الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne démocratique et populaire

وزارة التعليم الـعـالي و البحـث العـلمـي
Ministère de l'enseignement supérieur et de la recherche scientifique

جـامعة سعد دحلب البليدة
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا
Faculté de Technologie

قسم الإلكترونيك
Département d'Électronique

Mémoire de Master

Filière Électronique

Spécialité Instrumentation

présenté par

MOULAY Fatima Zahra

Human activity recognition using Accelerometer

Proposé par :

YKHLEF Farid

Année Universitaire 2019-2020-

Acknowledgments

To quote Alfred North Whitehead:

“No one who achieves success does so without acknowledging the help of others.The wise and confident acknowledge this help with gratitude”.

Many thanks to my family and loved ones, teachers, workers and everyone who contributed to this field, none of this would have been possible without your help.

ملخص:

في الوقت الحالي ، هناك العديد من تطبيقات الهواتف الذكية التي تستخدم بالفعل تلك البيانات لتقدير إحصائيات اللياقة البدنية الأساسية ، ولكن هذه التطبيقات تنتج طريقة سيئة لتقييم مستوى لياقة الشخص. دفعت هذه المشكلة الناس إلى اللجوء إلى الأجهزة القابلة للارتداء لتتبع أنشطة اللياقة البدنية الخاصة بهم. نظرًا لأن الهواتف الذكية أصبحت سريعًا جزءًا من الحياة في كل مكان في العالم الحديث ، فإن أجهزة الاستشعار المدمجة بها لديها القدرة على تسجيل كمية كبيرة من البيانات حول حركة الأشخاص ، لذلك في هذه الورقة ، سنختبر أساليب مختلفة للتعلم الآلي للتنبؤ بالإنسان النشاط باستخدام مستشعرات التسارع والجيروسكوب.

كلمات المفاتيح:مقياس التسارع، الهواتف المحمولة، التعرف على النشاط، التعلم الآلي.

Résumé:

À l'heure actuelle, de nombreuses applications pour smartphone utilisent déjà des données pour estimer des statistiques de base sur la condition physique. ces applications produisent une mauvaise méthode d'évaluation du niveau de forme de la personne. Ce problème a conduit les gens à Se tournent vers les appareils portables pour suivre leurs activités de remise en forme. Les smartphones deviennent rapidement une partie omniprésente de La vie dans le monde moderne, leurs capteurs embarqués ont la capacité d'enregistrer une quantité importante De données sur les mouvements des personnes. Nous allons donc tester différents Approches pour prédire l'activité humaine à l'aide de capteurs accéléromètres et gyroscopiques.

Mots clés:Accéléromètre, téléphones portables, reconnaissance d'activité, apprentissage automatique.

Abstract:

Presently, there are numerous cell phone applications that as of now utilize that information to gauge fundamental wellness statistics.But these applications produce a helpless technique for evaluating person'sfitness level. This issue has lead individuals to go to wearable gadgets to follow their wellness exercises. As cell phones are rapidly turning out to be pervasive piece of the life in the advanced world, their installed sensors can record a lot of information about individuals' movement.So in this theory, we will test diverse AI ways to deal with anticipate the human action utilizing accelerometer and whirligig sensors.

Keywords:Accelerometer, cell phones, movement acknowledgment,machine learning

List of acronyms and abbreviations

A

ACC,Accelerometer

C

CSV,Comma Separated Values

CV Cross Validation

D

DCT,Discrete Cosine Transform

E

Etc

F

FFT,Fast fourirer transform

G

Gyro,Gyroscope

H

HAR,Human Activity Recognition

Hz,Hertz

I

i.e,id test (in other words)

K

KNN,K nearest neighbours

M

MFCC,Mel frequency Cepectrum Coesfficient

P

PCA,Principle Component Analysis

R

RFE, Recursive feature elimination

RMS, Root Mean Square

S

SVM,Support Vector Machine

T

TSNE, t-distributed stochastic neighbour embedding

Contents

General introduction

Chapter 1 Data cquisition and feature extraction

1.1 Introduction ……3
1.2 Sensor based human activity recognition ……3
a. Accelerometer ……5
b. Gyroscope ……6
1.3 Data acquisition and feature extraction ……7
1.3.1 Data acquisition ……7
a. Sampling Rate……8
b. Data Preprocessing……9
c. Windowing……9
d. Dimensionality Reduction……10
1.4 Feature Extraction……11
1.4.1 Time-domain features……11
1.4.2 Frequency-domain features……13
1.4.3 Cepstral features……16
a.MFCC……16
1.5 Conclusion…… 23

Chapter 2 Basic Artificial Intelligence Technique

2.1 Introduction……24
2.2 Artificial intelligence techniques……24
2.2.1 Supervised Learning Algorithms……25
2.2.2 Supervised Machine learning algorithm types……26
2.2.3 Support vector machine ……27
a. Support Vectors……29
b. Degree of tolerance……31
c. Kernel Trick……31
2.2.4 Random Forest Classifier……33
2.2.5 K-Nearest neighbors……33
2.3 Unsupervised learning……35
2.3.1 Data Compression……35
2.3.2 Principal Component Analysis……35
2.3.3 Confusion Matrix……37
a. Definition of the Terms……37
b. Performance evaluation ……37
c. Recall……38
d. Precision……38

e. High recall, low precision..38
f. Low recall, high precision..38
g. F-measure..38
2.4 Online and offline recognition...39
2.4.1 Online recognition...39
a.Vigilante..39
b. ActiServ 39
c.Brezmes 39
2.4.2 Offline recognition...40
a.Bao 40
b.Khan 40
2.5 conclusion...40

Chapter 3 Human Activity Recognition: Experiment Results

3.1 Introduction..41
3.2 Data capture overview...41
3.2.1 Data collection...41
3.2.2 Activities Labelling..42
a. General Visualizations..41
b. Detailed visualizations...42
3.3 Signal processing..47
3.3.1 Preprocessing..47
a. Data Splitting..47
b. Noise filtering...47
c. Magnitude...48
d. Windowing..48
3.4 Exploring UCI HAR Dataset 49
3.4.1 Distribution of Activities...49
a. Separating activities using T-SNE...50
3.4.2 Participant Exploration...51
a. How Good Are the Participants Separable...51
b. Which Sensor Is More Important For Classifying Participants By Walking Style.....................51
c. Participants walk analysis 52
d. How Long Does The Participant Use The Staircase...53
e. Participants Walk Analysis 2: Walk style analysis for each participant 54
f. Exploring Personal Information...55
3.5 Feature extraction 55
3.5.1 Statistically derived features...57
3.6 Selecting model...57
3.6.1 Model training 58
3.6.2 Feature selection..59

3.6.3 Learning curve……………………………………………………………………………………………………..59
3.6.4 Applying RFECV with svm classifier………………………………………………………………………..60
3.6.5 Variance Threshold………………………………………………………………………………………………..61
3.6.6 Dimension Reduction using PCA (Principal Component Analysis) ………………………………..61
a. Applying PCA with number of components=21……………………………………………………….62
b. Checking Accuracy after applying PCA…………………………………………………………………….62
3.6.7 Confusion matrix…………………………………………………………………………………………………63
3.6.8 Parameter Tuning …………………………………………………………………………………………………63
3.7 Conclusion………..64

List of figures

Figure 1: Wearable Sensors……………………………………………………………………………………………….5
Figure 2: Acceleration axes……………………………………………………………………………………………..6
Figure 3: Smartphone The axis of gyroscope sensor in………………………………………………………....6
Figure 4: Feature extraction process……………………………………………………………………………..…..7
Figure 5: Wavelet energy……………………………………………………………………………………………..…..14
Figure 6: MFCC block diagram……………………………………………………………………………………..….17
Figure 7: Hamming window…………………………………………………………………………………………..….18
Figure 8: Hamming window applied to a frame……………………………………………………………...………18
Figure 9: Spectrum for Fourier transform………………………………………………………………….………..19
Figure 10: Plot of mels versus hertz..…………………………………………………………………………..…...20
Figure 11: Mel filter bank……………………………………………………………………………………….………..21
Figure 12: Machine learning methods…………………………………………………………………………………25
Figure 13: Arichitecture of supervised machine learning………………………………………………………..26
Figure 14: Maximum Margin Classifier……………………………………………………………………………….28
Figure 15: Hyperplane in 2D and 3D space…………………………………………………………………………..28
Figure 16: Possible Hyperplanes and Hyperplane with maximum margin………………………………29
Figure 17: Visual representation of Hyperplane in 3D…………………………………………….……………29
Figure 18: Support vectors…………………………………………………………………………………………….....30
Figure 19: Non-linearly separated data……………………………………………………………………………...30
Figure 20: Degree of tolerance………………………………………………………………………………………….31
Figure 21: Pictorial representation of Kernel Trick……………………………………………………………….32
Figure 22: Random Forest Classifier…………………………………………………………………………………..33
Figure 23: K-Nearest Neighbor Classifier…………………………………………………………………………….34
Figure 24: Dimensionality reduction with PCA………………………………………………………………………36
Figure 25: Number of rows per activity,number of useful rows per activity……………………………42
Figure 26: Mean time associated to each activity…………………………………………………………………. .43
Figure 27: Acceleration and gyroscope signals of the 1st experience……………………………………….44
Figure 28: Acc signals captures while the user 01 in experiment 01 was walking……………………44
Figure 29: Gyro signals captures while the user 01 in experiment 01 was walking…………………..45
Figure 30: Acc and Gyro signals captures while the user 01 in experiment 01 was LAYING……...46
Figure 31: Acc and Gyro signals captures while the user 01 in experiment 01 (LIE TO SIT)……….46
Figure 32: Data splitting…………………………………………………………………………………………………….47
Figure 33: The same acc signal before and after applying Median filter……………………………………48
Figure 34: Magnitude...…………………………………………………………………………………………………….48
Figure 35: Distribution of Activities..……………………………………………………………………………………49
Figure 36: Separating activities using T-SNE..……………………………………………………………………...50
Figure 37: Comparative Sensor Importance for classification. ……………………………………………….52

Figure 38: Comparative Sensor Importance for classification (Acc & gyro) ……………………………….52
Figure 39: Walk time for 30 participants……………………………………………………………………………53
Figure 40: participants compared by their staircase walking duration……………………………………….53
Figure 41: Normality check for the Walk data…………………………………………………………………….54
Figure 42: Walk patterns for individual participants……………………………………………………………….54
Figure 43 : T-SNE walkig style by participant……………………………………………………………………….55
Figure 44: Recognition performance for different classifiers…………………………………………………..58
Figure 45: Learning curve……………………………………………………………………………………………………..59
Figure 46: Applying RFECV with svm classifier…………………………………………………………………….60
Figure 47: Dimension Reduction using PCA………………………………………………………………………….61
Figure 48: PCA 2D projection..……………………………………………………………………………………………62
Figure 48: Confusion matrix……………………………………………………………………………………………….63

<u>Liste des tableaux</u>

Table 1: Smartphone sensors……………………………………………………………………………………………….7
Table 2: Class confusion matrix……………………………………………………………………………………………….37
Table 3: Distribution of Activities……………………………………………………………………………………………….49
Table 4: Time to gather data………………………………………………………………………………………………..51
Table 5: Features…….56
Table 6: Accuracy by applying RFE with optimal number of features……………………………………………60
Table 7: Accuracy after applying Variance threshold…………………………………………………………………61
Table 8: Accuracy after applying PCA..…………………………………………………………………………………….62
Table 9: Accuracy after applying Parameter tuning.…………………………………………………………………63

General Introduction

During the past decade, there was an exceptional development of microelectronics and computer systems, enabling sensors and mobile devices with unprecedented characteristics. Their high computational power, small size, and low cost allowed people to interact with these devices as part of their daily living. That was the genesis of Ubiquitous Sensing, an active research area with the main purpose of extracting knowledge from the data acquired by pervasive sensors. Particularly, the recognition of human activities has become a task of high interest within the field, especially for medical, military, and security applications. For instance, patients with diabetes, obesity, or heart disease are often required to follow a well-defined exercise routine as part of their treatment . Therefore, recognizing activities such as walking, running, or resting becomes quite useful to provide feedback to the caregiver about the patient's behavior. An interactive game or simulator might also require information about which activity the user is performing in order to respond accordingly.

For this reason, the development of solutions that recognize human activities (HAR) through computational technologies and methods has been explored in recent years . In this sense, the HAR problem has previously been treated as a typical pattern recognition problem, and more specifically, a classification problem, that is, to identify the activity being performed by an individual at a given moment. For this reason, most HAR solutions have been developed using artificial intelligence methods through various machine learning techniques, including shallow (e.g., Support Vector Machine (SVM), Decision Tree, Naive Bayes, and KNN) and deep algorithms (e.g., Convolutional Neural Network (CNN), Recurrent Neural Network (RN)).

Our aim is to implement a system of activity recognition able to classify six activities(3 static activities and 3 dynamic activities),and a brief review of postural transition static movements. the thesis work focuses on algothim for both feature extraction and classification .The developed system is tested and evaluated through existing common methods in recent literature.A large part of the work has to be concerned with the problem of human activity recognition.So to achieve this ,we must go through these steps:

1. Data acquisition from Smartphone

2.Accelerometer and gyroscope

3.Artificial Intelligence techniques for human activity recognition

4.Python description and implementation

We will end this dissertation by defining the various research perspectives that will be interesting to continue this work.

Chapter 1 Data cquisition and feature extraction

1.1 INTRODUCTION

When computers came into existence, researchers wanted to make them act like humans.These efforts created a new field of science called artificial intelligence. AI can help make what is impossible for humans to do on their own, possible. Computer vision,a major subfield of AI, involves translating real-world pixels into a language that computers can understand and work with. Specifically, human-centric computer vision develops algorithmic tools that allow machines to detect and track humans. Human motion data can then be used to generate quantifiable insight into human behaviour, and create interactive applications within real, augmented, or virtual worlds. In other words, human-centric computer vision provides the tools needed for machines to work and interact with humans.

a great number of different types of chronic data, for instance, video recorders, photos streams and spatial-temporal logs, there will be the significant need for personal customization using human activity recognition The task of human activity recognition (HAR) is to classify body gesture or motion, and then determine or predict states of action or behaviour. Its extensive applications, appearing in military health care, physical recovery from disability or injury and clinical deformity correction, are drawing more and more attention on the further development and exploitation from industry and academe. Especially, in public health care, with the pervasion of portable personal digital devices such as smart phones, intelligent watches and multi-media terminals, generating.

In this chapter,we discuss the human activity technology then we introduce the methodology of extracting data from smartphone including signal processing that make it possible.

1.2 SENSOR BASED HUMAN ACTIVITY RECOGNITION

Human activity recognition has been addressed in two different ways, namely using external sensors and wearable sensors[1]. In the case of external sensors, the devices are fixed in foreordained points of interest, so the deduction of activities entirely depends on the voluntary interaction of the users with the sensors. In the case of wearable sensors, the devices are attached to the user[1][2]. We discuss each of the approaches in detail in the later sections

- ***External Sensor***

Video Signals from cameras are used as input for the recognition of gestures. Hand and body gestures are detected using an edge detection algorithm. These algorithms are computationally expensive and dedicated hardware is required to detect the gestures or body movement. A research carried at Cambridge on digits, utilizes a camera for the gesture control offered by Kinect.The camera can be worn on a wrist with the help of a wrist strap and it tracks 2D movement of the hand rather than 3D. It could also recognize finger movement that is used to control the software.This approach can be used in applications such as controlling the games with hand movements and translation of a sign language into a written text.[2]

- ***Wearable Sensors***

To be successful in practice, HAR systems should not require the user to wear many sensors nor interact too often with the application.

Smartphone based applications are an increasingly prominent solution as smartphones have become an indispensable part of our daily life. Especially with the rapid evolution of hardware, ever-increasing computing and networking capacity, and rich embedded sensors, smartphone based HAR systems can tell us different kinds of human activities in real time using machine learning techniques. In addition, using smartphones for human activity recognition has a wide range of applications including healthcare, daily fitness recording, anomalous situation alerting, personal biometric signature identification, and indoor localisation and navigation. All this benefits from the fast development of mobile phone software and hardware.Smartphones in the market have embedded sensors, and the advanced MEMS (micro-electro-mechnical systems) design has enabled low-power and high-quality sensors for mobile sensing. The best-known MEMS sensors in smartphones are accelerometer and gyroscope, but there are a lot more MEMS sensors in today’s mobile device like electronic compass, pressure sensor, light sensor, and microphone.[3][4]

Figure1 Wearable Sensors

a. Accelerometer

The accelerometer sensor is a device that can measure acceleration (the rate of change in velocity), but in smartphones, it is able to detect changes in orientation and tell the screen to rotate.Basically, it helps the phone know up from down directions. The main use for this sensor within smartphones is to measures the linear acceleration of the device on the X- axis (lateral), Y-axis (longitudinal), and Z-axis (vertical).

It has been used heavily in smartphone sensors based activity recognition. For instance, if a user changes his/ her activity from walking to jogging, it will react on the signal shape of the acceleration reading: along the vertical axis there will be an abrupt change in the amplitude. Furthermore, the acceleration data could demonstrate the motion pattern within a given a period,which is helpful in the difficult HAR. Due to enhancements in microelectromechanical systems (MEMS) technology, today's accelerometers are not only coming in small size and at a low-price but are also capable of demonstrating a high degree of reliability in measurement.Accelerometers use transducers for measuring acceleration. These come in different varieties, such as piezoelectric crystals, piezo resistive sensors, servo force balance transducers, electronic piezoelectric sensors and variable capacitance accelerometers.Examples of some of the accelerometer based gesture recognition devices include Nintendo Wii-mote, Chronos watch from Texas Instruments , Fitbit and smart phones. The Nintendo Wii-mote contains an integrated 3-axis acceleration sensor, and connects via the Bluetooth human interface device protocol for transmitting data. [4][5]

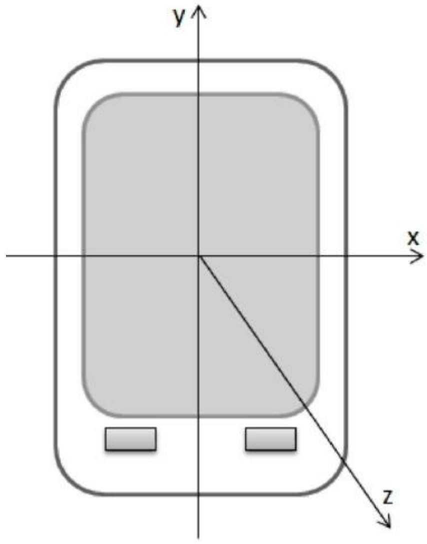

Figure 2 Acceleration axes

b. Gyroscope

The gyroscope is a device that can provide orientation information as well, but with greater precision. The gyroscope also has been used in smartphones to measure the phone's rotation rate by detecting the roll, pitch, and yaw motions of them along the x, y, and z axis,respectively. It adds an additional dimension to the information supplied by the accelerometer by tracking rotation or twist. Also, it measures the angular rotational velocity and rate of change. The accelerometer sensor can give either a "noisy" information output that is responsive, or give a "clean" output that's sluggish. But when we combine the 3-axis accelerometer with a 3-axis gyroscope, we get an output that is both clean and responsive at the same time.[6]

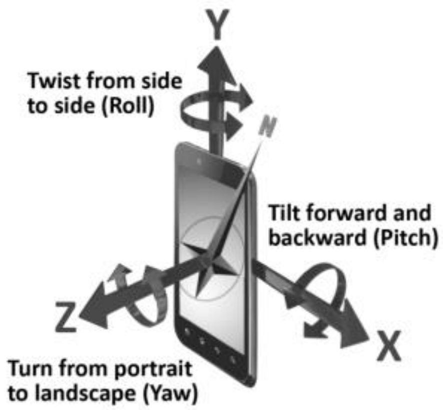

Figure 3 Smartphone The axis of gyroscope sensor in

Table 1 shows descriptions of accelerometer and gyroscope sensors, as well as their functions that are supported by the Android platform.

