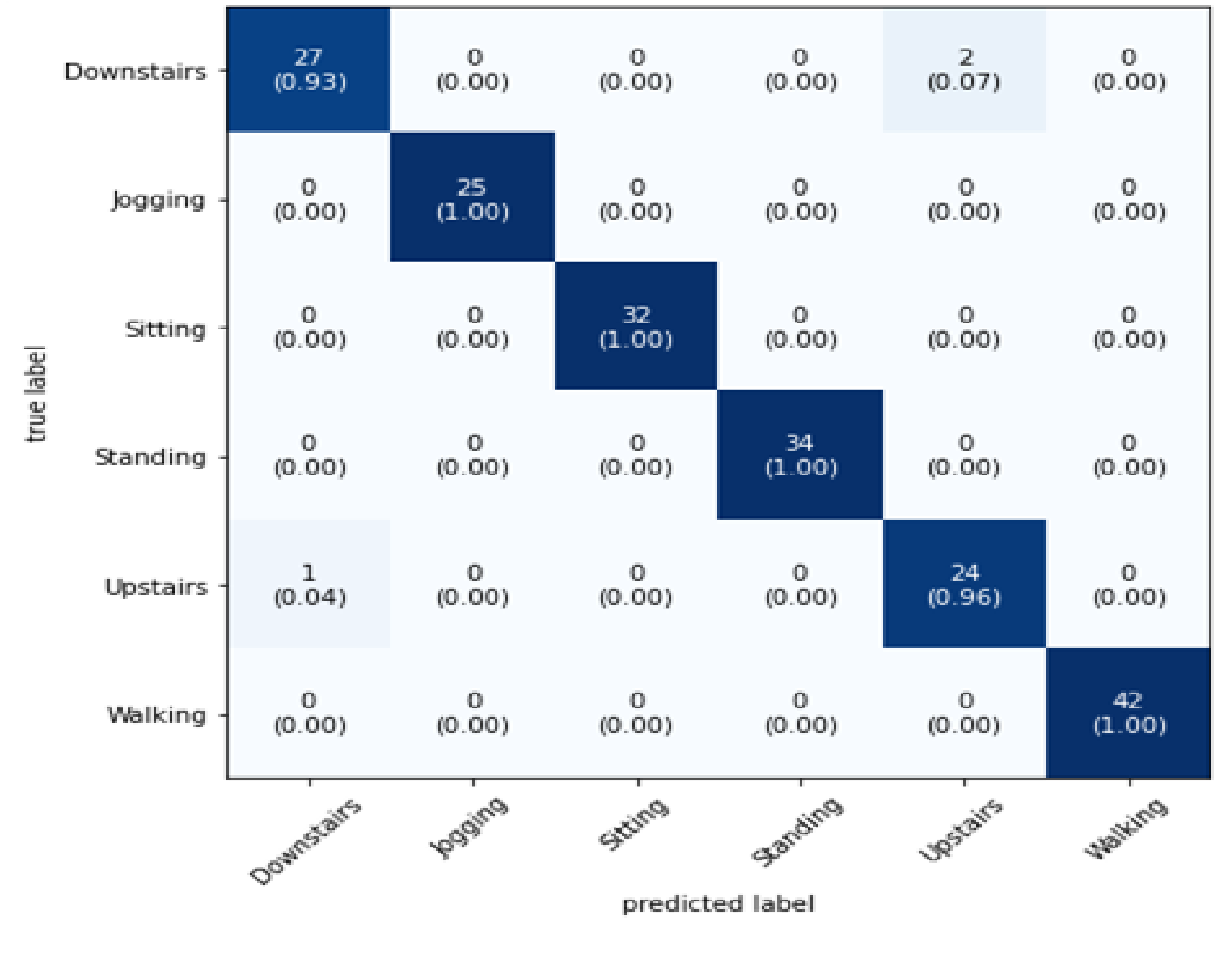
Third step is splitting the data into training &testing sets, we will learn data sets to can understand not memorize then we can test.

Fourth step is creating model, it means choosing algorithm that will be learned from data set, and our algorithm is CNN.

Fifth step is training the model, model must learn by different ratios value to achieve good fitting in prediction avoiding under and over fitting.

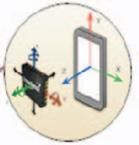
Sixth step is making predictions, in this step we can predict what is activity that is performed ( i.e. :human activity recognition )

Last step is evaluating and improving result (freezing and optimizing).



**Fig.(1.10):**Predicted table

##### Tools/Technology

 We use some SW tools to get the best result and ensure that result is good :

1. Smartphone with accelerometer
2. Google Colab
3. Jupyter(installing platform anaconda)
4. ImportantDeep Learning Libraries



Take place by the user

1. TensorFlow
2. Keras
3. Android studio

##### Take place by the user

We import libraries as:

* Numpay
* Panda

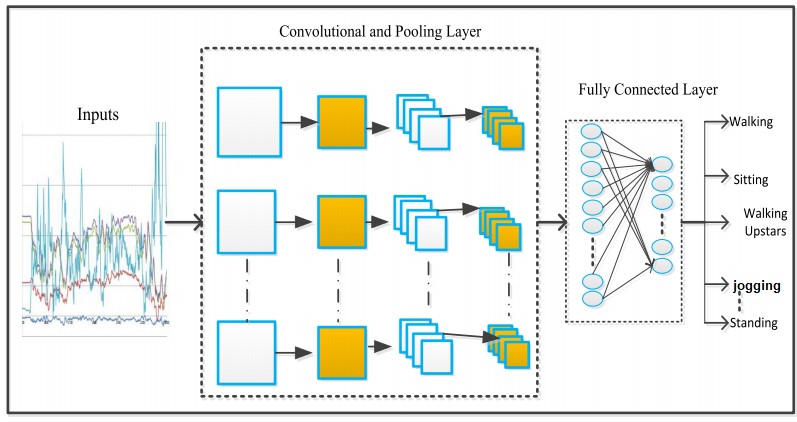
Take place by the user

Take place by the user

# 1.6 Chapter Summary

Human activity recognition systems are developed as part of a framework to enable continuous monitoring of human behaviours in the area of living, his /her case detection, elderly care, and surveillance in their environments. The extraction of relevant features is the most challenging part of the mobile sensor-based human activity recognition pipeline.

Feature extraction influences the algorithm performance and reduces computation time and complexity.



**Figure. (1.11**): Deep conventional neural network

**CHAPTER TWO**

**Literature review**

# 2.1 Background

* In 2004

Five accelerometers used

Tree classifier

Accuracy 84%

* In 2009

Used KNN classifier

Accuracy 95%

* In 2010

Smartphone tri acc used

Kernel disclimination

Accuracy 96%

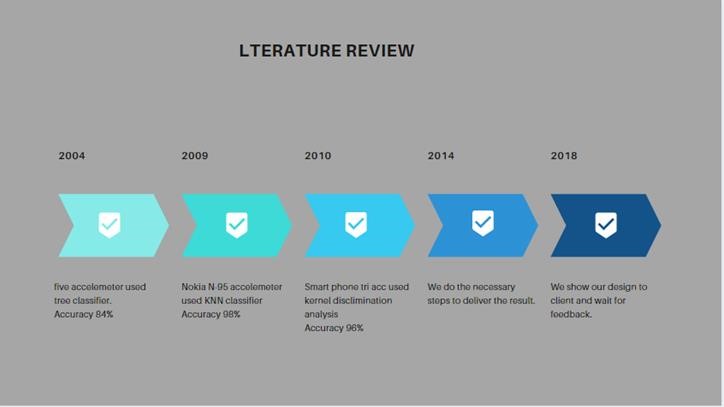
* In 2014

Start using the necessary stop to deliver the result

Accuracy =96.9%

* In 2018

Show the client the result and wait for feedback accuracy =83%.



**Figure (2.1**): Lierature review of HAR

# 2.2 Chapter Summary

Human Activity Recognition(HAR) is a challenging problem that tried to be implemented over the years using traditional ML technique which needs professional experts to provide feature extraction from time-series data but with the new DL approaches the model performance is improved and there will be more advance in HAR systems yet to come.

# Chapter 3

# System design

# 3.1 Process

Designing a machine learning system is an iterative process, mainly there are four components of the process: problem statement data pipeline, modeling (selecting, training, optimization), and serving (testing, deployment)

The output from one step might be used to update the previous steps as:

* After examining the available data, you realize it's impossible to get the data needed to solve the problem you previously defined, so you have to frame the problem differently.
* After training, you realize that you need more data or need to re-label your data.

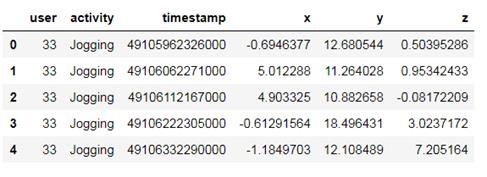
Problem statement:

It is a multi-class classification problem of human activities. Our main goal is to take input data of accelerometer sensor data represents by x, and z axes and our target labels six different activities (walking, walking upstairs, walking downstairs, sitting, standing, and jogging)and they what we want the ML model to learn and be able to predict on unseen data when being deploying on smartphones

Data pipeline:

Data availability and collection

Data set is collected through controlled laboratory conditions the total number of examples is 1.098.207 and it contains six labels which are ( sitting, walking, standing, jogging, downstairs, upstairs) the data used available at [WISDM: Wireless Sensor Data Mining](https://www.cis.fordham.edu/wisdm/dataset.php)



**Figure(3.1)** First five row in the data set in Jupyter by df. head

Data preprocessing & Visualization:

* How do you process the raw data into a form useful for your models?

We reshape our dataset to be suitable to the model which is the best fit for the data .

* Does it need normalization?

Yes to ensure our model doesn’t bias towards the highest scale.

* Privacy: What privacy concerns do users have about their data?

What anonymizing methods do you want to use on their data?

Can you store users' data back to your servers or can only access their data on their devices?

In our project, we state that privacy is a must to protect user data and it made possible with the TensorFlow packages now the model can predict in the device without the need to store the user’s data on the cloud

Modeling:

Input to Model is time-series data which means data changes over time (Unstructured data) so the best way to deal with it by using deep learning algorithms, therefore

* Model Selection

We choose CNN because it is supported by Tenserflowlite and would be compatible with smartphones and best suitable type of deep learning for recognition system is CNN

* Model Architecture

The CNN-based model has three kinds of layers:

* 1. An input layer whose values are fixed by the input data
  2. Hidden layers whose values are derived from previous layers
  3. Output layers whose values are derived from the last hidden layer

Feature extraction is one of the key components of activity recognition.

In this project, we will focus on one of these features that is CNN

* + CNN: Conventional neural network
  + RBM : Restricted Boltzmann machines
  + PCA: Principal component analysis.
  + FFT: Fast Fourier Transform.
  + Statistical: science of probability.

These features are then used to train a classification model.

In the classification phase, we first extract features from unseen raw data and then use the trained prediction model to predict an activity label.

Conventional Neural Network (CNN): it is more important in scientific ML it distributes probabilities in its inputs (visible unit), contains weights and hidden (output) layers.

It can be used to model complex relationships between inputs and outputs or to find patterns in data.

Conventional Neural networks are used for solving many business problems such as customer research, data validation, and risk management.

Conventional Neural Network provides an effective tool for extracting high-level features hierarchies from high-dimensional data which is useful in classification and regression tasks (prediction).

 The computational complexity or simply complexity of an algorithm :

Is the number of resources required to run

Trying to have the best or average case avoiding the worst-case O(n!)

After applying the spectrogram on𝑆𝑥𝑡, 𝑆𝑦t, and 𝑆𝑍𝑡 the length of the spectral signal is L = 3( N /2 + 1) **Eq**(3.1)

While the time domain signal length is 3N, N is the number of digital points.

This reduces the computational complexity of any classification method due to the lower data.

So, the spectrogram signal of the triaxial accelerometer is denoted as

**[21]**

Xt ∈𝑅𝐿, where L = 3( N /2 + 1) is the concatenated spectrogram signals from the triaxial input data.

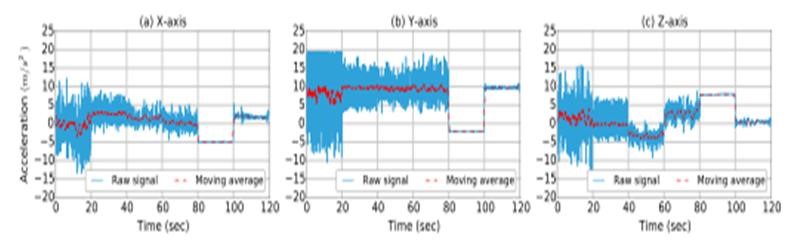
L: is the length of the spectral signal that taken by action of human N/2 +1: is amplitude values, spectral components N is the number of digital points.

Time-domain is n

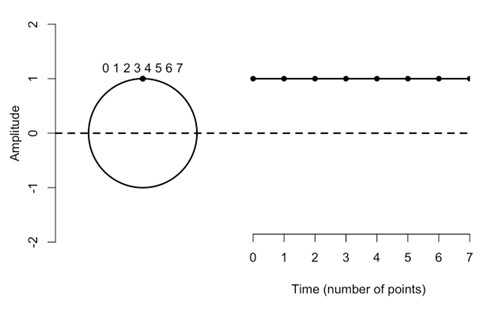
F: is the number of cycles

We can compute L for 10000 HZ and eight points for each cycle,N =8 P0INTS

L=3(8/2+1) =15 0 to 7 cycles by runiff(N,+Amp,-Amp)



**Fig(3.2)** Acceleration signals at 3D for one action.



**Figure(3.3):** Spectral analysis by cycles

Algorithm of CNN for HAR

Algorithm 1: Convolutional Neural Network for Activity



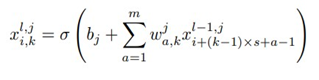
Recognition

Input: Labeled dataset D labeled = {((Xi, Yi, Zi), ai)}

Output: Activiy Recognation For one action from the six actions

Algorithm:

Repeat

 For each Accelerometer data from tri-axises, (x, y, z) do

Use Eq (3.2)

to conduct convolution operation with the input data



Use Eq(3.3) to conduct maxpooling with

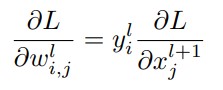
the output of convolution

End

Use fully-connected layer to integrate the pooling result of tri-axises

{((Xi, Yi, Zi), ai)} data

Use soft-max to do classification and update the weight of each edge in the network

Use  Eq (3.4) to conduct backward propagation

Until *Wi* convergences;

Use the trained network to predict the labels



i,j are features ,L is layer , Xi is output of convention layer ,ϭ is variance , W is width of layer ,partial deferential for gradient descent by measuring slope ,max function to reduce the size of the convolutional output Training:

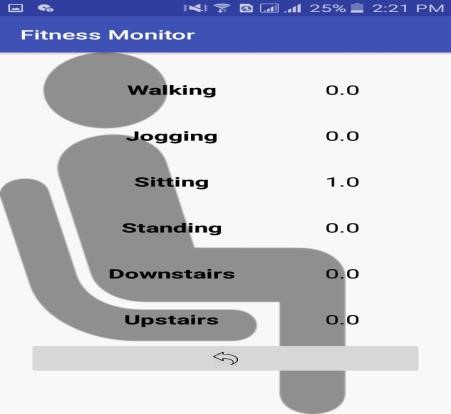
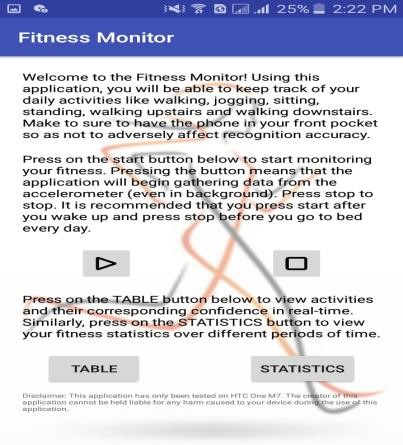
Should know what type of problems might arise and be able to address them such as overfitting, underfitting,

Debugging and optimization:

If the model performed poorly then you should return to optimize the model using hyper parameter tuning

Serving:

Testing the model in production after deploying it and consider the scale of the model and each constraints of every platform. Our plateform is android application which needs to integerate a scalabale ML model and redefining model input and output layers and use this android app to track health and in more clinical contexts. For example, if a company is doing a drug trial and wants to know if their drug makes study subjects more or less active



# 3.2 parameter

* frame preparation

Frequency = 20HZ

Time steps = 80

Step= 40

Features = 3 **Figure3.4**: User interface for activity classifier when I am sitting

* model training

Learning rate = 0.0001

Validation split = 0.1

Epochs = 10

# 3.3 Chapter summary

Designing a machine learning model it is a systematic way of implementation where it still considers a state of art but it always better to setup project first and analyze the problem to reduce the time and cost of redo in the step of the process and to keep in mind the priority of doing ML model to be suitable for production and end-user.

# 

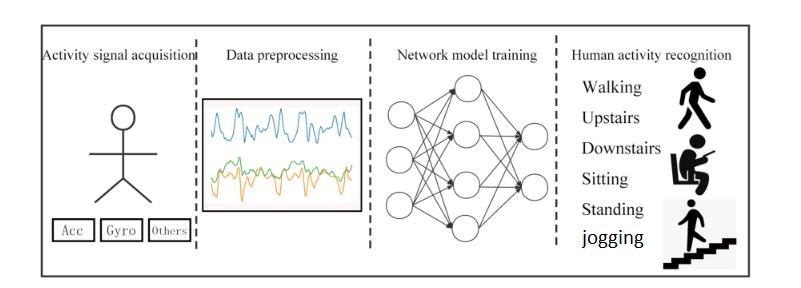
# CHAPTER 4

# Implementation

# 4.1 Configuration Code

In this project, we are going to use accelerometer data to train the model so that it can predict human activity.

We are going to use 2D Convolutional Neural Networks to build the model.



**Fig (4.1):** Activity Signal

**Dataset**

We are going to use WISDM Dataset in our project. This dataset contains data collected through controlled, laboratory conditions. The total number of examples is 1,098,207. The dataset contains six different labels (Downstairs, Jogging, Sitting, Standing, Upstairs, And Walking) Import nessacerry libraries to build CNN model Using Keras, TensorFlow

* pandas is used to read the dataset.
* NumPy is used to perform basic array operations.
* pyplot from matplotlib is used to visualize the results.
* train\_test\_split from sklearn is used to split the data into training and testing datasets.
* LabelEncoder from sklearn is used to encode target labels with values between 0 and the number of classes-1.
* StandardScaler from sklearn is used to bring all the data on the same scale.

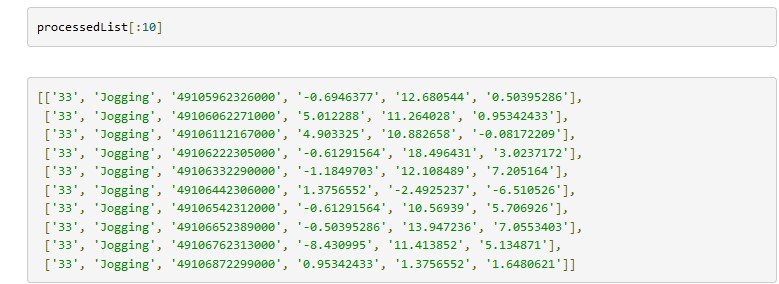
**Load and preparing the dataset**

If we try to read this data directly using pd.read\_csv() we will get an error because this data is not preprocessed properly. So we will have to read this data into a native python file and then pre-process it.

Using open() we will first open the file. Then we will read all the lines of the file into the read variable. Now we will consider all the lines one by one using a for loop. For each line, the following operations will be performed-

* line = line.split (',') splits the line wherever there is a comma and returns an array of separated elements.
* last = line[5].split (';') [0] removes the semicolon after the last element in the array.
* last = last.strip() removes any extra space.
* Then if the last is not empty we copy all the elements into temp.
* Now that the line is ready we append it to processedList

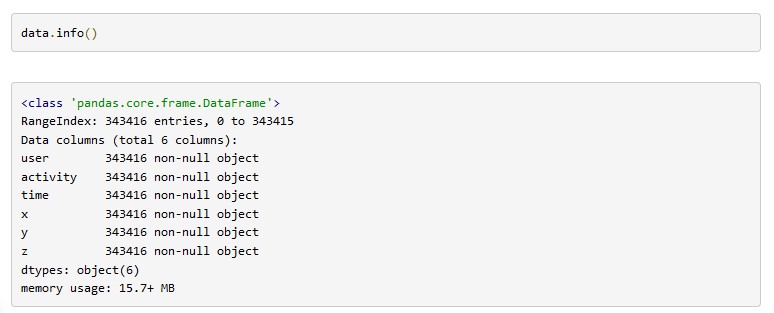
 try and except is used for error handling. In this process, if we get an error, the number of the line which is throwing that error is displayed.

 Now we have the processed list. It is a list of lists. Each inner list has the user ID, activity, timestamp and then the x, y, z data.

 Now we will create a DataFrame with the processed data and proper column names. data.head() will display the first 5 rows of data.

## 4.1.1 Data Analysis (EDA)

This data has 343416 rows and 6 columns.



This will give more information about data. It says that all the values are string objects.

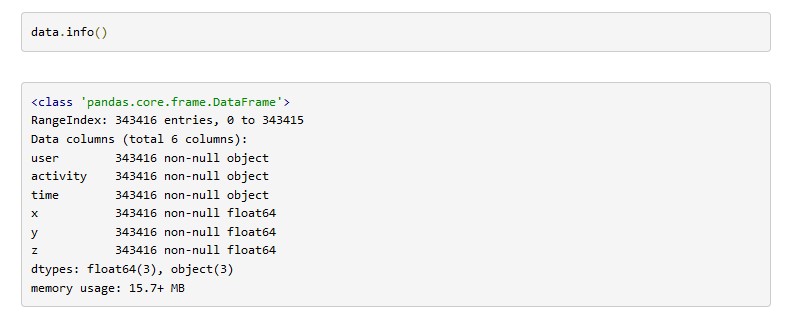
Now we will see if any null values are present in the dataset using isnull().