Hardware	sensors	communication	Service
CPU Battery Internal memory Touchscreen Sid card Sim Card speakers	Camera Accelerometer gyroscope compass proximity GPS Barometer Light sensor Microphone	WI FI GPRS EDGE BLUETOOTH USB 2G,3G,4G NFC	SMS MMS EMAIL INTERNET Third Party App Radio gaming

Table 1: Smartphone sensors

A gyroscope has been used in many applications such as inertial navigation systems, aerial vehicles for stability augmentation (e.g. in quadcopters) and recently, it has been introduced in electronic devices (e.g. smartphones, game consoles) for enhancing user interfaces and gaming experience. For HAR, this sensor has been employed in various applications such as for the detection of activities (e.g. Walking, stairclimbing) and transitions between postures (e.g. From standing to sitting). [7]

1.3 Data acquisition an feature extraction

Activity recognition using smartphone sensors involves identification of which activity is being performed at a particular moment based off of the data collected by the sensors on the human body[8]. Some of the processes in activity recognition include:

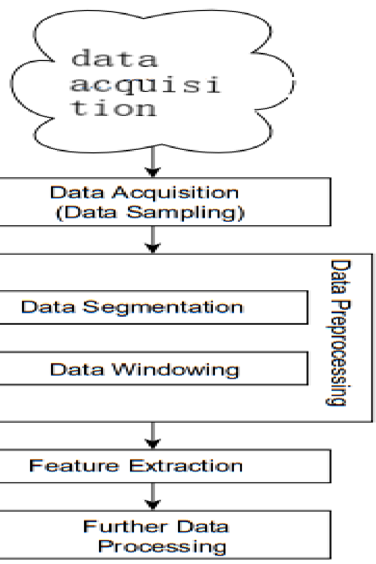

Figure 4: Feature extraction process

1.3.1 Data acquisition

Data acquisition is the process of sampling signals that measure real world physical conditions and converting the resulting samples into digital numeric values that can be manipulated by a computer.[9]

Data acquisition is a critical part in HAR. The objective of HAR is analysis or interpretation of the data that is being tracked so as to derive insight from it. Studies suggests that acquiring data involves signal preprocessing which focuses on filtering and extracting the important features that will be required in order to train classifier

The basic idea in the activity recognition based on accelerometer measurements is to map each measurement to some activity label, and of course, this activity label should be the same as the actual performed activity. Therefore, the problem of recognizing activities from data can be defined as follows:

Given activity labels **A = {y0,..., yn−1}** and a time series **S = [{t0,m0},...,{tk ,mk}]**, where t_i **.** is the time stamp and **mi** vector of inertial measurements collected at this time stamp. The purpose of the activity recognition process is to find mapping **f : mi → A**, so that **f(mi)** is as similar as possible to the actual activity performed at t_i .[10]

a. Sampling Rate

An important factor that affects the quality of data that is captured is the sampling rate.The sampling rate is the frequency with which different sensors capture data[10].

Sampling can be done for functions varying in space, time, or any other dimension, and similar results are obtained in two or more dimensions .For functions that vary with time, let s(t) be a continuous function (or "signal") to be sampled, and let sampling be performed by measuring the value of the continuous function every T seconds, which is called the sampling interval or the sampling period. Then the sampled function is given by the sequence:

s(nT), for integer values of **n**.

The sampling frequency or sampling rate**,** f_s**,** is the average number of samples obtained in one second (*samples per second*), thus

$$f_s = \frac{1}{T_s} \tag{1.1}$$

As :

$$t = nT_s \tag{1.2}$$

The sampling theorem ensures that original analog signal can be reconstructed from a finite set of N samples.the condition must be :

$$f_s \geq 2f_h \tag{1.3}$$

Where f_h is the highest frequency in a(t) also known as Nyquist frequency

b. Data Preprocessing

The purpose of the pre-processing can be to speed up computation by reducing the amount of data Often the data that is obtained from the real-world can be inconsistent or even incomplete. There is also a possibility of having errors in the data-types of the obtained data. Hence it is necessary to clean and pre-process data before it can be used to make predictions. One of the most commonly occurring errors is that of missing values. There might be times where a particular sensor fails to register reading for a particular part of the human body[11]. There are times when multiple sensors might fail to register the readings as well. One of the common ways to deal with missing values is to drop a particular row that has a missing reading for a feature. However, dropping a row means loss of data. It is possible that there are hundreds of rows that contain missing values. Dropping all of these rows will cause us to lose important feature information that can otherwise be retained in order to train our models for better prediction. Another way of dealing with missing values is to calculate an arithmetic mean, median or mode of the missing feature and replace it with the missing value.[12]

c. Windowing

Efficient data acquisition is critical when it comes to mobile devices as they have a limited energy source. A potential solution to this problem is to use a technique called windowing, in which the sensors capture data only during a small timeframe. The length of each window is decided in such a way that it can accurately determine each activity and at the same time enables energy preservation[13].

To extract features from raw data, windowing techniques are generally used, which consist of dividing sensor signals into small time segments. Three types of windowing techniques are usually used Human activity recognition:

- sliding window ,where signals are divided into fixed-length windows;[11]
- event-defined windows, where pre-processing is necessary to locate specific events, which are further used to define successive data partitioning
- activity-defined windows ,where data partitioning is based on the detection of activity changes.

The window size is very important and according to various research should contain at least on cycle of an activity to uniquely identify it, this means that the window size can be anywhere between **1** and **7.5 s** depending on the required activity recognition.

The sliding window approach is well-suited to real-time applications since it does not require any pre-processing treatments.

d. Dimensionality Reduction

Sensors used to record human activity data are usually high in dimension. In order to train classifiers for recognizing human activity, it is necessary to transform the data in such a way that it reduces the dimensionality while still retaining useful and important information necessary for HAR. Dimensionality reduction aims at reducing the random variables that are under consideration. Data is collected by multiple sensors such as accelerometer, gyroscope and magnetometer which collect data for 3 different axes (x-axis, y-axis, z-axis). As a result of this, the number of features that are collected is way too large to be used for training any machine learning classifier.Even though this data can be used for training a classifier, it would most likely cause over-fitting of data. This leads to a well-known issue commonly referred to as the curse of dimensionality. demonstrates that the different features that are captured by sensors are of little interest and only a subset of the entire feature set contributes to the features that are worthy of interest. In order to obtain an effective and robust characterization of the domain, dimensionality reduction can prove to be a useful tool . In order to make useful predictions regarding what activity is being done, the aim is to minimize the number of predictors and also make sure that these predictors are independent of each other.also found that using only the most informative features that capture the maximum variance in data contribute towards higher accuracy of the mode[14].

1.4 Feature extraction

The segmented data is collected as a series of instances containing three values corresponding to acceleration along the x-axis, y-axis, and z-axis. Feature extraction, converts the signals into the most significant and powerful features which are unique for the activity. The features can be extracted using various feature extraction tools. The features can be extracted in both Time and Frequency domain.We extract six frequency domain features from each window for each axis x, y and z (3 for accelerometer and 3 for Gyroscope). Features ¨are extracted as feature vectors X_i on the set of segments *W,* with Ƒ being the feature [10][11][12].

$$X_i = Ƒ(D', w_i) \tag{1.4}$$

1.4.1 Time-domain features

Here are some of extraction tools utilised in time domain:

- **Magnitude**

Magnitude which means a measure of the normalized value of the FFT coefficients and facilitates the recognition of the differences between activities[15].

$$SV\ Mag = \sqrt{(A^2{}_x + A^2{}_y + A^2{}_z)} \tag{1.5}$$

- **Mean**

Mean is the most common and easy implemented feature of the time domain. It only finds the mean of EMG amplitude values over sample length of the signal[16].

$$mean(\mu) = \frac{1}{N}\sum_{n=1}^{N} x_n \tag{1.6}$$

- **Standard deviation**

the standard deviation is a measure of the amount of variation or dispersion of a set of values[16].

$$std(\sigma) = \sqrt{\frac{1}{N-1}\sum_{n=1}^{N}(x_{n-}\mu)^2} \qquad (1.7)$$

- **Variance**

The Variance is defined as the average of the squared differences from the Mean.

$$var = \frac{1}{N-1}\sum_{n=1}^{N}(x_{n-}\mu)^2 \qquad (1.8)$$

- **Skewness**

Skewness is measure of asymmetry of a signal or measure of third order cumulative

$$skew = \frac{\frac{1}{N}\sum_{n=1}^{N}(X_{n-}\mu)^2}{\sigma^3} \qquad (1.9)$$

- **Kurtosis**

Kurtosis is measure of peakness of probability distribution or measure of fourth order cumulative.

$$kurt = \frac{\frac{1}{N}\sum_{n=1}^{N}(X_{n-}\mu)^4}{\sigma^4} \qquad (1.10)$$

- **Mean absolute deviation**

The average of the absolute deviations of data points from their mean.

$$MAD = \frac{1}{N}\sum_{n=1}^{N}|x_n - ORT| \qquad (1.11)$$

- **Max and Min**: Show the changing range of the signal.

- **Mean Crossing Rate**

$$\mu_x = \frac{1}{N} \sum_{n=0}^{N-1} x(n) \quad (1.12)$$

- **Root mean square**

$$E_{rms} = \frac{1}{N} \sqrt{\sum_{n=0}^{N-1} x^2(n)} \quad (1.13)$$

- **Autocorrelation**

Autocorrelation is the average of the product of a data sample *x*[*n*] with a version of itself advanced by a lag. The autocorrelation function is described by the equation[18]

$$r_{xx}[k] = \frac{1}{N} \sum_{n=1}^{N-k} x[n]x[n+k] \quad (1.14)$$

1.4.2 Frequency-domain features

Which are obtained by converting the time based signal into the frequency domain using the Fourier Transform, like: fundamental frequency, frequency components[17],some feartures are described below

- **Energy**

Energy is elicited by the summation of the squared FFT parameters called as coefficients.

- **Wavelet energy**

A wavelet is a mathematical function used to divide a given function or continuous-time signal into different scale components. Since wavelet transform has advantages over traditional Fourier transforms for representing functions in Signal to Noise RatioSignal to Noise Ratio is a ratio of the signal power and noise power. The signal power and noise power are estimated separately[18].

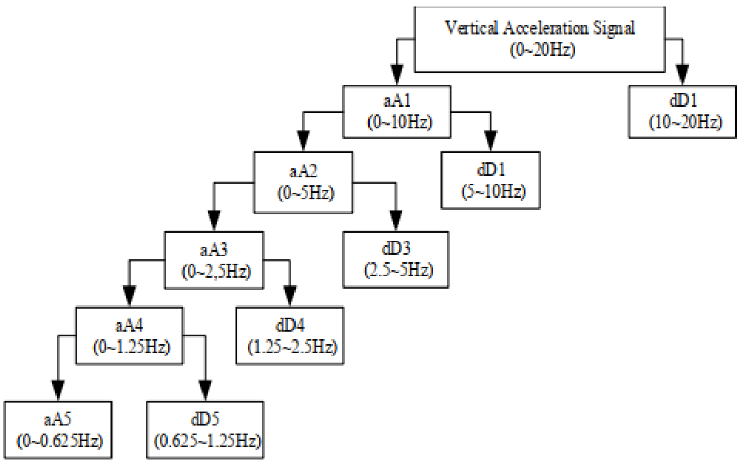

Figure 5 Wavelet energy

- **Discrete Fourier Transform**

DFT is used to compute frequency spectrum of the discrete data signal x. The DFT is described as follows:

$$X(f) = \sum_{i=0}^{N-1} x_{ie}{}^{\frac{-j2\pi fi}{N}} \quad (1.15)$$

where X denotes the frequency spectrum, f the f *th* Fourier coefficient in the frequency domain and N the length of the sliding window. Equation can be rewritten using the following form[3]:

$$X(f) = \sum_{i=0}^{N-1} a_i + jb_i \tag{1.16}$$

With:

$$a_i = x_i cos(\frac{2\pi f i}{N})$$

$$b_i = x_i sin(\frac{2\pi f i}{N})$$

One of the most important frequency-domain features used for human activity recognition is the Power Spectral Density (PSD). This feature has been used by to recognize activities such as walking, running and driving. PSD can be computed as the squared sum of its spectral coefficients normalized by the length of the sliding window:

$$P(f) = \frac{1}{N} \sum_{i=0}^{N-1} a^2{}_i + b^2{}_i \tag{1.17}$$

Peak frequency represents the frequency corresponding to the highest computed power spectrum density over the sliding window. The peak frequency has been used in several studies related to activity recognition.[3]

- **Entropy**

Another feature that is widely used in human activity recognition . Generally,this feature helps to discriminate between activities that have different patterns of movement . Entropy can be formulated as follows:

$$H(f) = \frac{1}{N}\sum_{i=0}^{N-1} c_i log\ (c_i) \quad , c_i = \frac{\sqrt{a^2{}_i + b^2{}_i}}{\sum_{k=0}^{N-1}\sqrt{a^2{}_i + b^2{}_i}} \qquad (1.18)$$

The DC component is another important feature also used in human activity recognition .It represents the PDS at frequency $f = 0$ Hz. It can be formulated as the squared sum of its real spectral

coefficients normalized by the length of the sliding window:

$$DC = \frac{1}{N}\sum_{i=0}^{N-1} a^2{}_i \qquad (1.19)$$

1.4.3 Cepstral features

a. MFCC

MFCC is an advanced feature extraction methods in speech signal processing[21],coefficients are obtained through the following process:

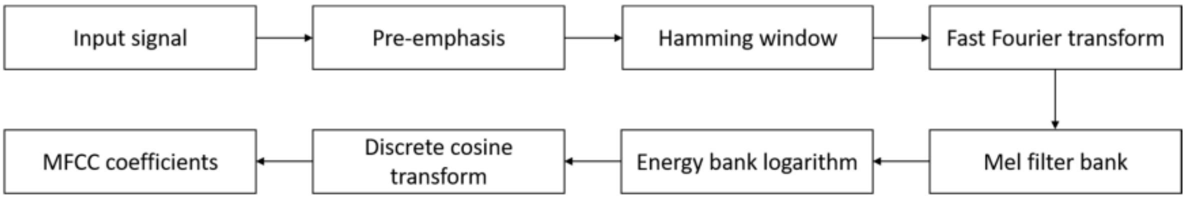

Figure 6: MFCC block diagram

Here are six steps in computing MFCC coefficients. The first step is amplifying the high frequency on the signal by applying a pre-emphasis filter. It was described that the pre-emphasis step may be unnecessary because the frequency range in the inertial signal is smaller than the speech signal[19].

- **Hamming window**

A good window function has a narrow main lobe and a low side lobe. A smooth tapering at the edges is desired to minimize discontinuities[21]. The most common window used in speech processing is the Hamming window. Each frame has to be multiplied with a hamming window in order to keep the continuity of the first and the last points in the frame If the signal in a frame is denoted by **s(n), n = 0,…N-1**, then the signal after Hamming windowing is **s(n)*w(n)**, where **w(n)** is the Hamming window defined by:

$$\mathbf{w(n, \alpha) = (1 - \alpha) - \alpha \cos (2\pi n / (N-1))} \quad 0 \leqq n \leqq N-1$$

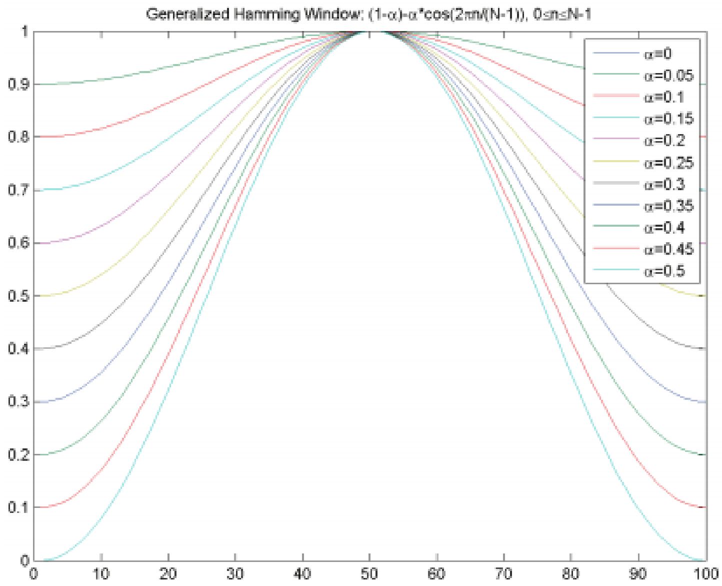

Figure 7: Hamming window

Below we can see a frame before and after the Hamming window is applied. The result is achieved by multiplying the Hamming window curve above with the curve in the frame below.

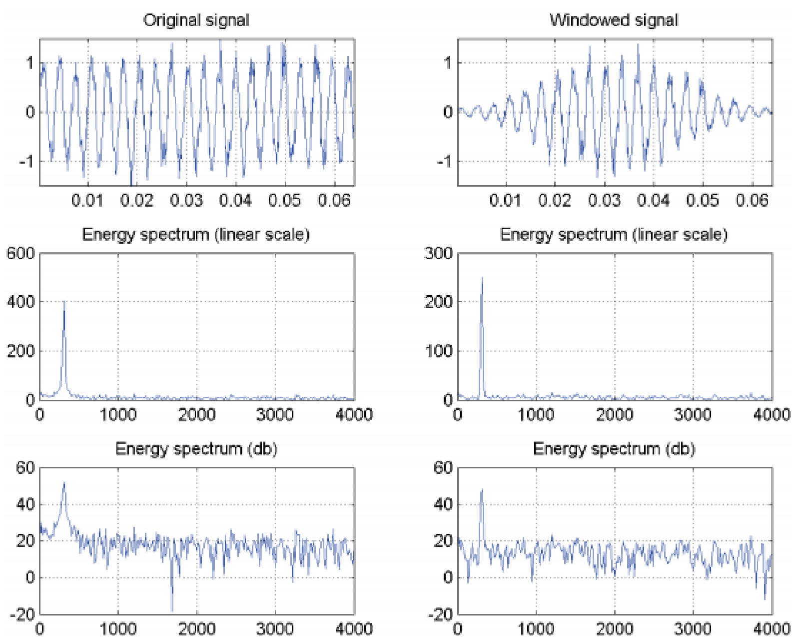

Figure 8: Hamming window applied to a frame

❖ FFT

The next step is the application of Fast Fourier Transform , which converts each frame of **N** samples from the time domain into the frequency domain. The FFT which is a fast algorithm to implement the Discrete Fourier Transform (DFT) is defined on the set of N samples **{xn}**, as follows:

$$X_k = \sum_{n=0}^{N-1} x_n \, e^{\frac{-2\pi kn}{N}}, k = 0, \ldots\ldots, N-1 \qquad (1.20)$$

In general X_ks are complex numbers and we consider only their absolute values. The resulting sequence X_k is interpreted as follows: positive frequencies **0 < f < Fs / 2** correspond to **0 < n < N / 2 - 1** values , while negative frequencies **– Fs /2 < f< 0** correspond to **N/2+1 < n < N-1.** Fs denote the sampling frequency. The result after this step is often referred to as spectrum or periodogram.

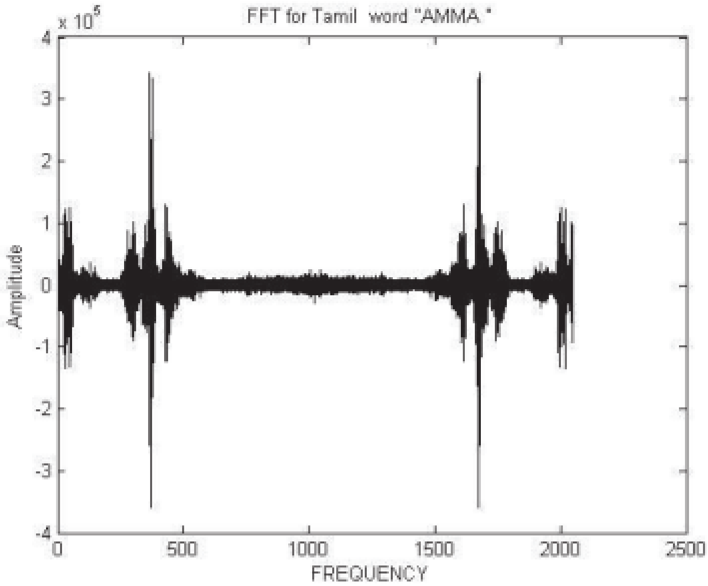

Figure 9: Spectrum for Fourier transform

❖ **Mel bank filter**

The Mel scale is based on how the human hearing perceives frequencies. It was defined by setting 1000 Mel equal to 1000 Hz as a reference point. Then listeners were asked to adjust the physical pitch until they perceived it as two-fold ten-fold and half, and those frequencies were then labeled as 2000 Mel, 10000 Mel and 500 Mel respectively. The resulting scale was called the Mel scale and is approximately linear below frequencies of 1000hz and logarithmic above. The Mel frequency can be approximated by the following equation[19]:

$$Mel(f) = 2595 * log_{10}(1 + \frac{f}{700}) \quad (1.21)$$

where, **f** is the actual frequency and $Mel(f)$ is the perceived one.

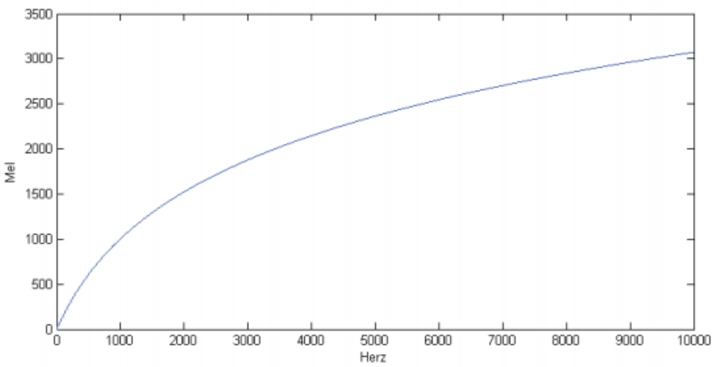

Figure 10: Plot of mels versus hertz.

The Mel frequency warping when calculating MFCCs is accomplished by the use of a triangular Mel spaced filter bank, which consists of several triangular shaped and Mel spaced filters. In practice, we have two choices for the triangular band pass filters. The reasons for using triangular band pass filters are twofold:

- Smooth the magnitude spectrum such that the harmonics are flattened in order to obtain the envelop of the spectrum with harmonics. This indicates that the pitch of a speech signal is generally not presented in MFCC. As a result, a speech recognition system will behave more or less the same when the input utterances are of the same timbre but with different tones/pitch.

- Reduce the size of the features involved[19].

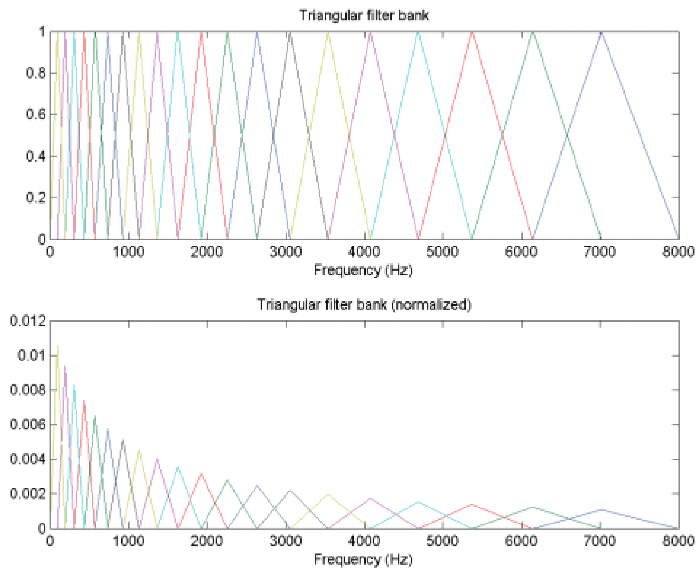

Figure 11: Mel filter bank

$$Y(i) = \sum_{j=1}^{N} S_j H_{ij} \tag{1.22}$$

Where, S_j is the N-point magnitude spectrum,

H_{ij} the sampled magnitude response of an M-channel filter bank.

The mel frequency filter bank is applied in the frequency domain. One approach towards simulating the subjective spectrum is to use a filter bank which is spaced uniformly on the mel-scale. The filter bank has a triangular band pass frequency response.The spacing and the bandwidth is determined by a constant mel frequency interval. We choose K, the number of mel spectrum coefficients to be 20. This filter bank being applied in the frequency domain simply amounts to applying the triangle-shape windows to the spectrum. A useful way to think about this filter bank is to view each filter as a histogram bin (where bins have overlap) in the frequency domain.

❖ **Discrete Cosine Transforming**

While the Mel filtering approximates the non-linear characteristics of the human auditory system in frequency, the natural logarithm deals with the loudness non-linearity. It approximates the relationship between the human's perception of loudness and the sound intensity. Besides this, it converts the multiplication relationship between parameters into addition relationship[20]. The convolutional distortions, such as the filtering effect of microphone and channel, and the multiplication in frequency domain, become simple addition after the logarithm. The log Mel filter bank coefficients are computed from the filter outputs as:

$$S(m) = 20log_{10}(\sum_{k=0}^{N-1} |X(k)|H(k))\,, 0 < m < M \qquad (1.23)$$

Where, M is the number of Mel filters (20 to 40),

$H(k)$ is the N-point FFT of the specific window frame of the input speech signal, and $H(k)$is the Mel filter transfer function .

❖ **Energy**

The next analysis involves adding the energy of the framed signals,the performance increase when adding energy of framed signals

❖ **Delta and Delta delta MFCC**

The final analysis for MFCC feature extraction is the adding of the deltas (Δ) and delta–deltas (ΔΔ) coefficients. The performance of a speech recognition system can be greatly enhanced by adding time derivatives to the basic static parameters. The deltas coefficient is also known as the differential and acceleration coefficient[21]. The signal derivatives provide information about the signal evolution. The deltas coefficients are defined as follows

$$d_t = \frac{\sum_{n=1}^{N} n(C_{t+n} - C_{t-n})}{2\sum_{n=1}^{N} n^2} \qquad (1.24)$$

Where d_t is a deltas coefficient at time **t** in terms of the static coefficients C_{t+n} **to** C_{t-n}.

1.5 CONCLUSION

In this chapter we discussed essential aspects regarding the recognition of human activities in the light of the existing research literature. It included common strategies for the development andevaluation of HAR systems.Additionally,feature extraction approaches for model evaluation whis is a clruucial step for every data collection.

2.1 INTRODUCTION

When most people hear the term artificial intelligence, the first thing they usually think of is robots. That's because big-budget films and novels weave stories about human-like machines that wreak havoc on Earth. But nothing could be further from the truth.

Artificial Intelligence is an approach to make a computer, a robot, or a product to think how smart human think. AI is a study of how human brain think, learn, decide and work, when it tries to solve problems. And finally this study outputs intelligent software systems.The aim of AI is to improve computer functions which are related to human knowledge, for example, reasoning, learning, and problem-solving.

Computers are faster when it comes to calculation and analytical abilities, but computers cannot take decisions on their own, that is they don't have the ability to make a decision. Empowering computers to make a decision with their own intelligence is Artificial Intelligence.

Here we come across Artificial Intelligence algorithms. These special algorithms are capable of finding patterns and coming up with a process to make a decision.

So in this chapter we're going to introduce AI techniques for evaluating a HAR system for simple activities for this we're going to use supervised learning ,we ll go through some of its algorithms ,and giving an approach to the online and offline recognition.

2.2 ARTIFICIAL INTELLIGENCE TECHNIQUES

Artificial Intelligence algorithm is a broad field which consists of Machine Learning algorithms as well as Deep Learning Algorithms. The main goal of these algorithms is to enable computers to learn on their own and make a decision or find useful patterns. Artificial Intelligence algorithms Learn from the data itself[22]. In a broader sense learning can be divided into 3 categories:

- Supervised Learning: When input and output both labels are known and the model learns from data to predict output for similar input data.
- Unsupervised Learning: When output data is unknown or it is needed to find patterns in data given, such type of learning is unsupervised learning.

- Reinforcement Learning: Algorithms learns to perform an action from experience. Here algorithms learn through trial and error which action yields greatest rewards. The objective is to choose actions that maximize the expected reward over a given amount of time.

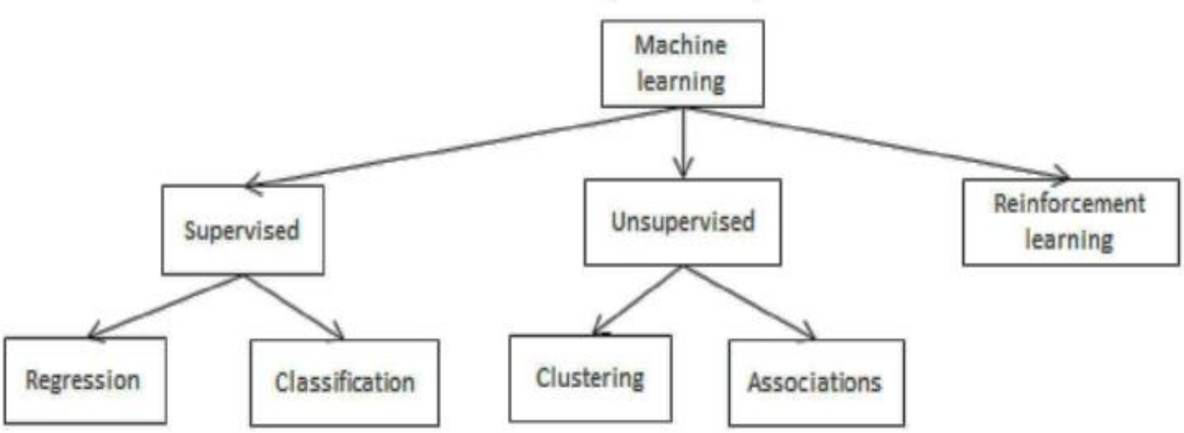

Figure 12: machine learning methods

2.2.1 Supervised Learning Algorithms

When we train the algorithm by providing the labels explicitly it is known as supervised learning. This type of algorithm uses the available dataset to train the model. The model is of the following form.

Y=f(X) where x is the input variable, y is the output variable and f(X) is the hypothesis[23].

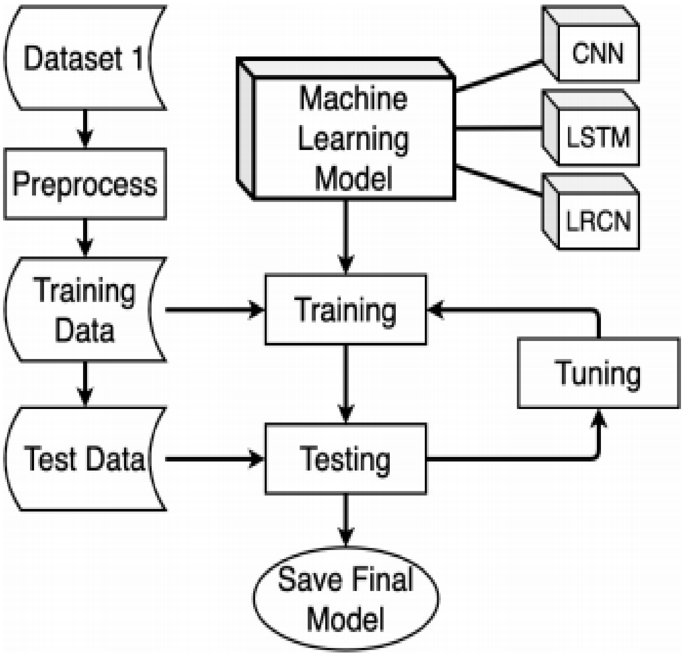

Figure 13: Arichitecture of supervised machine learning

2.2.2 Supervised Machine learning algorithm types

Classification is the process where incoming data is labeled based on past data samples and manually trains the algorithm to recognize certain types of objects and categorize them accordingly. The system has to know how to differentiate types of information, perform an optical character, image, or binary recognition (whether a particular bit of data is compliant or non-compliant to specific requirements in a manner of “yes” or “no”).supervised learning are grouped into regression and classification.

- **Regression**

  In Regression the output variable is numerical(continuous) i.e. we train the hypothesis(f(x)) in a way to get continuous output(y) for the input data(x)[24]. Since the output is informed of the real number regression technique is used in the prediction of quantities, size, values, etc.

For Example, we can use it to predict the price of the house given the dataset containing the features of the house like area, floor, etc.

Few popular Regression Algorithm is[24]:

- ❖ Linear Regression
- ❖ Support Vector Regression
- ❖ Poisson Regression

- **Classification**

In classification the output variable is discrete. i.e. we train the hypothesis(**f(x)**) in a way to get discrete output**(y)** for the input data**(x)**. The output can also be termed as a Few Popular Classification Algorithm is:

- ❖ Logistic Regression
- ❖ Neural Network
- ❖ Decision Tree
- ❖ Support vector machine(SVM)
- ❖ K-nearst Neighbors(K-NN)

2.2.3 Support vector machine

Support Vector Machines (SVM) are among one of the most popular and talked about machine learning algorithms.

They were extremely popular around the time they were developed in the 1990s and continue to be the go-to method for a high-performing algorithm with a little tuning[26].

SVM or Support Vector Machine is a linear model for classification and regression problems. It can solve linear and non-linear problems and work well for many practical problems. The idea of SVM is simple: The algorithm creates a line or a hyperplane which separates the data into classes.

At first approximation what SVMs do is to find a separating line(or hyperplane) between data of two classes. SVM is an algorithm that takes the data as an input and outputs a line that separates those classes if possible.

The objective of SVM is to find a hyperplane in an N-dimensional space (N-Number of features) that distinctly classifies the data points[26].

Support Vector Machine is a generalization of maximal margin classifier. This classifier is simple, but it cannot be applied to the majority of the datasets since the classes must be separated by a boundary which is linear. But it does explain how the SVM works.

In the context of support-vector machines, the optimally separating hyperplane or maximum-margin hyperplane is a hyperplane which separates two convex hulls of points and is equidistant from the two.

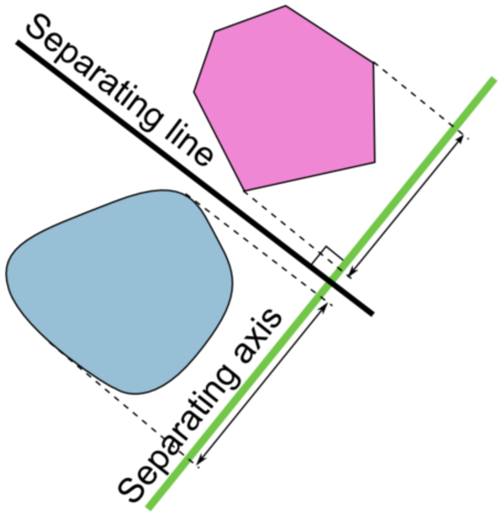

Figure 14: Maximum Margin Classifier

In an N-dimensional space, a hyperplane is a flat affine subspace of dimension N-1. Visually, in a 2D space, a hyperplane will be a line and in 3D space, it will be a flat plane.In simple terms, hyperplane is a decision boundary that helps classifying data points[26].