### 4.1.1.1 Balance this data

From the data distribution shown above, we can observe that the data is unbalanced. Standing has very few examples compared to Walking and Jogging'. If we use this data directly it will overfit and will be skewed towards Walking and Jogging'.

As we saw earlier the data is in string data type, here we have converted the x, y, z values into floating values using astype('float').

We can see that the data type of x, y, z has changed.



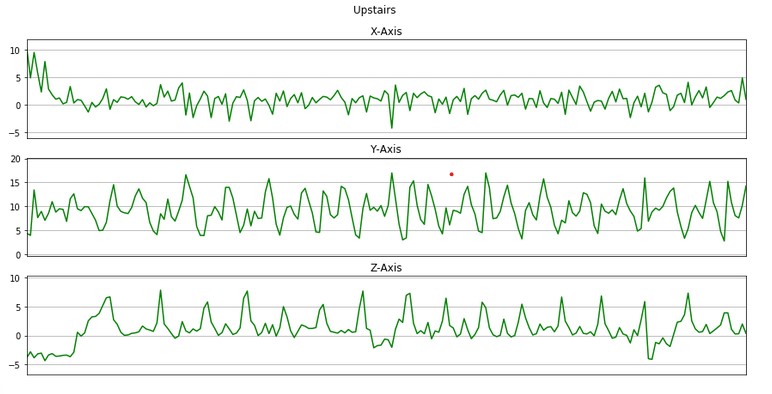
Now we will plot x, y, z for few seconds. The sampling rate of this data is 20Hz. So we have set a variable Fs=20. activities is a list of all the unique activities.



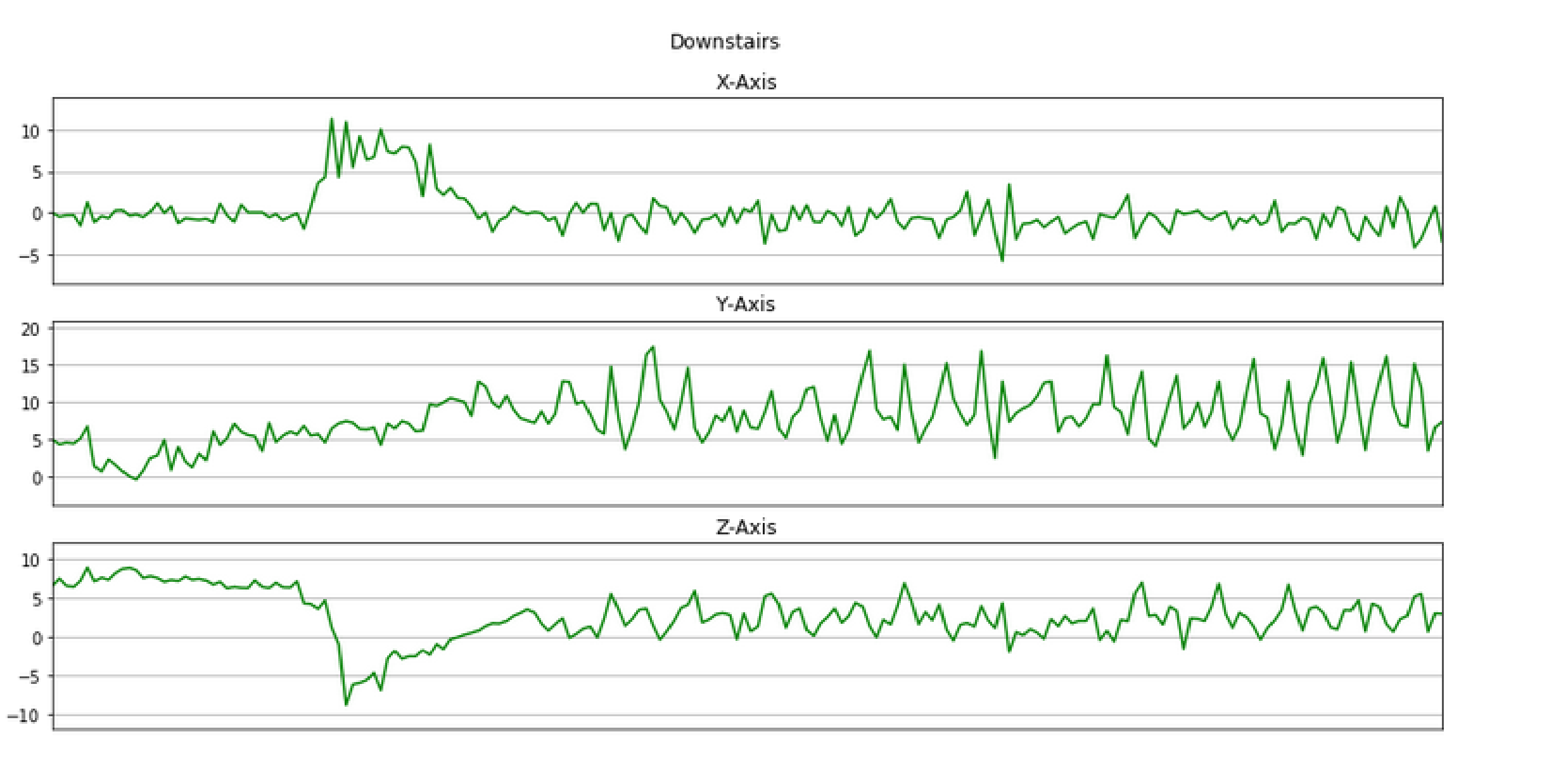
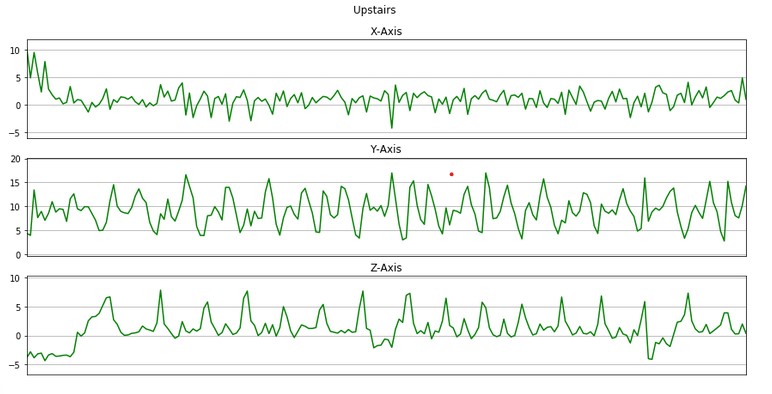
### 4.1.1.2 Data Visualization

Now we will plot x, y, z for each activity for 10 seconds.