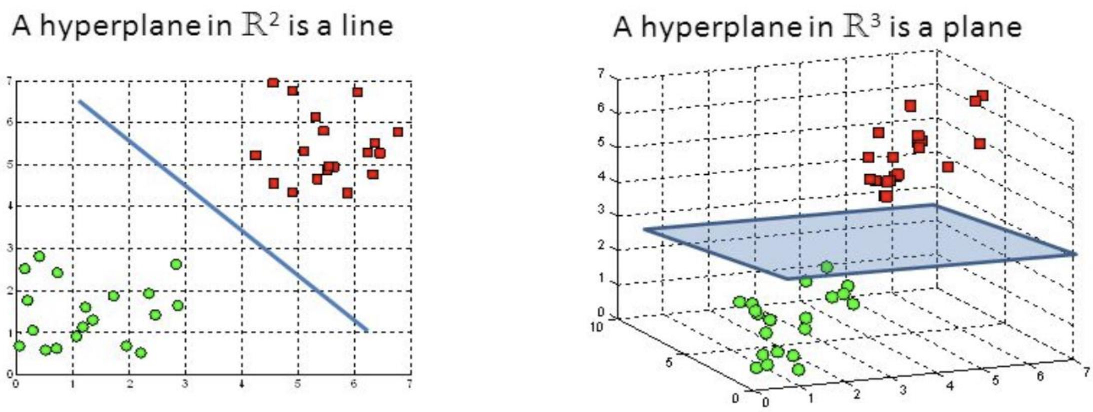

Figure 15: Hyperplane in 2D and 3D space

Now, to separate two classes of data points, there are many possible hyperplanes that could be chosen. Our objective is to find a plane that has the maximum margin i.e. the maximum distance between data points of both classes and below figure clearly explains this fact

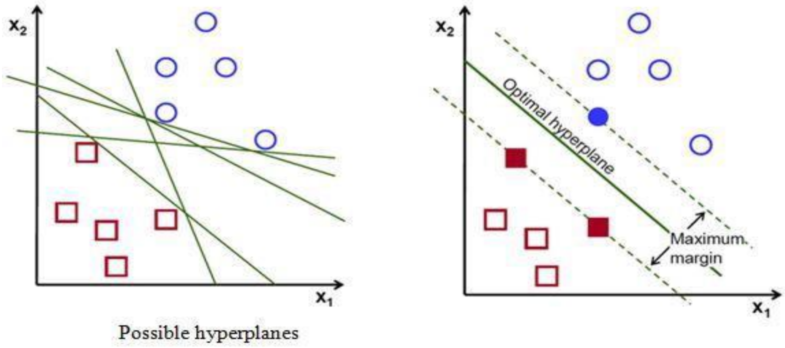

Figure 16: Possible Hyperplanes and Hyperplane with maximum margin

Hyperplane with maximum margin looks something like this in 3D space:

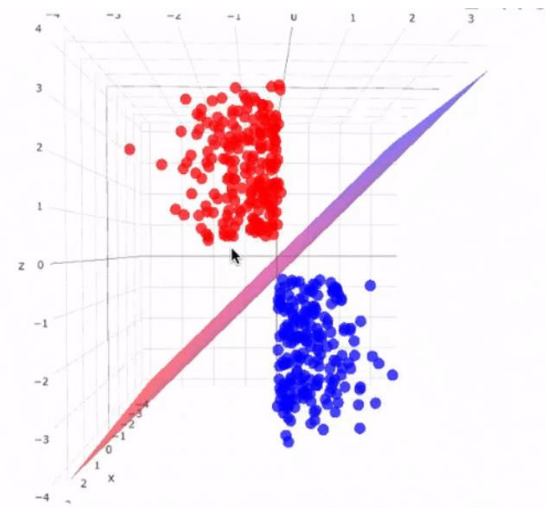

Figure 17: Visual representation of Hyperplane in 3D

The dimension of Hyperplane depends on the number of features.

a. Support Vectors

Support Vectors are the data points that are on or closest to the hyperplane and influence the position and orientation of the hyperplane. Using these support Vectors we maximize the margin of the classifier and deleting these support vectors will change the position of the hyperplane. These are actually the points that help us build SVM.

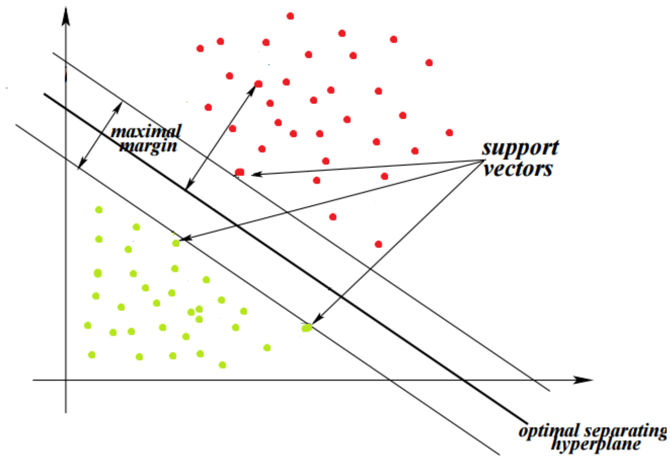

Figure 18: Support vectors

Support Vectors are equidistant from the hyperplane. They are called support vectors because if their position shifts, the hyperplane shifts as well. This means that the hyperplane depends only on the support vectors and not on any other observations.

SVM that we have discussed until now can only classify the data which is linearly separable.

If the data is non-linearly separated then,

For example: look at the below image where the data is non-linearly separated, of course, we cannot draw a straight line to classify the data points.

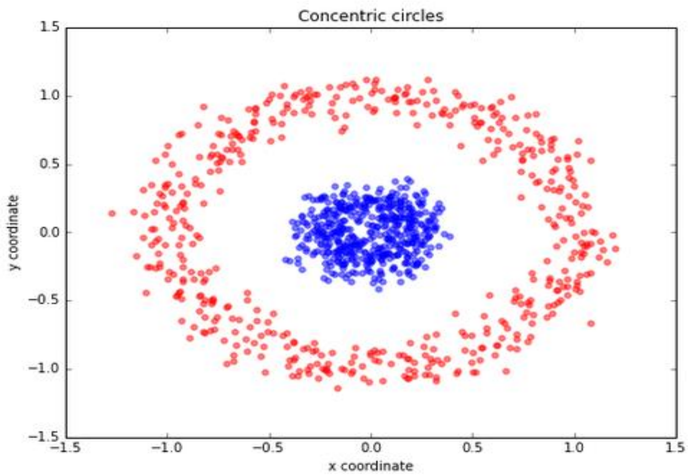

Figure 19: Non-linearly separated data

Here comes the concept of Kernel in SVM to classify non-linearly separated data. A kernel is a function which maps a lower-dimensional data into higher dimensional data.There are two ways by which kernel SVM will classify non-linear data[26].

- Soft margin
- Kernel tricks

It allows SVM to make a certain number of mistakes and keep the margin as wide as possible so that other points can still be classified correctly.

"In other words, SVM tolerates a few dots to get misclassified and tries to balance the tradeoff between finding the line that maximizes the margin and minimizes misclassification."

There are two types of misclassifications can happen:

- The data point is on the wrong side of the decision boundary but on the correct side
- The data point is on the wrong side of the decision boundary and on the wrong side of the Margin

b. Degree of tolerance

How much tolerance we want to set when finding the decision boundary is an important hyper-parameter for the SVM (both linear and nonlinear solutions). In Sklearn, it is represented as the penalty term — 'C'.

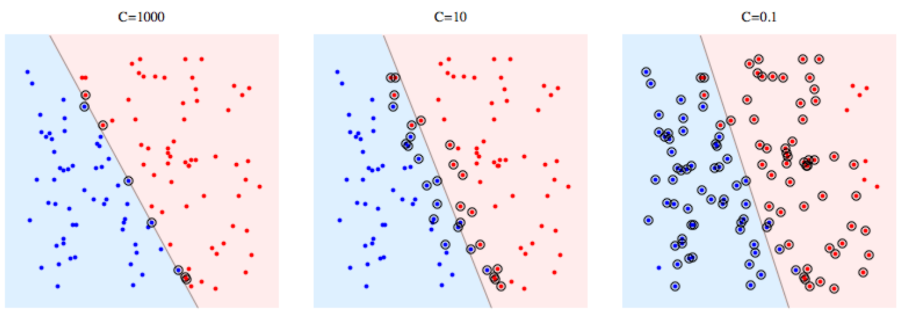

Figure 20: Degree of tolerance

The bigger the C, the more penalty SVM gets when it makes misclassification. Therefore, the narrower the margin is and fewer support vectors the decision boundary will depend on.

c. Kernel Trick

The idea is mapping the non-linear separable data from a lower dimension into a higher dimensional space where we can find a hyperplane that can separate the data points.

So it is all about finding the mapping function that transforms the 2D input space into a 3D output space and to reduce the complexity of finding the mapping function SVM uses Kernel Functions[26].

Kernel Functions are generalized functions that take 2 vectors(of any dimension) as input and output a score(dot product) that denotes how similar the input vectors are. If the dot product is small, vectors are different and if the dot product is large, vectors are more similar.

More formally,if we have data $x, z \in X$ and a map $\emptyset: X \to R^N$ then $k(x,z) = (\emptyset(x), \emptyset(z))$ is kernel function.

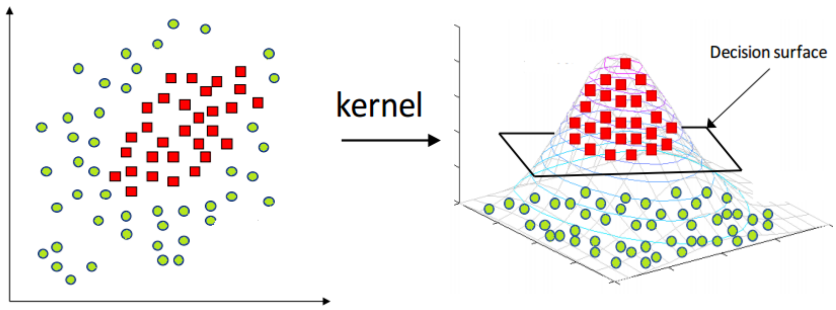

Figure 21: Pictorial representation of Kernel Trick

Types of Kernel Functions

- Linear
- Polynomial
- Radial Basis Function(rbf)
- Sigmoid

Let's talk about the most used kernel function i.e. Radial Basis Function (rbf).

Think of rbf as a transformer/processor to generate new features of higher dimension by measuring the distance between all other data points to a specific dot.

The most popular rbf kernel is Gaussian Radial Basis function. Mathematically:

Where gamma(***γ***) controls the influence of new features on the decision boundary. Higher the value of ***γ***, more influence of features on the decision boundary.

Similar to Regularization parameter/penalty term(C) in the soft margin, Gamma(γ) is a hyperparameter that can be tuned when we use kernel trick.

2.2.4 Random Forest Classifier

Random forest classifier (RFC) is a combination or ensemble of different decision tree predictors wherein the output of each predictor depends on feature vectors that are sampled independently . The classifier is called random forest as it generates forests with random amounts of trees. Normal decision trees are based only on rules for prediction of any outcome. However, random forest classifier use information gain in order to split features from a particular

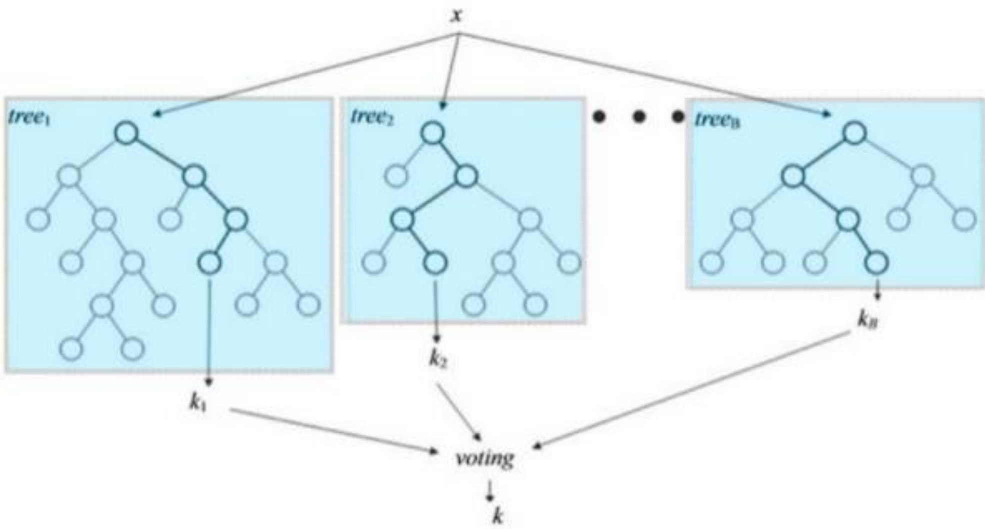

Figure 22: Random Forest Classifier

It usually starts with the creation of many random decision trees where each one of those predicts a class from the features that are provided to it. These results are then used to get a majority vote to predict final class.

2.2.5 K-Nearest neighbors

K-Nearest neighbors is an algorithm that is used for classification that is based on the similarity of its neighbors. It is called lazy since there is no training that is involved, instead all of the data points will only be used at the time of making a prediction. The data that the KNN classifier gets can be scalars or even multidimensional vectors. All of the data points are assumed to be in the feature space, as a result of which there is a notion of distance. A commonly used metric for calculating distance between data points is Euclidean distance. KNN can be used for binary classification as well as multi-label classification. A number K needs to be specified in order to decide the number of neighbors that will effectively influence the decision[27].

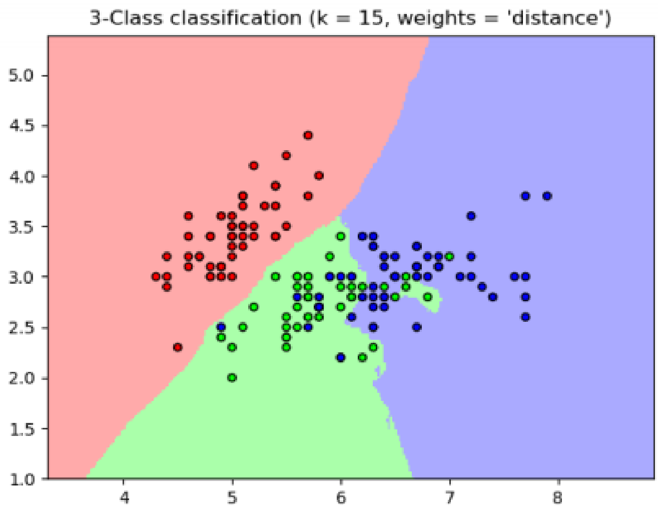

Figure 23: K-Nearest Neighbor Classifier

❖ **Standardization**

When independent variables in training data are measured in different units, it is important to standardize variables before calculating distance. For example, if one variable is based on height in cms, and the other is based on weight in kgs then height will influence more on the distance calculation. In order to make them comparable we need to standardize them which can be done by any of the following methods [28]:

$$X_s = \frac{X - mean}{s.d.} \tag{2.26}$$

$$X_s = \frac{X - mean}{max - min} \tag{2.27}$$

$$X_s = \frac{X-min}{max-min} \tag{2.28}$$

2.3 Unsupervised learning

Unsupervised learning is the training of machine using information that is neither classified nor labeled and allowing the algorithm to act on that information without guidance. Here the task of machine is to group unsorted information according to similarities, patterns and differences without any prior training of data.

Unlike supervised learning, no teacher is provided that means no training will be given to the machine. Therefore machine is restricted to find the hidden structure in unlabeled data by our-self.For instance, suppose it is given an image having both dogs and cats which have not seen ever.

Unsupervised learning classified into two categories of algorithms:

- Clustering: A clustering problem is where you want to discover the inherent groupings in the data, such as grouping customers by purchasing behavior.
- Association: An association rule learning problem is where you want to discover rules that describe large portions of your data, such as people that buy X also tend to buy Y.

2.3.1 Data Compression

Even with major advances over the past decade in computing power and storage costs, it still makes sense to keep your data sets as small and efficient as possible. That means only running algorithms on necessary data and not training on too much. Unsupervised learning can help with that through a process called dimensionality reduction.

Dimensionality reduction (dimensions = how many columns are in your dataset) relies on many of the same concepts as Information Theory: it assumes that a lot of data is redundant, and that you can represent most of the information in a data set with only a fraction of the actual content. In practice, this means combining parts of your data in unique ways to convey meaning. There are a couple of popular algorithms commonly used to reduce dimensionality ***[29].***

2.3.2 Principal Component Analysis

Principal component analysis (PCA) is a technique that is used to transform a number of different and possible uncorrelated variables into a smaller set of variables that are uncorrelated. It is a technique that uses variance in order to determine whether a certain feature is of interest to us or not, with the aim of finding vectors in the feature space that capture the maximum variance in the data.

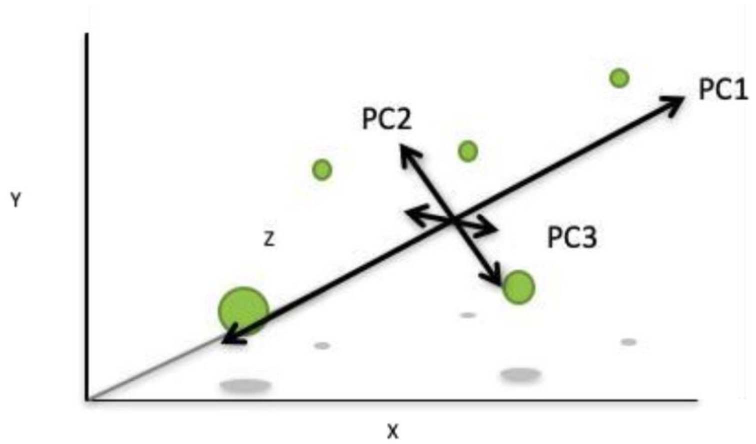

Figure 24: Dimensionality reduction with PCA

PCA starts with standardization of the matrix. This is done because the values in certain columns might have values that are higher than the values in other attributes. This might cause such attributes to dominate the entire principal component matrix . In order to achieve such standardization, by subtracting the mean of each column value from each of the attribute column and Next step in PCA is to obtain the covariance matrix from the standardized matrix. This is achieved by multiplying the standardized matrix (X) with its transform (XT) and normalizing it by **1/n-1** where n is the total number of features that present in the standardized matrix.[30]

Following the covariance matrix calculation, the eigenvectors of the covariance matrix is calculated. This results in an eigenvector matrix which contains columns in descending order which correspond to the first principal component, the second principal component and so on.Obtaining the PCA of a particular dataset, allows us to capture the maximum variability in the data without any loss of information . PCA transformation is also a pretty convenient way of achieving dimensionality reduction since it extracts all of the meaningful feature information without us having to provide any additional information regarding the data source or domain knowledge regarding the problem that we are trying to solve[29][30].