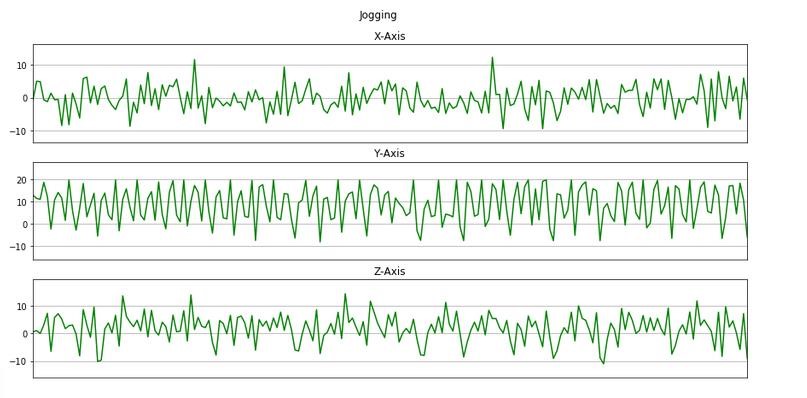




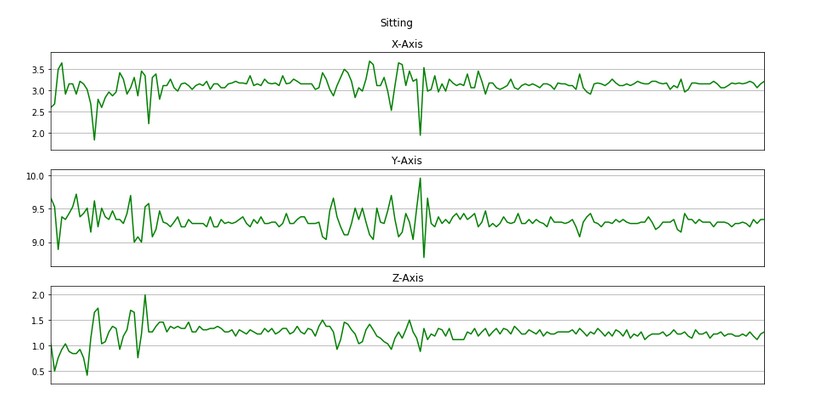
**Fig4.2**:(Signals of upstairs action in3D)

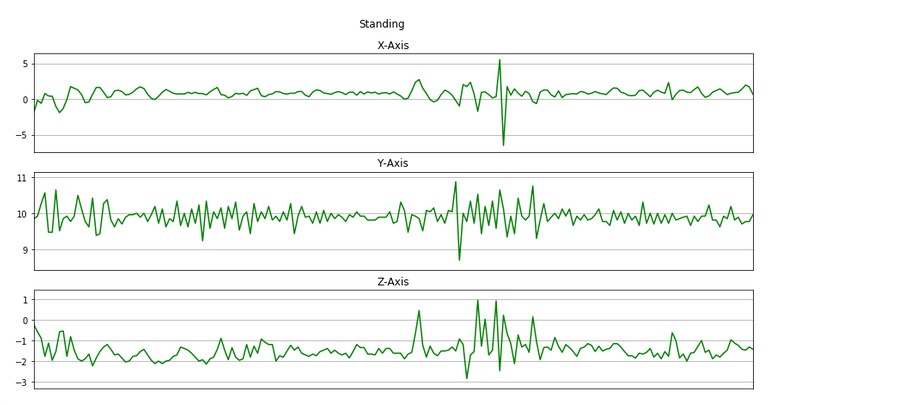


**Fig4.3**:(Signals of downstairs action in3D)

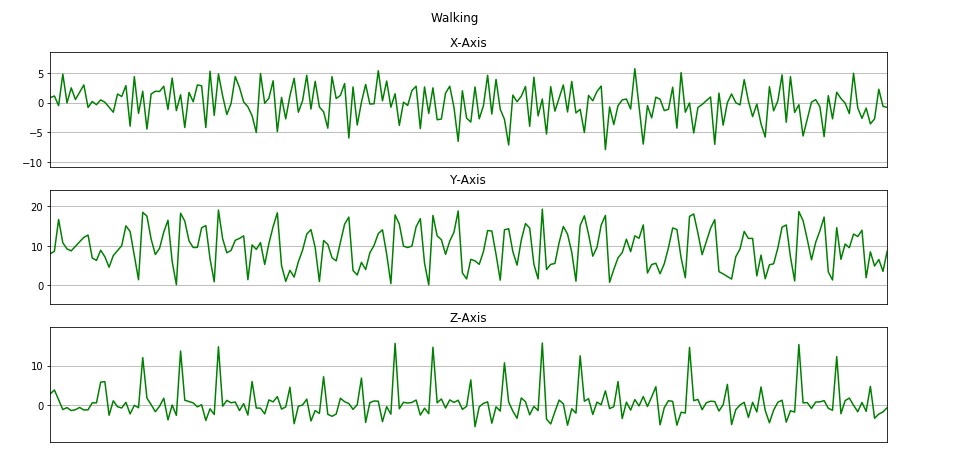


**Fig4.4** :( Signals of jogging action in3D)

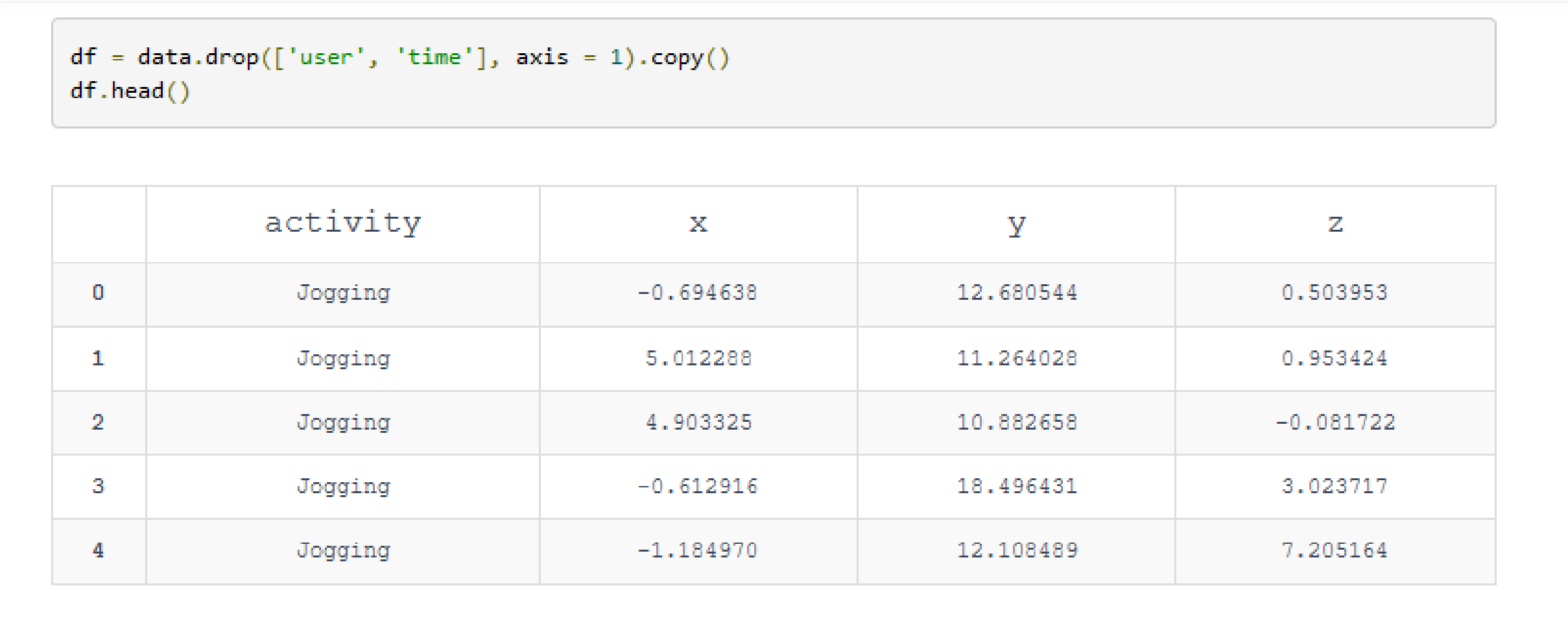


 **Fig4.5** :( Signals of sitting action in3D)

**Fig4.6** :( Signals of standing action in3D)



**Fig4.7** :( Signals of walking action in3D)

 Here we will remove the columns user and time from the dataset by using drop().

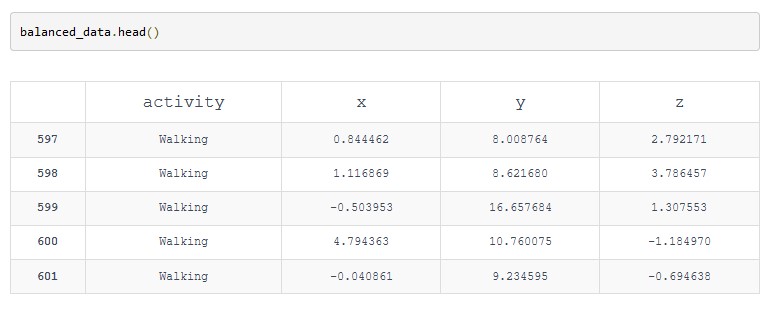
To see the distribution of data we will see the count of each unique activity using value\_counts().

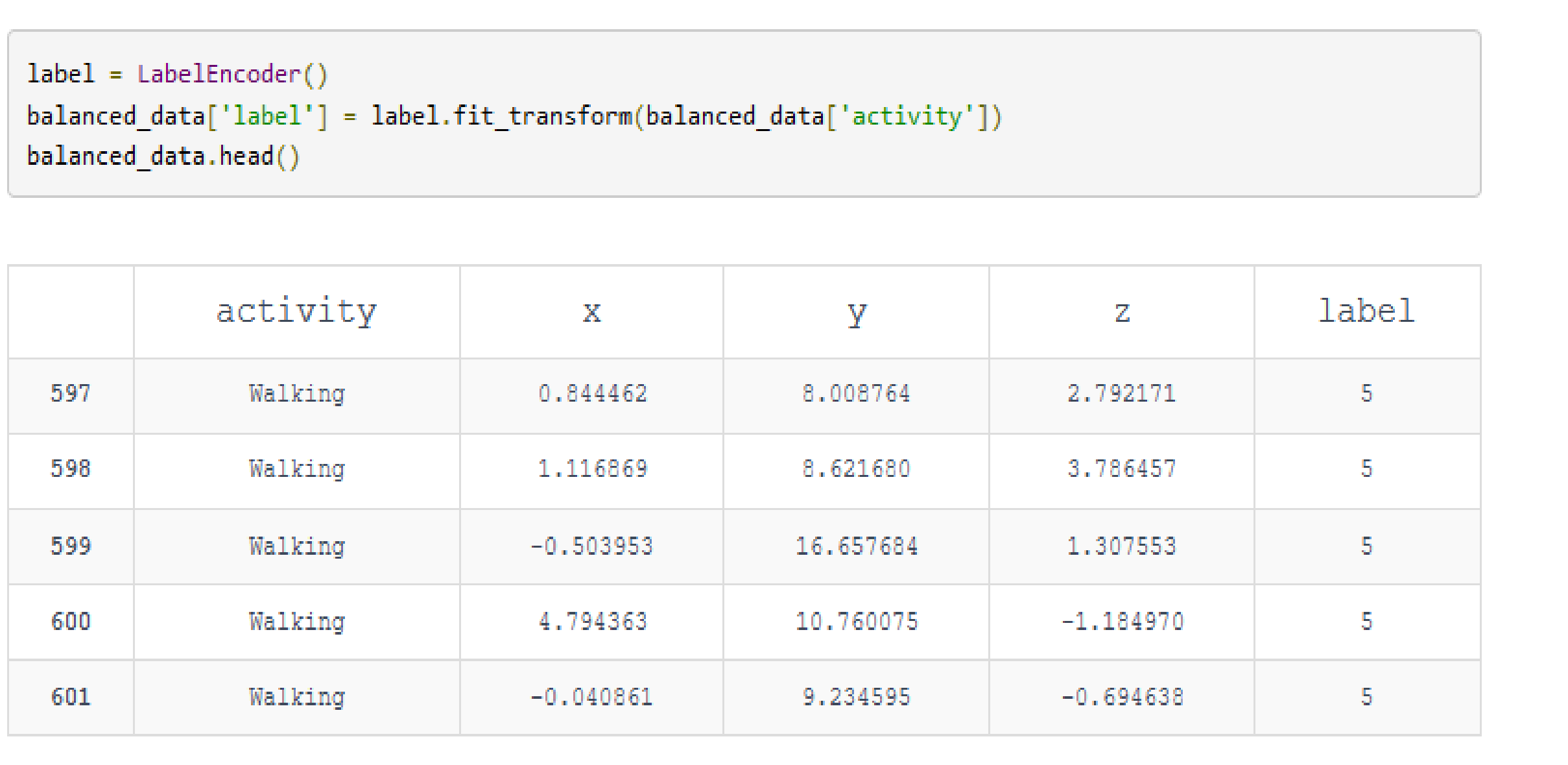
As this data is highly imbalanced we will take only the first 3555 lines for each activity into separate lists. Then we will create a dataframe balanced\_data using pd.DataFrame() and append all the lists to balanced\_data. The final shape of balanced\_data is 21330 rows and 4 columns.



Now the data is balanced. We can see this by calling value\_counts() on the activity column of balanced\_data.





As we can see above, the values in activity are of data type string. We will convert them into numeric values using LabelEncoder from sklearn which we have already imported. fit\_tranform fits label encoder and returns encoded labels. We will add a new column in the dataset with the name label which will have the encoded values.

 We can use .classes\_ attribute to recover the mapping of classes.

### 

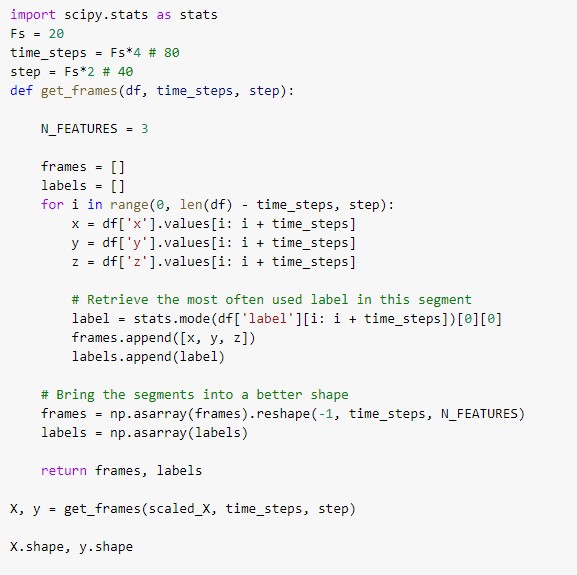
### https://lh3.googleusercontent.com/5QNS8WweX_mD7OJ_H2pZHG_RjJTIstFgKKVS04AZotsqgq9dcDkoZGMJtH-6jFvGfYbfS62NkSjA3GL4tqKbgk3H4_TYxbn0B5IBo-lx58AbeQyyLBX5gUgezvX-kHurMSIRHjJe4.1.1.3 Standardization of data

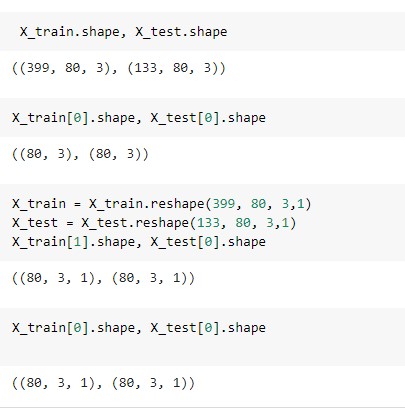
Here we are reading the feature space into X and the label into y.