2.3.3 Confusion Matrix

A confusion matrix is a summary of prediction results on a classification problem. The number of correct and incorrect predictions are summarized with count values and broken down by each class. This is the key to the confusion matrix. The confusion matrix shows the ways in which your classification model is confused when it makes predictions. It gives us insight not only into the errors being made by a classifier but more importantly the types of errors that are being made.

	Class 1	Class 2
Class 1	TP	FN
Class 2	FP	TN

Table 2: Class confusion matrix

- Class 1 : Positive
- Class 2 : Negative

a. Definition of the Terms:

- Positive (P) : Observation is positive
- Negative (N) : Observation is not positive
- True Positive (TP) : Observation is positive, and is predicted to be positive.
- False Negative (FN) : Observation is positive, but is predicted negative.
- True Negative (TN) : Observation is negative, and is predicted to be negative.
- False Positive (FP) : Observation is negative, but is predicted positive.

b. Performance evaluation

Classification Rate or Accuracy is given by the relation[30]:

(2.29)

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

However, there are problems with accuracy. It assumes equal costs for both kinds of errors. A 99% accuracy can be excellent, good, mediocre, poor or terrible depending upon the problem.

c. Recall

$$Recall = \frac{TP}{TP + FN} \tag{2.30}$$

Recall can be defined as the ratio of the total number of correctly classified positive examples divide to the total number of positive examples. High Recall indicates the class is correctly recognized small number of FN).

d. Precision:

$$Precision = \frac{TP}{TP + FP} \tag{2.31}$$

To get the value of precision we divide the total number of correctly classified positive examples by the total number of predicted positive examples. High Precision indicates an example labelled aspositive is indeed positive (a small number of FP)[30].

e. High recall, low precision:

This means that most of the positive examples are correctly recognized (low FN) but there are a lot of false positives.

f. Low recall, high precision:

This shows that we miss a lot of positive examples (high FN) but those we predict as positive are indeed positive (low FP)

F-measure:

Since we have two measures (Precision and Recall) it helps to have a measurement that represents both of them. We calculate an F-measure which uses Harmonic Mean in place of

Arithmetic Mean as it punishes the extreme values more.

The F-Measure will always be nearer to the smaller value of Precision or Recall.

(2.32)

$$F - measure = \frac{2 * Recall * Precision}{Recall + Precision}$$

2.4 Online and offline recognition

HAR system can also be classified into two main groups depending on the response time that systems take to perform activity classification online methods aim for real-time prediction of activities while, conversely, offline methods usually need extra processing time***[31]***.

2.4.1 Online recognition

a. Vigilante

Vigilante is a mobile application developed for real-time human activity recognition [9] and it uses android platform. A hardware is a chest sensor strap, which can measure acceleration and physiological characteristics. Three ambulation activities with C4.5 decision tree classifier, gave an overall accuracy of 92.6%. The response time of the new gesture classification is 8% of the window length e. Evaluation results were with different users, without the need of training for each user.System uses permanent Bluetooth to communicate between the sensor and the mobile device, which makes it moderately energy efficient[31].

b. ActiServ

ActiServ is an activity recognition service for mobile phones. They make use of only the accelerometer sensor from the mobile phone. Using fuzzy inference system, they classify ambulation
and phone activities. ActiServ is energy efficient system and portable, there is no use of external sensor involved.

c. Brezmes

It was proposed for HAR, featuring a mobile application. They have used the k-nearest neighbors' classifier. This is computationally expensive and is not suitable to scale in smart phones, since it need the complete data set, which can be huge. Besides the system requires each user to provide his own data for better accuracy.

2.4.2 Offline recognition

a. Bao

Bao introduced a system that classifies twenty daily activities and ambulation, such as watching TV, vacuuming and scrubbing. A total of 5 accelerometers were used to collect the acceleration data, located at subject's arm, knee, ankle and hip. A 5% reduce accuracy is reported with only two accelerometers.

b. Khan

Khan introduced a system that classifies ambulation, activities and transitions between such as watching TV, vacuuming and scrubbing. A single accelerometer is placed on the subject's chest and the data is transferred to the computer via Bluetooth protocol. The sampling rate of the accelerometer is 20Hz.

2.5 CONCLUSION

In this chapter we described the background required to contextualize the problem of HAR. We first covered the techniques of artificial intelligence our research work is targeting to:Finding the best classifier to our moel. Then, we explored how dimensiality reduction can help evaluationg the model ,and ending with the offline and online recognition.We focused on SVM since it is the one that we are using in the implementation in the next chapter

3.1 INTRODUCTION

The objective of this study is to analyse a dataset of smartphone sensor data of human activities of about participants and try to analyse the same and draw insights and predict the activity using Machine Learning. We also try to detect if we could identify the participants from their walking styles and try to draw additional insights in highly sensitive and secure workplaces etc.

3.2 DATA CAPTURE OVERVIEW

3.2.1 Data collection:

Jorge Luis Reyes Ortiz had developed an android application called MotoLog ,to capture and store signals coming from the inertial sensors (accelerometer and gyroscope) In log files, while volunteers were performing the protocol. This application is capable to capture signal's values at each 0.02 second (frequency=50hz) 50 values of each axial inertial signal per second.

The experiments were carried out with a group of 30 volunteers within an age bracket of 19-48 years. They performed a protocol of activities composed of six basic activities: three static postures (standing, sitting, lying) and three dynamic activities (walking, walking downstairs and walking upstairs). The experiment also included postural transitions that occurred between the static postures. These are: stand-to-sit, sit-to-stand, sit-to-lie, lie-to-sit, stand-to-lie, and lie-to-stand. All the participants were wearing a smartphone (Samsung Galaxy S II) on the waist during the experiment execution. We captured 3-axial linear acceleration and 3-axial angular velocity at a constant rate of 50Hz using the embedded accelerometer and gyroscope of the device.

Raw triaxial signals cannot be fed directly to machine learning models. A signal processing pipeline should be built to filter noise, extract useful and clean signals splitting them into windows. From each window a vector of features is generated to obtain a classical dataset. The target column will contain activity labels associated to each vector of features(window). Recognizing the activity associated to each vector is a multiclass classification problem. To solve it, I intend to use some supervised learning classifiers (since each vector has an activity ID label) and compare predictions.

For the application,the data set was divided in two types:

- Data set type I: contain basic activities (UCI HAR DATASET).
- Data set type II:contain basic activities and postural transitions (HAPT DATASET)

After the processing each dataset was saved in a separated folder since for the training we are using just the data set type I.The metdata is stored in sheet of a comma separated values (CSV)file,which contains Train and test files for DATA SET TYPE I and II as well as the corresponding label for it.

3.2.2 Activities Labelling:

a.*General Visualizations:*

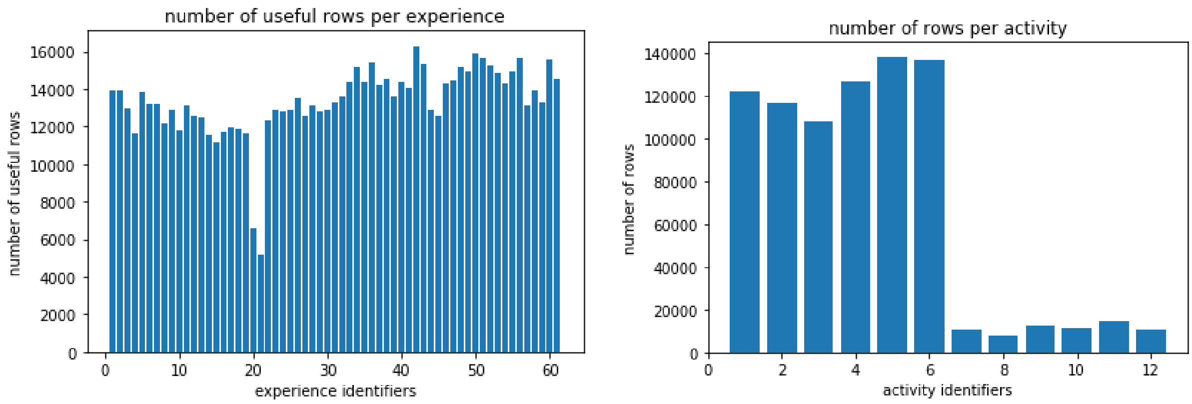

Figure 25: Number of rows per activity,number of useful rows per activity.

In this figure It appears clearly that number of captures is between 11000 and 16000 rows for the majority of experiences .Except The two experiences 20 and 21 ,they have less 8000 captures each. The figure bellow shows the number of rows per activity in all records. Basic activities (activity Ids from 1 to 6) have the majority of rows since they have the first priority in this study.

Postural Transitions have low number of rows due to the short duration of the transition from a static state to anther static state.Basic Activities contains at least more 10000 capture per second for all users which I think is enough in terms of data points number will be fed to machine learning algorithms. Basic activities columns are approximately balanced if we ignore postural transitions(from 7 to12).

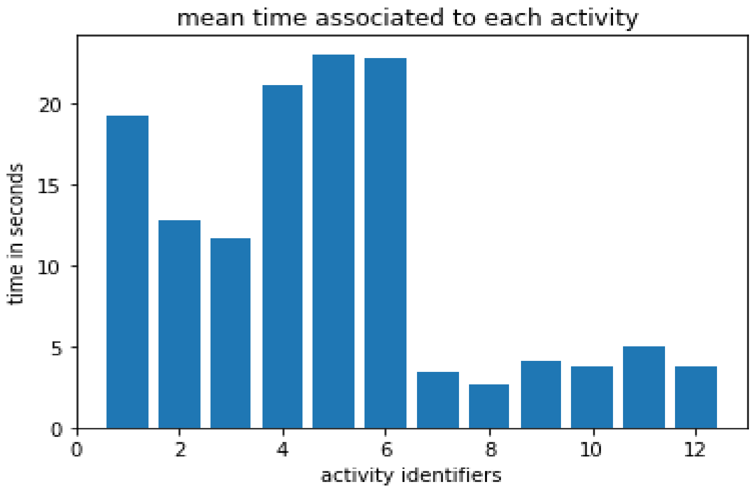

Figure 26: Mean time associated to each activity.

From the figure above, it appears that:

- Activities: 1(walking) ,4 , 5 and 6 (are static postures) have a mean duration near to 20s.
- Activities 2 and 3 (walking upstairs and Walking down stairs) have approximately 12s as a mean duration due to the short path of upstairs.
- For postural transitions: all postural transitions durations are less than 5 seconds they have approximately the same duration mean.

a. Detailed visualizations:

The figure bellow represents acceleration signals of the 1st experience:

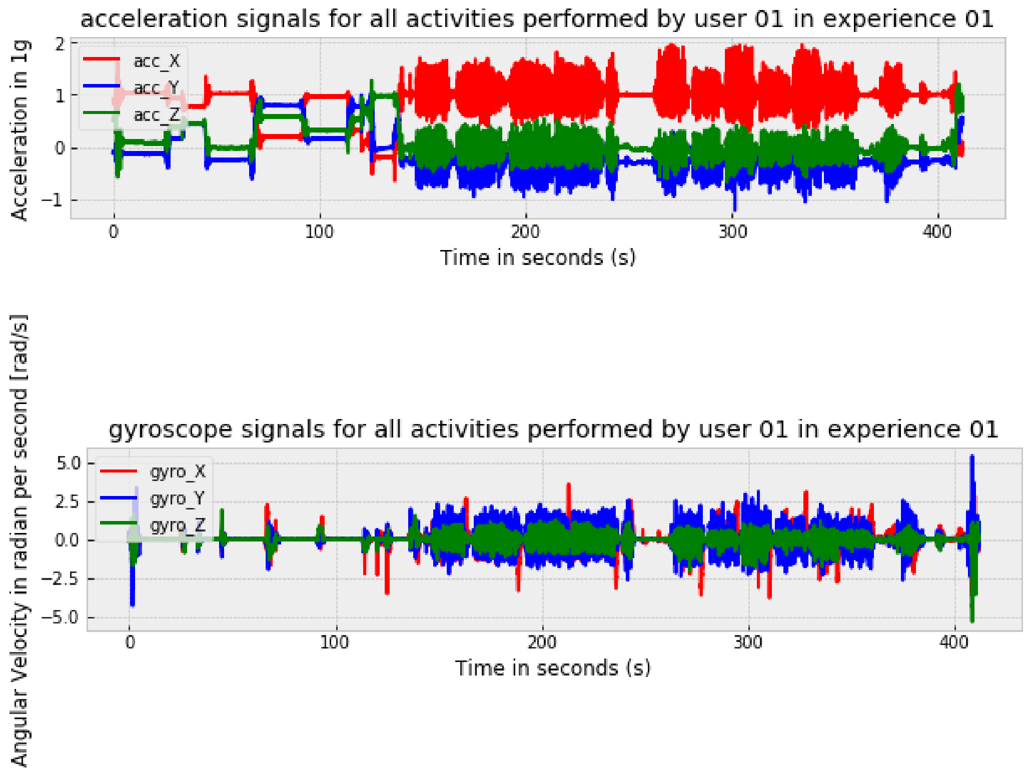

Figure 27: Acceleration and gyroscope signals of the 1st experience.

Acc values values varie approximately from -1g to 2g. During static postures acc signals should be constant. The figure bellow shows the gyro signals variation during the same experiment. Gyro values are approximately between -4 rad/second and 4 rad/second

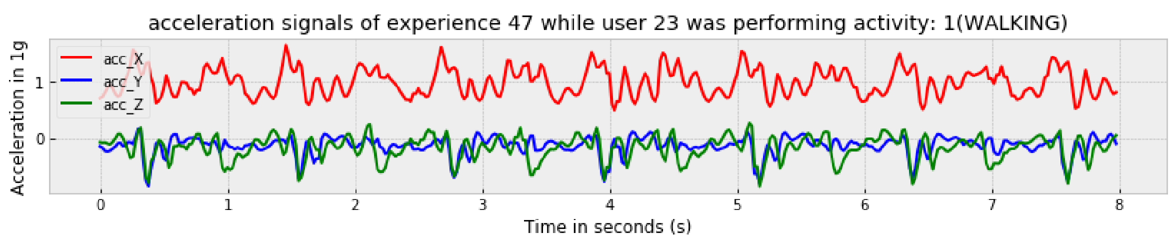

Figure 28: Acc signals captures while the user 01 in experiment 01 was walking.

The figure above represents acc signals captures while the user 01 in experiment 01 was walking. Signals are periodic which reflects the periodicity of walking. Acc y and acc Z means are near or bellow 0g. acc_X mean is near to 1g this can be justified by the phone orientation.X axis of the phone have the same direction and the same orientation as the earth gravity force which is approximately equal to 1. Acc_Y and acc_Z are not perpendicular exactly on the gravity force during walking as a result the gravity force will have negative projections on these axes.

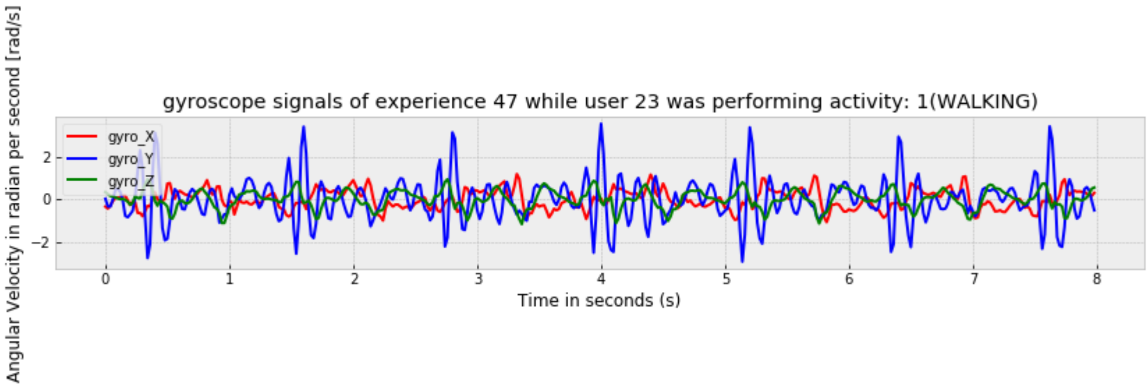

Figure 29: Gyro signals captures while the user 01 in experiment 01 was walking.

For gyro signals the figure above shows gyro signals related to this activity. X and Y axis mean values are near to 0. The angular velocity of Z axis varies periodically with a high speed from -3rad/s to 3 rad/s. The Z axis is the axis perpendicular on the phone screen it's normal that its angular velocity varies the most due the fact of moving legs during walking.For postural transitions the most important thing is the start and end points values of each acc signal. Since the gravity force projections on each axis will vary from static posture to another static posture.

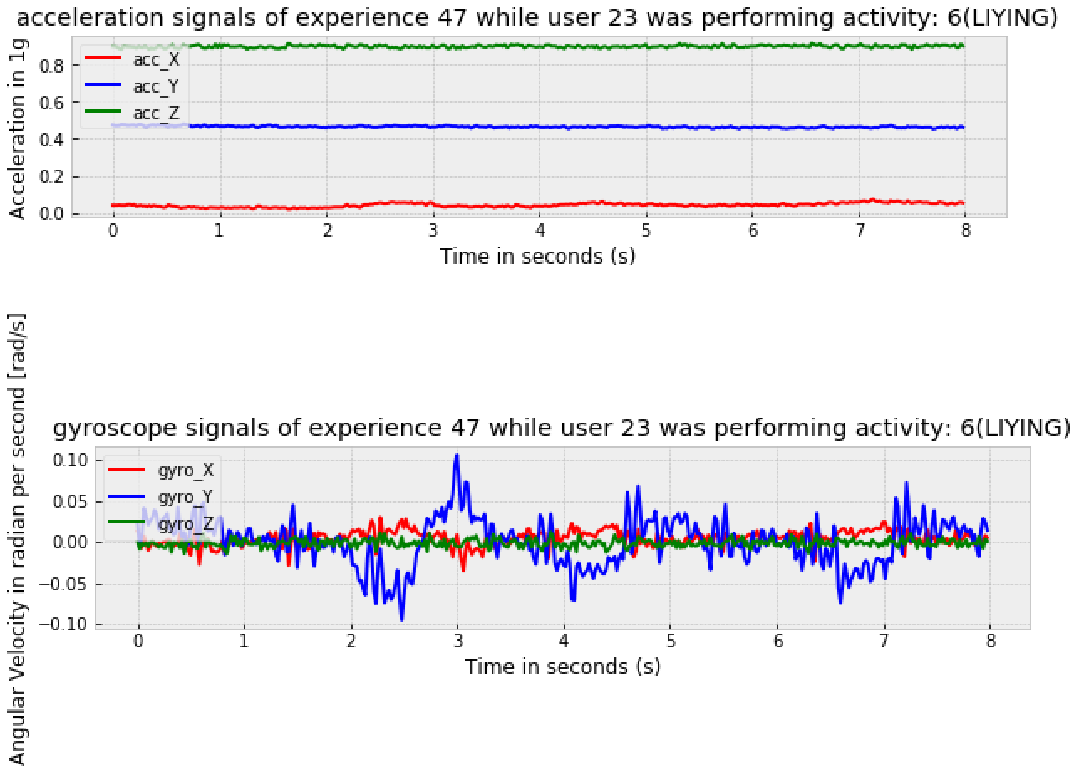

Figure 30: Acc and Gyro signals captures while the user 01 in experiment 01 was LAYING.