Now we will bring all the values in X in the same range using StandardScaler() from sklearn which we have already imported. scaled\_X contains the scaled values of x, y, z and the labels.

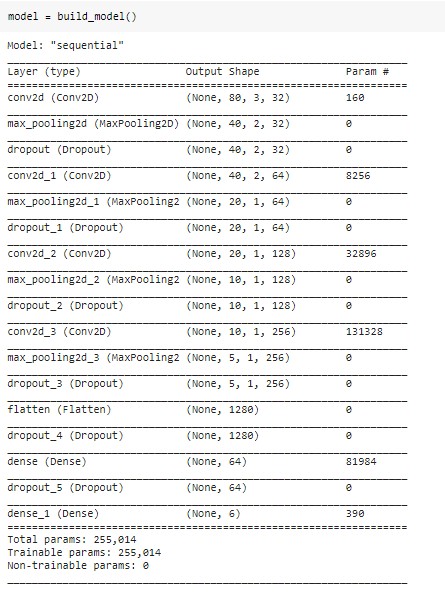
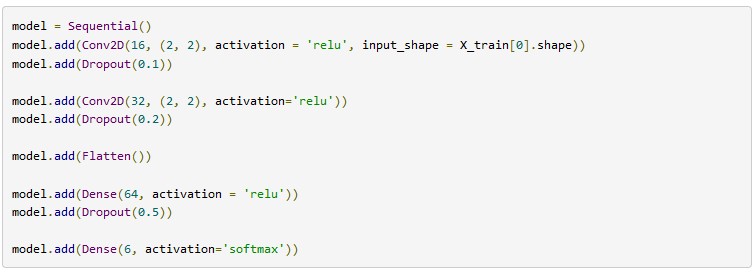
### https://lh5.googleusercontent.com/-qm5KlkDHQwhx2SWkn9Zk3E09yyWzQR8Ce6t6IblYnVPjLG31T15OCbKg2CHE4gT2XWsWgXrcrVvEkqUZr7G60h0UrtUwhc2r00JAMj0TpA026jfnDCimBhm87kQ3G0VmWxNbioc4.1.1.4 Frame Preparation

We are going to divide the data into frames of 4 seconds. To do this we will import scipy.stats.

We will multiply the frequency by 4 seconds. Hence we will consider 80 observations at a time.step will be 40 which means there will be some overlapping.  https://lh3.googleusercontent.com/3gdiIJ0TIAjsrSduHe5Fm65na3pUK3NdxvtCrH8au5VhNPUp1BRIujIaeONdXf8iiUrK_MsZ25CoafMlARXfpE-_TUmuMAR6e967YLPrlTvD7B3P1T4prlRiYnV2jU26puyCpYmj

Splitting data and reshape it into 3\_dimentional to suit CNN.

## 4.1.2 Build 2D CNN Model





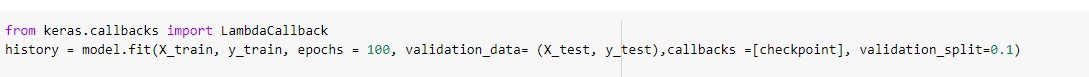
Model Optimization

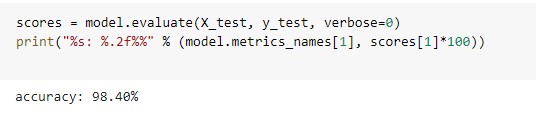
Compile Model

Here we are compiling the model and fitting it to the training data. We will use 10 epochs to train the model. An epoch is an iteration over the entire data provided. validation\_data is the data on which to evaluate the loss and any model metrics at the end of each epoch. The model will not be trained on this data. As metrics = ['accuracy'] the model will be evaluated based on the accuracy.

https://lh6.googleusercontent.com/5QtbtpmZGUY5ik19n7yORqddlS3j2gGYPtSn3JB0jAaAcjqeYbTB0l-7eUcwlzVFYUlLXfbjc3ddlqjAe8-jnVsj3kDbU0_mjsk6KjwIZ4jFLXgRtnp_efXxo5a3_4ejCazcL5m6

### 4.1.2.1 Train Model





Model evaluation

Freezing the CNN model and saving it as a .pb formate file



# 4.2 Chapter Summary

We followed the pipeline for the machine learning process and implemented data analysis, Exploratory data analysis(EDA), Data visualization, data preprocessing, normalize dataset to prevent overfitting and bias towards the largest scale, split data set into test and train and kept in mind apply validation test to help us measure the model performance and make sure if it generalizes well on the test set, build Model and optimize it to give us the nearest desired accuracy and lastly freeze and save the model to put it into production phase which in our case smartphone.

# Chapter 5

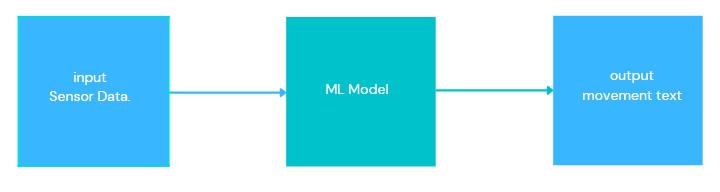
# Results, Analysis, and Discussion

# 5.1 Experimental Consideration

We keep in mind various considerations when it comes to building an ML model to classify our target label which are the six activities performed by any human.

Its input & output data:

The model takes a data set collected in laboratory conditions that have a value of accelerometer sensor and it’s collected from a smartphone’s accelerometer this data set is a time-series data change over time and it is an unstructured type, therefore the best way to deal with it using deep learning algorithms.



**Figure5.1** :( input, output of HAR)

* ML Model

There are several techniques used for HAR using various ML models and most of them depend on good feature extraction methods. DL uses neural networks which mimic the human brain to drive meaning from the given data, from the deep learning algorithm we choose CNN which will learn complex features automatically from the raw accelerometer signal to differentiate between activities.

* CNN Model consideration

We have to prepare the dataset in a format required for the CNN model

First, define fixed sized segments (window size)

Second, CNN accepts only 3-dimensional data so we must reshape data

# 5.2 Experimental Results

The accuracy of the ML model is **97**%

# 5.3 Results Analysis and Discussion

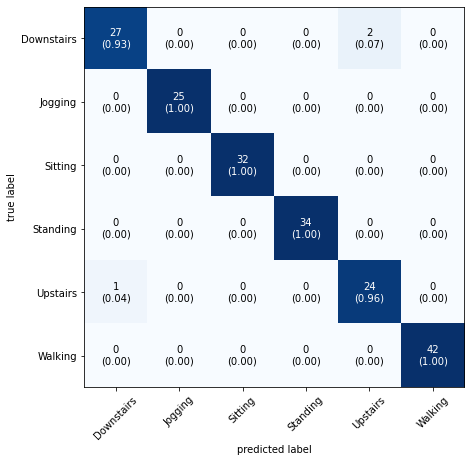
Analysis and discuss the performance of the ML model

* Using confusion matrix

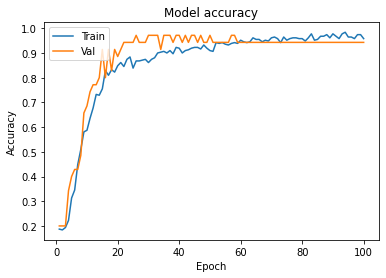
A confusion matrix is a table that is often used to describe the performance of a classification model (or “classifier”) on a set of test data for which the true values are known.