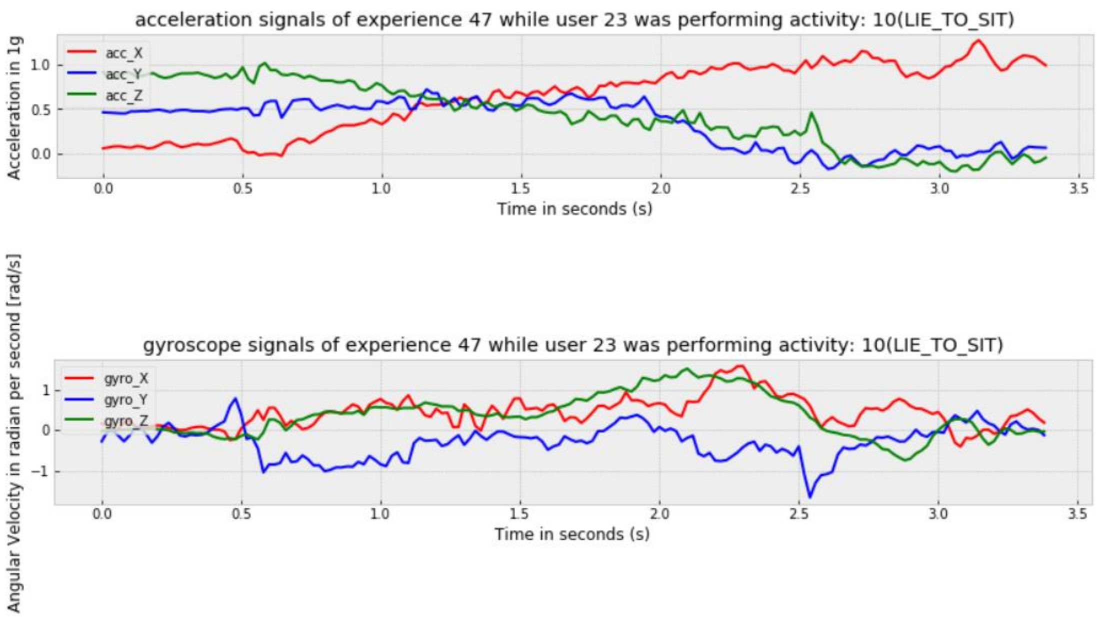

Figure 31: Acc and Gyro signals captures while the user 01 in experiment 01 (LIE TO SIT).

Gyro signals are hard to analyse during the transition but the start points values of all axis andend point values of all axis should be near to 0 rad/s since it is a transition from a static posture to another static posture.

3.3 SIGNAL PROCESSING

3.3.1 Preprocessing

The files contain raw and unfiltered data ,so processing is the first step to take to unsure that the data is filtred from any extra information that would degrade the model performance

a. Data Splitting:

The figure below show the structure of our data and how it is partitioned:

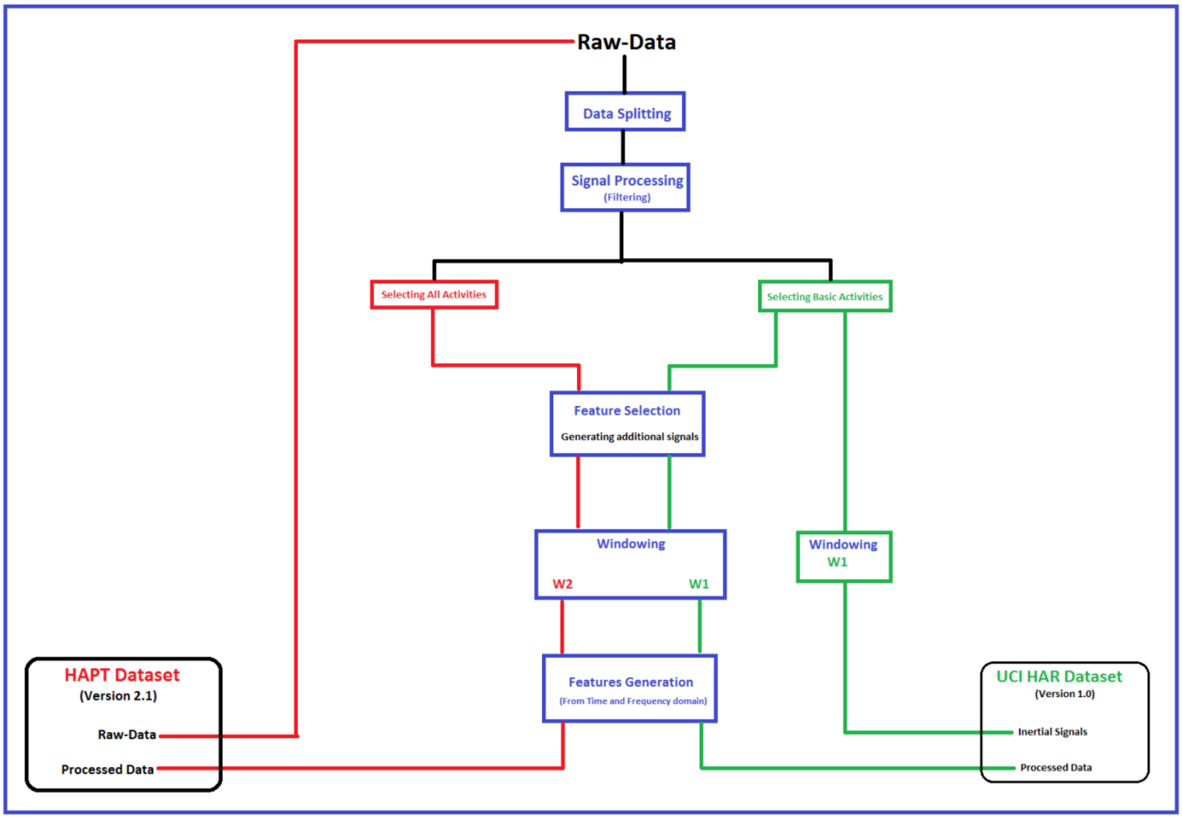

Figure 32: Data splitting

b. Noise filtering:

The features selected for this database come from the accelerometer and gyroscope 3-axial raw signals t_Acc-XYZ and t_Gyro-XYZ. These time domain signals (prefix 't' to denote time) were captured at a constant rate of 50 Hz. Then they were filtered using a median filter and a 3rd order low pass Butterworth filter with a corner frequency of 20 Hz to remove noise.

Median Filter: was applied to reduce background noise.

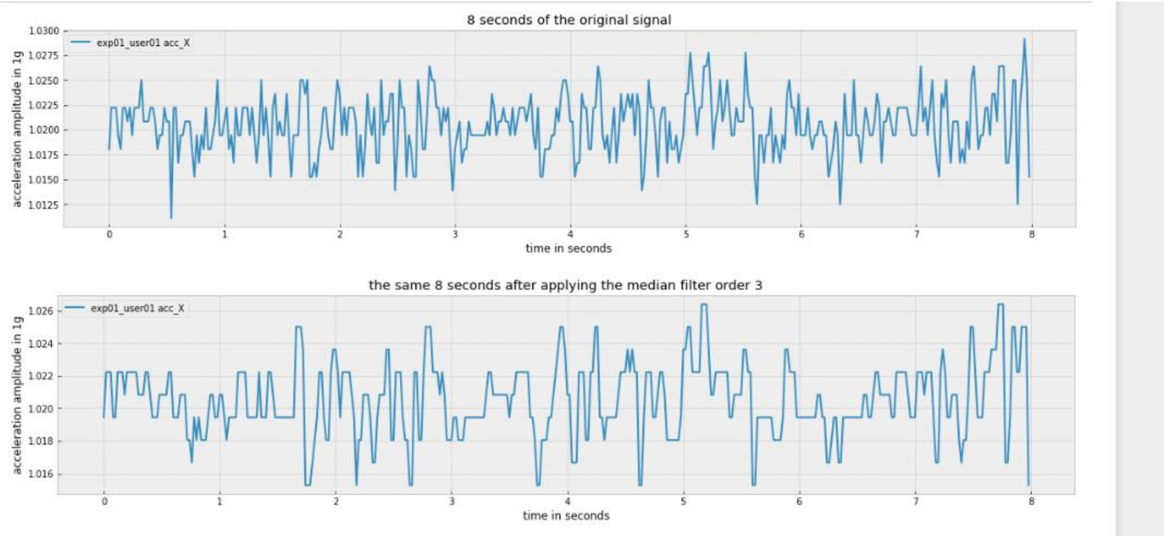

Figure 33: The same acc signal before and after applying Median filter

c . Magnitude

During the experiments the gravity magnitude (gravity of earth) is approximately constant since all experiments were carried out in the same place (laboratory). As a result, the gravity magnitude signal should be approximately equal to 1 for the majority of data points (since the unit used to measure acceleration is 1.

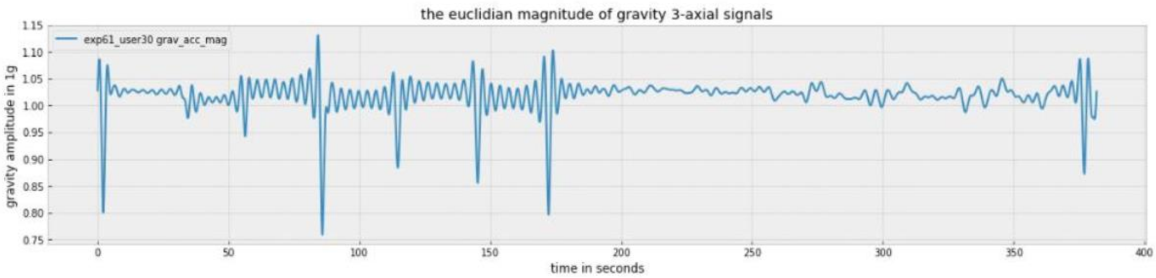

Figure 34: Magnitude

From the figure above, it appears clearly that grav_acc_mag is approximately near to 1g for the majority of data points. This proves that component selection was done correctly.

d. Windowing

The sensor signals window ere sampled in fixed-width sliding windows of 2.56 sec and 50% overlap (128 readings).Finally, the real value Fast Fourier Transformation (FFT) algorithm was used to transform these windows into the frequency domain, resulting in facc-xyz, fgyro-xyz, and faccjerk-xyz .The commonly used features are statistical measures including the mean, variance, standard deviation, root mean square fast Fourier transform (FFT), coefficients,and discrete cosine transform (DCT) coefficients.

After the signal processing was completed as described above, these signals were used as variables for estimating the feature vector of each mode, and the behavior was modeled using a windowed method (the average human walking rhythm is at least 1.5 steps/second);

3.4 EXPLORING UCI HAR DATASET

3.4.1 Distribution of Activities

We try to check the data volumes for each of the activities in order to see how balanced the data sets is with respect to the different activity labels. We find from the figure below :that the volumes are more or less well balanced:

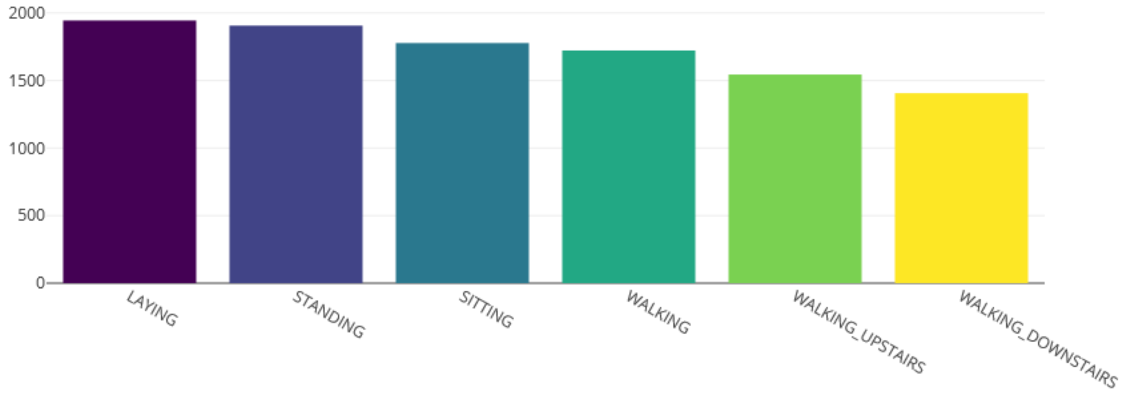

Figure 35: Distribution of Activities

Number of activity	Name of activity	Features
1	LAYING	1944
2	STANDING	1906
3	SITTING	1777
4	WALKING	1722
5	WALKING UPSTAIRS	1544
6	WALKING DOWNSTAIRS	1406

Table 3: Distribution of Activities

a. Separating activities using T-SNE:

t-SNE a non-linear dimensionality reduction algorithm finds patterns in the data based on the similarity of data points with features, the similarity of points is calculated as the conditional probability that a point A would choose point B as its neighbor.

It then tries to minimize the difference between these conditional probabilities (or similarities) in higher-dimensional and lower-dimensional space for a perfect representation of data points in lower-dimensional space.

The dataset is geared towards classifying the activity of the participant. Let us investigate the separability of the classes.

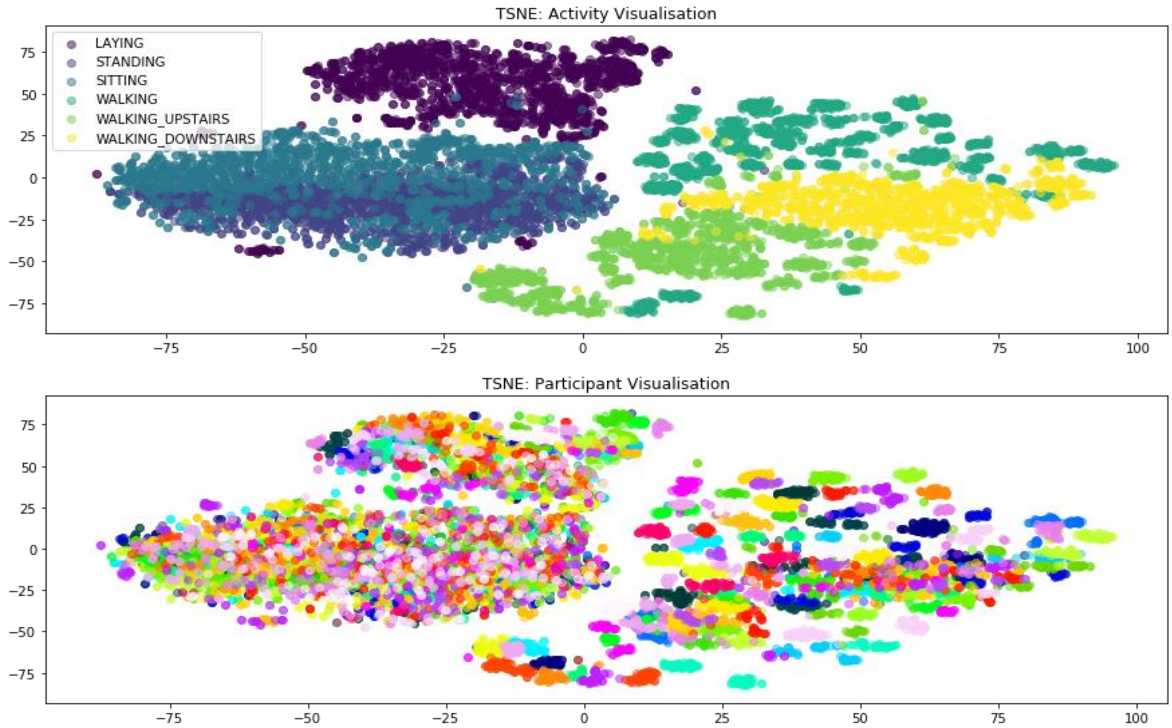

Figure 36: Separating activities using T-SNE

- n plot 1 we can clearly see the activities are mostly separable.
- Plot 2 reveals personal information of the participants. Everybody has for example a unique/sparable walking style (on the upper right). Therefore the smartphone should be able to detect what we are doing and also who is using the smartphone (if we are moving around with it).

Let us see how Good the Activities are Separable:

Without much preprocessing and parameter tuning a simple LGBM Classifier should work decently.

With a basic untuned model the activity of the smartphone user can be predicted with an accuracy of 95%.

This is pretty striking regarding six equally distributed labels.If the smartphone or an App wants to know what we are doing, this is feasible

3.4.2 Participant Exploration

a. How Good Are the Participants Separable

As we have seen in the second t-SNE plot the separability of the participants seem to vary regarding their activity. Let us investigate this a little bit by fitting the same basic model to the data of each activity separately.And see How long does the smartphone gather data for this Accuracy

Activity	Accuracy	Seconds
LAYING	0.648148	82.944000
STANDING	0.551363	81.322667
SITTING	0.557303	75.818667
WALKING	0.946636	73.472000
WALKING_UPSTAIRS	0.930052	65.877333
WALKING_DOWNSTAIRS	0.911932	59.989333

Table 4: Time to gather data

Detecting the correct participant regarding their current activity is not alone possible but astonishing accurate regarding the 30 different persons (94% by walking style).Noticable is that the accuracy seems to rise if the participant moves around. This implies a unique walking/movement style for each person.

The smartphone in gathering data,is quite fast (1 - 1.5 min).

b. Which Sensor Is More Important For Classifying Participants By Walking Style

We will fit another basic model to the walking data and investigate the feature importances afterwards. Since there are so many features we are going to group them by their sensor (accelerometer = Acc, gyroscope = Gyro)

The accelerometer supplies slightly more information. Both sensors are important for classification and refraining from using both sensors will be a drawback for the quality of the model.

We check between the Accelerometer and Gyroscope, how are they placed in the overall feature importance that is to say we sum up the feature importance of ever individual features from variables generated or computed from the two sensors and plot them to have a visual comparison. Fig. 37 shows the top ten features with most classification power and Fig 38.

Shows the comparative classification power between the two types of sensors used in the study , i.e. gyroscope and accelerometer.

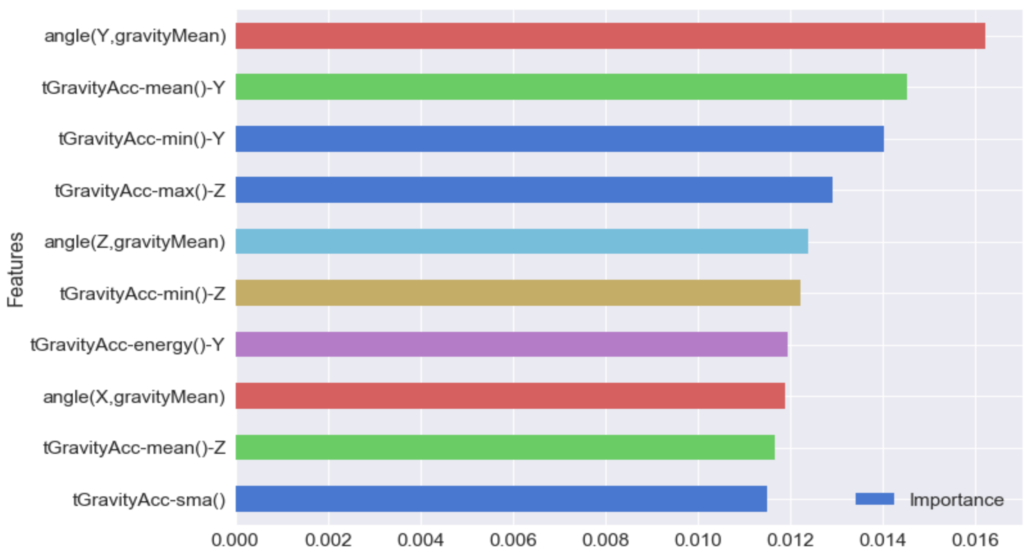

Figure 37: Comparative Sensor Importance for classification

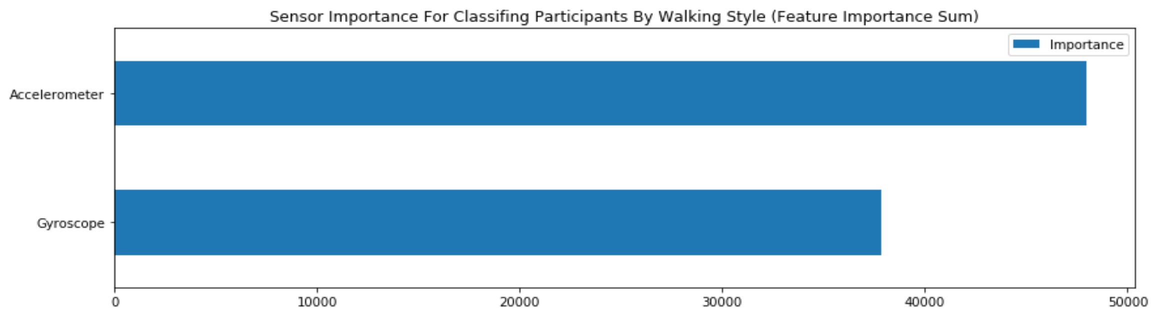

Figure 38: Comparative Sensor Importance for classification (Acc & gyro)

c. Participants walk analysis

In this study we capture the time involved in the activity 'Walking' for the different participants and we try to study the distribution to check for anomalies if any. We find from Fig. 39 that the data is distributed over a range. We assume that the test subjects had a fixed walk distance and the variations for the walk time are natural. We check this by checking (See Figure. 41) if the data follows a normal distribution as in most real world scenarios. We find that the data is close to normal distribution barring 2 or 3 outlier points.

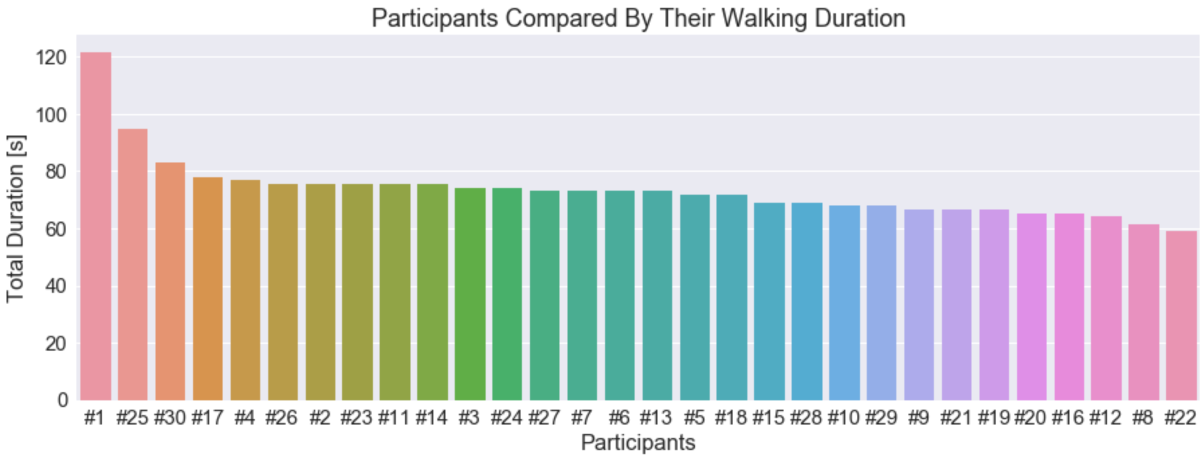

Figure 39: Walk time for 30 participants

d. How Long Does The Participant Use The Staircase

Since the dataset has been created in scientific environment nearly equal preconditions for the participants can be assumed. It is highly likely for the participants to have been walking up and down the same number of staircases. Let us investigate their activity durations.

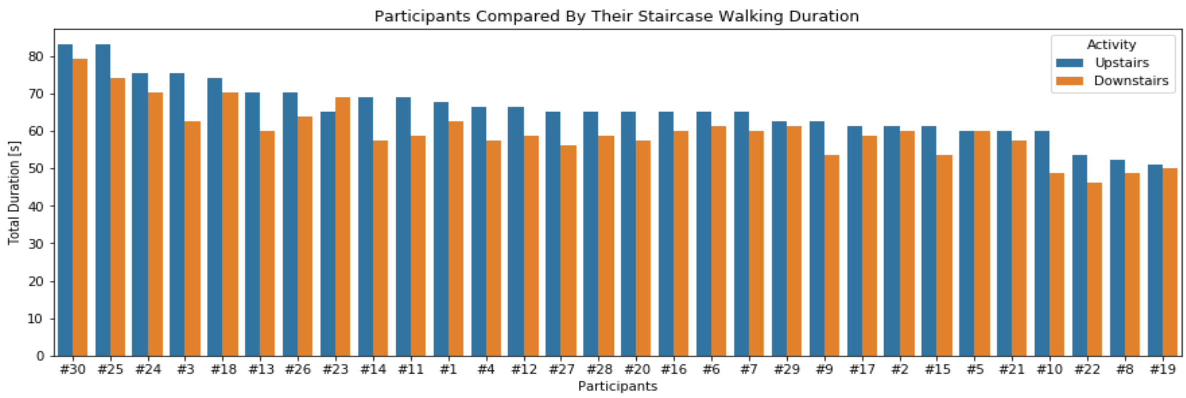

Figure 40: Participants compared by their staircase walking duration

Nearly all participants have more data for walking upstairs than downstairs. Assuming an equal number of up- and down-walks the participants need longer walking upstairs. Furthermore the range of the duration is narrow and adjusted to the conditions. A young person being ~50% fast in walking upstairs than an older one is reasonable.

Let us see if there are any conspicuities in the staircase walking duration distribution

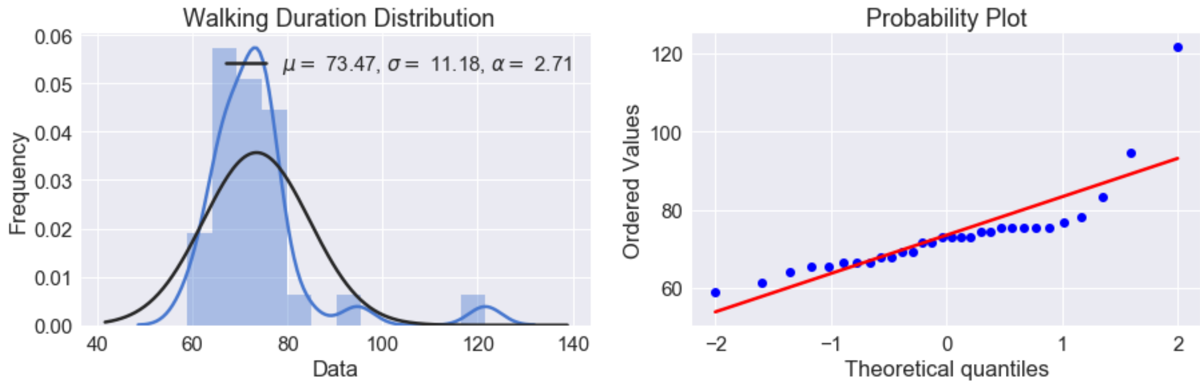

Figure 41: Normality check for the Walk data

Since the duration of each participant walking is distributed over a range we assume the participants had a fixed walking distance for their experiment rather than a fixed duration.

e. Participants Walk Analysis 2: Walk style analysis for each participant

In this study we try to capture the walk pattern or walk signature for each participant. This is an important feature to be studied, as walk patterns can be used to distinguish and with more data points even identify uniquely individuals or capture change in physical behaviour like signs of fatigue vs healthy walking etc. In order to plot the multivariate data in two dimensions, we reduce the data dimensions to just two using PCA and t-SNE (t Distributed Stochastic Neighbour Embedding) as shown in Fig. 42. We observe from the plots that each participant has a different walk pattern. We also note that for most of the patterns we can find two clusters. We can infer from the patterns as each cluster may denote a 'Walk experiment'.

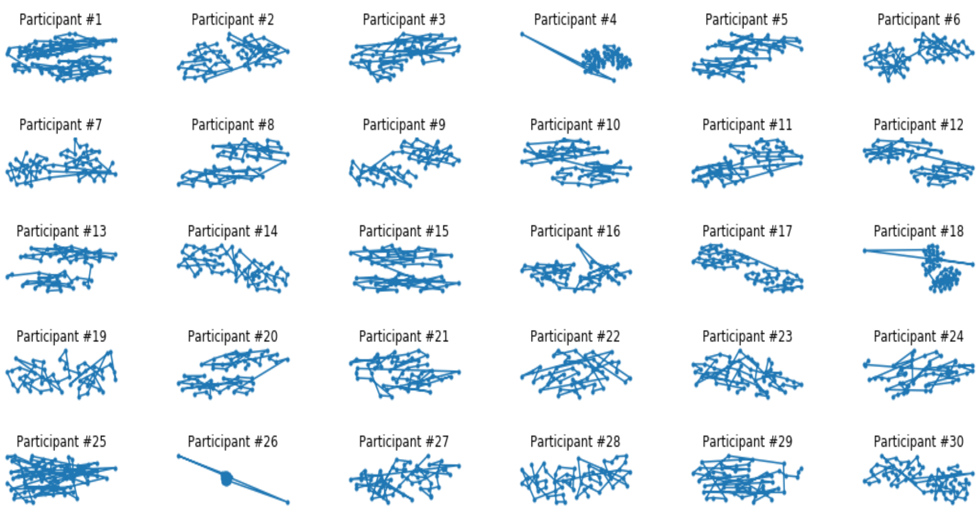

Figure 42: Walk patterns for individual participants

For this visualisation we are assuming the datapoints were not shuffled and are in the correct order (time series).
Visualising the walking structure for each participant we can see some outliers (e. g. #4, #18 and #26) in the data, which could be 'starting to walk', 'stopping' or 'stumble'. Additional there are two clusters for each participant. How these clusters should be interpreted is not clear.It cannot be the steps for each foot, since there would be connections between the clusters for each alternating step. Due to the fact that there is (mostly) only a single connection between the clusters and each cluster has just about the same size we conclude each cluster represents a single walking experiment.

f . Exploring Personal Information

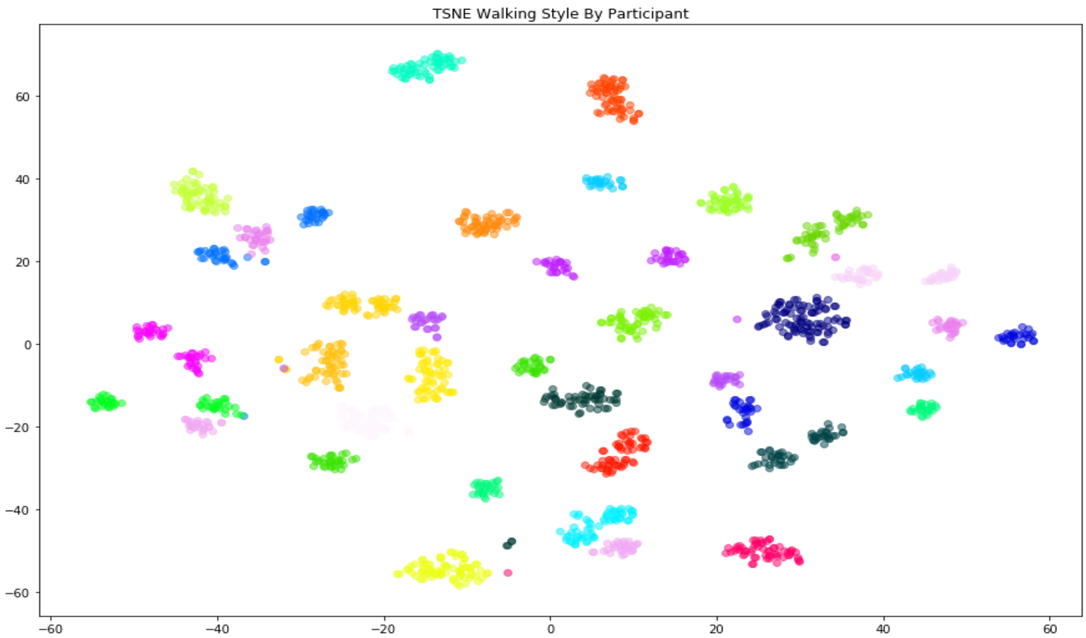

Figure 43: T-SNE walkig style by participant

For the decomposition we have only shown a single participant with the two walking styles. By computing a t-SNE visualization of the walking of all participants we can see two clusters for each participant (Some more distinct than others). Therefore the option of splitting the styles by decomposition suggests itself.

3.5 FEATURE EXTRACTION

The features seem to have a main name and some information on how they have been computed attached. Grouping the main names will reduce the dimensions for the first impression.

Now we that know the accelerometer signals for each activity repeat with a specific pattern over time. We can train supervised machine learning classifiers to predict or classify physical activity based on the accelerometer signals and various features extracted from them.As we daw above,twelve activities were preprocessed,here we are going to train the first six activities (UCI HAR dataset).

We explore the combine train and test data set and try to see how the data is distributed and the how separable are the participant and activity labels .First, we split the array of all features into training and test sets in an 80:20 ratio. The training set is used to train the classifier.The training set is randomized to avoid classifier getting biased to a particular pattern in time-series. The test set is used for testing the accuracy of the classifier.

For each record in the dataset it is provided:

- Triaxial acceleration from the accelerometer (total acceleration) and the estimated body acceleration.
- Triaxial Angular velocity from the gyroscope.
- A 561-feature vector with time and frequency domain variables.
- Its activity label.
- An identifier of the subject who carried out the experiment.

Primary attributes time domain frequency domain:

features	count
fBodyAcc	79
fBodyGyro	79
fBodyAccJerk	79
tGravityAcc	40
tBodyAcc	40
tBodyGyroJerk	40
tBodyGyro	40
tBodyAccJerk	40
tBodyAccMag	13
tGravityAccMag	13
tBodyAccJerkMag	13
tBodyGyroMag	13
tBodyGyroJerkMag	13
fBodyAccMag	13
fBodyBodyAccJerkMag	13
fBodyBodyGyroMag	13
fBodyBodyGyroJerkMag	13
angle	7
subject	1
Data	1

Table 5: Features

Mainly there are 'acceleration' and 'gyroscope' features. A few 'gravity' features are there as well.Impressive how many features there are in regard of the limited number of sensors used.

3.5.1 Statistically derived features

The set of variables that were estimated from these signals are:

1. mean(): Mean value
2. std(): Standard deviation
3. mad(): Median absolute deviation
4. max(): Largest value in array
5. min(): Smallest value in array
6. sma(): Signal magnitude area
7. energy(): Energy measure. Sum of the squares divided by the number of values.
8. iqr(): Interquartile range
9. entropy(): Signal entropy
10. arCoeff(): Autorregresion coefficients with Burg order equal to 4
11. correlation(): correlation coefficient between two signals
12. maxInds(): index of the frequency component with largest magnitude
13. meanFreq(): Weighted average of the frequency components to obtain a mean frequency
14. skewness(): skewness of the frequency domain signal
15. kurtosis(): kurtosis of the frequency domain signal
16. bandsEnergy(): Energy of a frequency interval within the 64 bins of the FFT of each window.
17. angle(): Angle between to vectors.

3.6 Selecting model

Modelling is choosing a classifier and then we dig into optimizing it.before we do that,we try 6 classifiers and select the the one who who got the most accuracy.

The classifiers are

- Decision tree classifier
- Random forest classifier
- Neighbours classifier
- Support vector machine
- Gaussian NB

- Logistic regression

The figure below show the accuracy of each model :

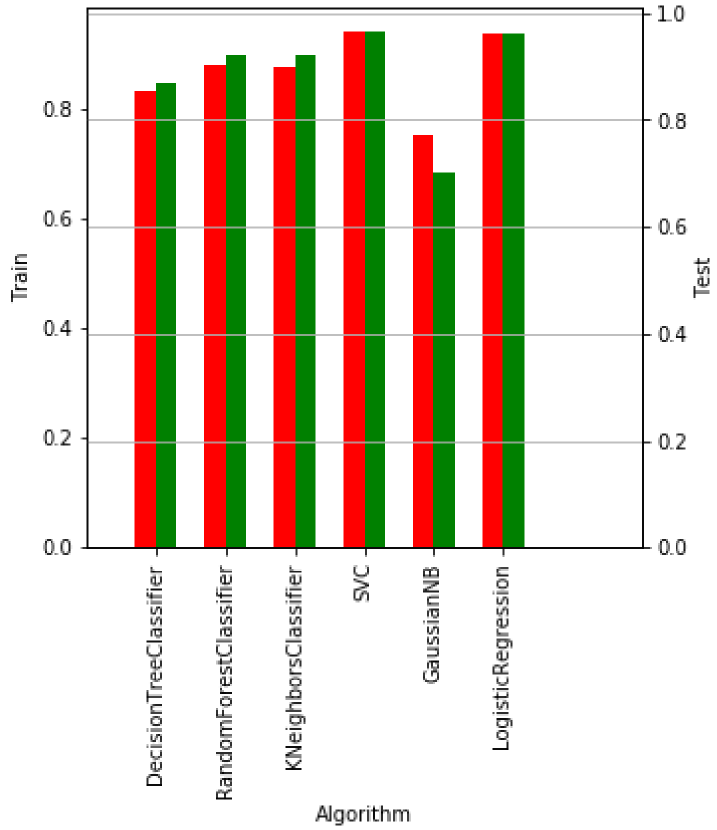

Figure 44: Recognition performance for different classifiers

Models: SVM give accuracy above 90%.