As we can see we are getting 100% accuracy for Sitting and Standing and jogging and walking.

The confusion matrix also tells us that our model is getting confused between Upstairs and Downstairs. We have got a decent accuracy for this data.

If you want a further increase in the accuracy you can play and change many things but it might lead our classifier to overfit so it is a goodfit so far.

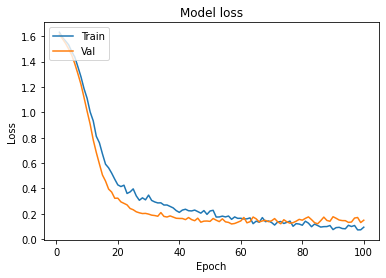
**Fig 5.2:** (Predicted table)



**Fig 5.3** Model accuracy

* Using learning curves

Learning curves are plots which used to know if our model generalizes well or not and it shows changes in learning performance over time in terms of experience.



**Fig (5.4)** Model Loss

Learning curves plot the training and validation loss of a sample of training examples by incrementally adding new training examples. Learning curves help us in identifying whether adding additional training examples would improve the validation score (score on unseen data). If a model is overfitting, then adding additional training examples might improve the model performance on unseen data. Similarly, if a model is underfitting, then adding training examples doesn’t help.

Our model is a good fit according to the learning curves accuracy and loss and it is ready to generalize well in unseen data.

# Chapter 6

# Conclusion

#### 6.1 Limitation

* The ML model doesn’t classify all the human activity in the medical context such as heart rate or blood sugar rate
* The ML model confuses between upstairs and downstairs and this due to the accelerometer dataset itself and they very similar in data visualization plots
* The TensorFlow lite package used for integrating ML model in smartphones is still unstable due to one device cutting edge technology but in its next update will perform well for deploy any size of ML model
* The data set doesn’t contain a falling activity and its a useful one in many contexts but its still in the research phases and it is hard for the ML model to be able to generalize well due to different weights

# 6.2 Future Works

Adding more data to the set such as heart rate and blood sugar, oxygen consumption.

Uploading the result data and make it available to be used in research.

Recognition systems which could predict actions before they take place by the user could be development in certain applications

Giving emergency voice if person in dangerous action and sending report location for his doctor

**6.3 References**

1. A. Khan, Y.-K. Lee, S. Lee, and T.-S. Kim, “*A triaxial accelerometer-based physical* *activity recognition via augmented-signal features and a hierarchical recognizer*,” in IEEE Trans. on Inf. Technol. in Biomedicine, vol. 14, no.

5, pp. 1166–1172, 2010

1. O. D. Lara, A. J. Perez, M. A. Labrador, and J. D. Posada, “Centinela: A human activity recognition system based on acceleration and vital sign data,” *Journal on Pervasive and Mobile Computing, 2011.*
2. O. D. Lara and M. A. Labrador, “A mobile platform for real-time human activity recognition,” in *IEEE Conference on Consumer Communications and Networks*, 2012.
3. J. Wang, Y. Chen, S. Hao, X. Peng, L. Hu, “Deep learning for sensor-based activity recognition: A survey,” *in Pattern Recognition Letters*, vol. 119, pp. 3-11, Mar 2019
4. J. R. Kwapisz, G. M. Weiss, and S. A. Moore. Activity recognition using cell phone accelerometers. *ACM SigKDD Explorations Newsletter*, 12(2):74–82, 2011.
5. S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(1):221– 231, 2013.
6. G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. Science, 313(5786):504–507, 2006.
7. D. Figo, P. C. Diniz, D. R. Ferreira, and J. M. Cardoso. Preprocessing techniques for context recognition from accelerometer data. *Personal and Ubiquitous Computing*, 14(7):645–662, 2010.
8. O. Abdel-Hamid, A.-r. Mohamed, H. Jiang, and G. Penn. Applying convolutional neural networks concepts to hybrid nn-hmm model for speech recognition*. In Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pages 4277–4280. IEEE, 2012
9. Tomas Brezmes and Juan-Luis Gorricho. Activity recognition from accelerometer data on a mobile phone. In Proc. *of ANN 2009*.
10. O. Abdel-Hamid, A.-r. Mohamed, H. Jiang, and G. Penn. Applying convolutional neural networks concepts to hybrid nn-hmm model for speech recognition. In Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on, pages 4277–4280. IEEE, 2012.
11. M. Fahriddin, M. G. Song, J. Y. Kim, and S. Y. Na, “Human Activity Recognition Using New Multi-Sensor Module in Mobile Environment,” 2011.
12. Ming Zeng, Le T Nguyen, Bo Yu, Ole J Mengshoel, Jiang Zhu, Pang Wu, and Joy Zhang. Convolutional neural networks for human activity recognition using mobile sensors. In *Mobile Computing, Applications and Services (MobiCASE), 2014 6th International Conference on*, pages 197–205. IEEE, 2014.
13. Jussi et al. Parviainen. Adaptive activity and environment recognition for mobile phones. *Sensors*, 2014.
14. A.A. Efros, A.C. Berg, G. Mori and J. Malik, "Recognizing Action at a Distance", *Proc. Ninth IEEE Int'l Conf. Computer Vision*, pp. 726-733, 2003.

التعرف علي أنشطة الانسان باستخدام الهاتف الذكي

في البدايه كان مايجعل من عملية التعرف علي أنشطة الانسان ممكنا وجود مصلطلح" Big Data" الموجود في الاجهزة الذكية التي يتم ارتداؤها" Wearable devices" والتي تجمع كميات كبيرة من البيانات الخاصة بمستخدميها فأصبح بأمكانها التعرف على نشاط الانسان وطرح السؤال هل يمكن استخدام آليات الذاكاء الاصطناعي وتعلم الالة Machine learning في أن تتعلم من الخبرة"experience" عن طريق بناء خوارزمية ML Model تتعلم من البيانات المعطاة لها والمعبرة عن بيانات السينسوروتحمل الأنشطة المراد تصنيفها وتحدد اذا كان الشخص يقوم بالمشي ام صعود السلم ام النزول ام الوقوف ام الجلوس ام الركض.

فمشكلة التعرف علي أنشطة الانسان هي عملية تتم بدون تدخل باستخدام بيانات سينسور Accelerometer ثلاثي الاستشعار وقمنا باستخدمنا الشبكات العصبية (NN) التي هي جزء من تقنيات التعلم العميق (DL) في التنفيذ وبالتحديد CNN، وتم تحسين ML model حتي أعطي أفضل نتيجة ممكنة حتي لاينحاز وقت التدريب علي البيانات المعطاه له و يؤدي بشكل سيء (underfitting) في بْيْة العمل وهي (Mobile platform) ، وعلي هذا توصلنا الي تطبيق APPقادر علي التعرف علي حركة الانسان اليومية باستخدام آليات الذكاء الاصناعي وتعلم الالة، كما وان التوجه الان في التكنولوجيا هو في دمج الذكاء الاصطناعي في تطبيقات الموبايل ومن مميزات هذه الطريقة عن غيرها كالساعات الذكية:

التكلفة المنخفضة و الخصوصية والأمان حيث لا يتم ارسال البيانات الي السحابة

الإلكترونية cloud حتى يتم معالجتها وانخفاض فترة الإنتظار low latency ويوجد العديد من التطبيقات لهذه التكنولوجيا وأهمها الموجودة فى السياق الطبى منها تتبع كبار السن فى التعرف على حالتهم