In real world, having domain knowledge is one of the most important aspects of machine learning Modelling. Here, we got pretty good accuracy of above 90%. This is very much due to the fact that features are very well engineered in signal processing part .

3.6.1 Model training

SVM is the machine learning algorithm of choice for prediction and detection tasks,using the scikit-learn python library.Recently,the support Vector Machine (SVM) paradigm has proved highly successful in a number of classification tasks.As classifier that discriminates the data by creating boundaries between classes rather than estimating class conditional densities,it may need considerably less data to perform accurate classification,and since it gave the most accurate model between the six classifiers here s other vadvantages:

- Versatile:different Kernel functions can be specified for the decision function.
- Effective in high dimentional spaces.
- Memory efficient.
- Best gamma,C and Kernel parameter combination is set using an exhaustive search over specified different values:
  C and gamma ={0.001,0.01,0.1,1,10,100,1000}
- Kernel ={rbf,linear}polynomial kernel was excluded due to very slow performance.

3.6.2 Feature selection

In order to prevent from high complexity and over fitting problem we need to choose only some of the given features. So the meaning of feature selection is to select a significant set of features which impact largely on the learning ability of machine learning algorithms. And feature extraction refers to the process through which we diminish the dimensionality of the available set of feature. For this we perform inter- feature transform to get a new dimensionally reduced representation. There are several feature selection methods,

3.6.3 Learning curve

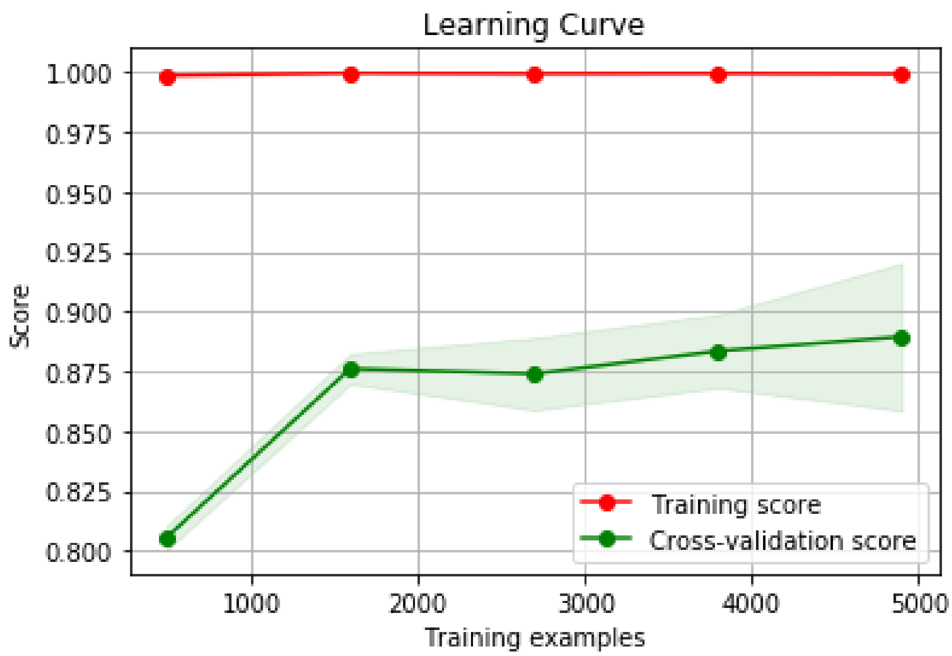

Figure 45: Learning curve

A learning curve shows the validation and training score of an estimator for varying numbers of training samples. It is a tool to find out how much we benefit from adding more training data and whether the estimator suffers more from a variance error or a bias error. Consider the figure above where we plot the learning curve of the SVM. for small amounts of data, the training score of the SVM is much greater than the validation score. Adding more training samples will most likely increase generalization.

3.6.4 Applying RFECV with svm classifier

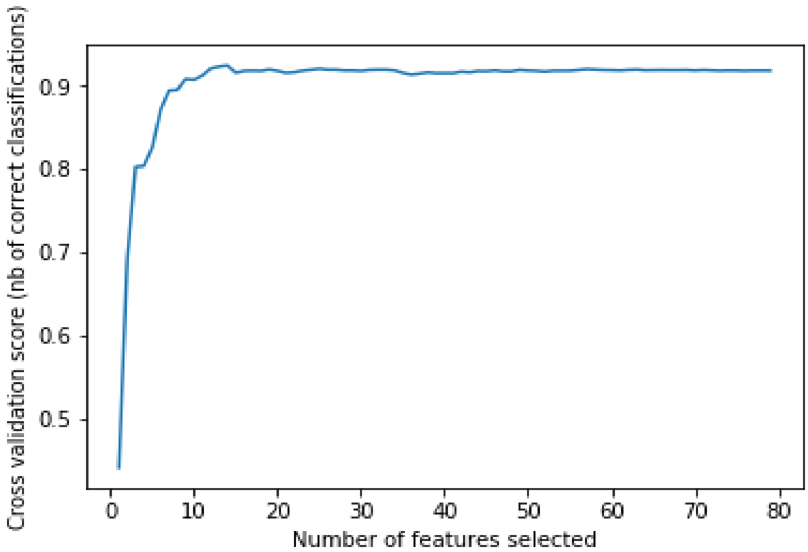

Figure 46: Applying RFECV with svm classifier

Given an external estimator that assigns weights to features.the goal of recursive feature elimination (RFE) is to select features by recursively considering smaller and smaller sets of features. First, the estimator is trained on the initial set of features and the importance of each feature is obtained either through a coef_ attribute or through a feature_importances_ attribute. Then, the least important features are pruned from current set of features. That procedure is recursively repeated on the pruned set until the desired number of features to select is eventually reached**.our optimal number of features is 14**.

- By appliying RFE with optimal number of features we get:

Train Accuracy	0.9254681389667219
Test Accuracy	0.8893790295215473

Table 6: Accuracy by applying RFE with optimal number of features

3.6.5 Variance Threshold

- Applying Variance Threshold method to remove low variance variable:

Train Accuracy	0.9239708915035661
Test Accuracy	0.8873430607397353

Table 7: Accuracy after applying Variance threshold

We see that the training accuracy has increased, to be become 92%

3.6.6 Dimension Reduction using PCA (Principal Component Analysis):

A vital part of using PCA in practice is the ability to estimate how many components are needed to describe the data. This can be determined by looking at the cumulative explained variance ratio as a function of the number of components:

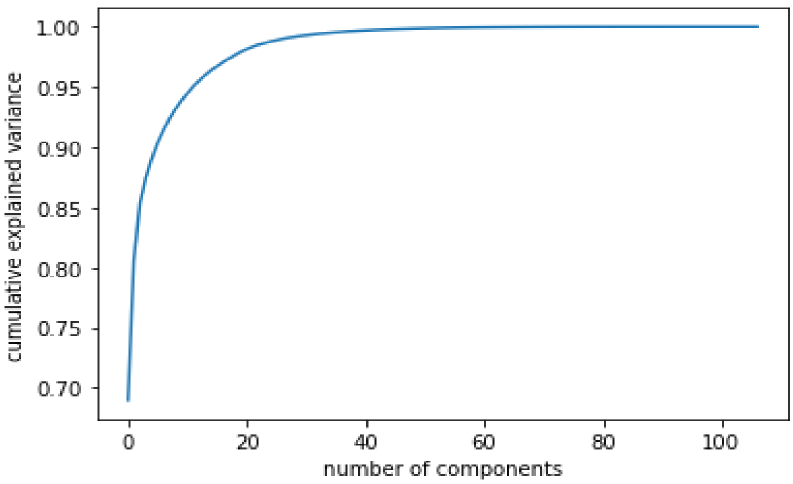

Figure 47: Dimension Reduction using PCA

This curve quantifies how much of the total, 110-dimensional variance is contained within the first N components. For example, we see that with the digits the first 14 components contain approximately 75% of the variance, while we need around 21 components to describe close to 100% of the variance.

a. Applying PCA with number of components=21

array([0.68980088, 0.8061375 , 0.85349808, 0.87469481, 0.89039493, 0.90395777, 0.91424124, 0.92372676, 0.93170012, 0.93846144,0.94480035, 0.95035127, 0.95528355, 0.96005964, 0.96374149,0.96717881, 0.97051228, 0.97356352, 0.97642399, 0.97898654,0.98131366])

Here we are visualizing top 2 principal components in scatter plot with data points segregated based on their activities

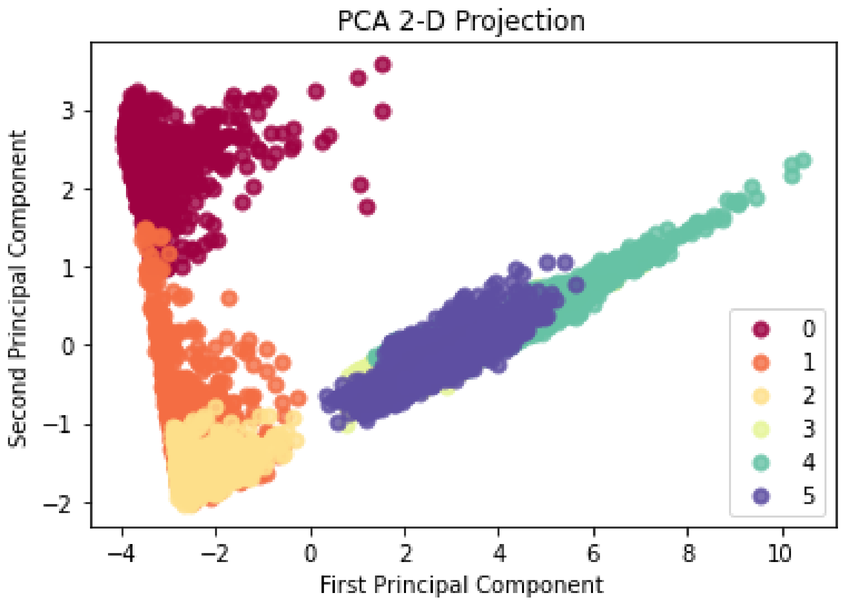

Figure 48: PCA 2D projection.

b. Checking Accuracy after applying PCA

Train Accuracy	0.9239708915035661
Test Accuracy	0.8873430607397353

Table 8: Accuracy after applying PCA.

We observe,after applying PCA the train and test accuracy has increased to 90%

3.6.7 Confusion matrix

To show how well each class performed during testing ,a confusion matrix is put in place.

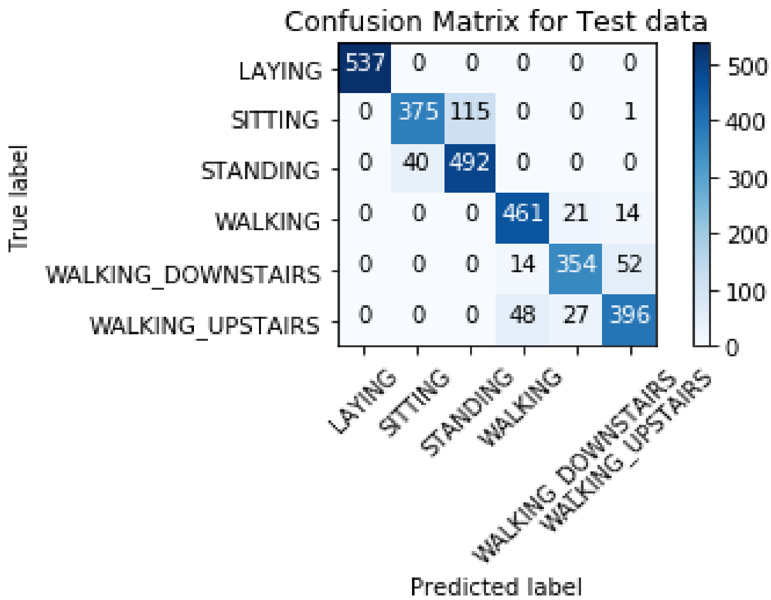

Figure 49: Confusion matrix

Accuracy: 0.887343,Which is :88%

We see that the darkest blue colours match the diagonal region with the 300to 500 range. Thus, predicted label has been equalized to the true label in most of the experimental cases.There is confusion between sitting and standing,and between walking downstairs and upstairs.

3.6.8 Parameter Tuning

Train Accuracy	0.9232939454418334
Test Accuracy	0.8883610451306413

Table 9: Accuracy after applying Parameter tuning.

By applying combination {'C': 1000, 'gamma': 0.001, 'kernel': 'rbf'} of SVM Linear kernel, it yielded the best accuracy.We conclude that if we want to trade off for faster training time.C:1000 and rbf kernel is a decent compromise.

With test and train accuracy almost equal to 90%, we are getting a maximum accuracy at this level.

3.7 CONCLUSION

In this case study we showcased how machine learning can help to study data from sensors which are already present in most smartphones maybe analysed to gain rich insights about the candidates studied. We can identify activities, classify or group participants to activities, get additional insights of activity durations and patterns of individuals involved in those activities. One can exploit the rich scope such insights has to offer in developing real time human asset monitoring in highly secured installations, tracking Elderly or population with movement disability or illness for any emergencies based on movement patterns, determining if a person id under fatigue or not and so on and so forth. The application domains are as broad as from healthcare to security services and fitness monitoring.

HAR is process of identifying activity. This activity could be fed as input to other monitoring or supporting applications.The sensor data, extracted features from data and classification models are main contributing elements in case of HAR. The classifiers plays important role in identification.we tried to find out the suitable models for HAR. We conclude that from the list of traditional supervised classifiers SVM performs better in case of HAR considering accuracy and execution time as its performance measuring criteria.

General conclusion

In this research project, we have presented the general architecture utilized to build human activity recognition systems and emphasized the design issues such as selection of sensors, obtrusiveness, flexibility, etc. which are independently evaluated based on the kind of system which is being developed. The paper further focuses on the importance of selecting important features from the data and provides a quantitative analysis of the metrics of execution time and accuracy.Classfication methods were utilized to select important features and were evaluated over four classification models. The results indicate that without a compromise in accuracy, the execution time and computational cost are greatly reduced with the use of feature selection methods. Better feature selection methods and improvement in tuning the parameters can assist further to improve accuracy and decrease computational cost.we experimented six static and dynamic activities (standing,sitting,lying,walking,walking upstairs,walking downstairs)and preprocessed additional static activities from the static ones.HAPT data in an extension of the UCI HAR data ,andwe discovered that adding more nformation about the transitional activities improvedwe have shown that SVM can reach the best recognition as a classifier.

References

[1] Labrador, M. and LARA, O., 2013. A Survey On Human Activity Recognition Using Wearable Sensors. IEEE Communications Surveys & Tutorials (Volume: 15 , Issue: 3 , Third Quarter 2013): IEEE, pp.1192 - 1209.

[2] Voicu, R., Dobre, C., Bajenaru, L. and Ciobanu, R., 2019. Human Physical Activity Recognition Using Smartphone Sensors. National Institute for Research and Development in Informatics, 011455 Bucharest, Romania.

[3] Attal, f., chamroukhi, f., Oukhellou, L. and Amirat, Y., 2015. Physical Human Activity Recognition Using Wearable Sensors. Vittorio M.N. Passaro, Academic Editor.

[4] bhat, L. and Nayak, M., 2013. MEMS Pressure Sensors-An Overview Of Challenges In Technology And Packaging.

[5] Bayat, A., Pomplun, M. and A.Tran, D., 2014. A Study On Human Activity Recognition Using Accelerometer Data From Smartphones. 34th ed. Department of Computer Science, University of Massachusetts, Boston, 100 Morrissey Blvd Boston, MA 02125, USA, pp.450-457.

[6] Dadafshar, M., 2014. ACCELEROMETER AND GYROSCOPES SENSORS: OPERATION, SENSING, AND APPLICATIONS. [online] Pdfserv.maximintegrated.com. Available at: <https://pdfserv.maximintegrated.com/en/an/AN5830.pdf> [Accessed 13 September 2020].

[7] ElProCus - Electronic Projects for Engineering Students. 2020. Gyroscope Sensor- Working, Types & Applications. [online] Available at: <https://www.elprocus.com/gyroscope-sensor/> [Accessed 13 September 2020].

[8] Lee, J. and Kim, J., 2016. Mobile Information Systems. Department of Computer Science and Engineering, Hanyang University, Ansan, Gyeonggi-Do 15588, Republic of Korea, p.12.

[9] En.wikipedia.org. 2020. Data Acquisition. [online] Available at: <https://en.wikipedia.org/wiki/Data_acquisition> [Accessed 15 September 2020].

[10] BULLING, A., BLANKE, U. and SCHIELE, B., 2014. A Tutorial On Human Activity Recognition Using Body-Worn Inertial Sensors. 46th ed. New York, p.33.

[11] Siirtola, P. and Roning, J., 2013. Ready-To-Use Activity Recognition For Smartphones. Oulu, Finland.

[12] Wannenburg, J. and Malekian, R., 2016. Physical Activity Recognition From Smartphone Accelerometer Data For User Context Awareness Sensing. IEEE.

[13] Wang, J., Chen, Y., Hao, S. and Peng, X., 2017. Deep Learning For Sensor-Based Activity Recognition: A Survey. Beijing, China.

[14] El Moudden, I. and Ouzir, M., 2016. Mining Human Activity Using Dimensionality Reduction And Pattern Recognition. 9th ed. Morocco.

[15] Caetano, C., de Melo, V., A. dos Santos, J. and Schwartz, W., 2020. Activity Recognition Based On A Magnitude-Orientation Stream Network. Brazil.

[16] Bland, M., 2006. Health Sciences M.Sc. Programme Applied Biostatistics Mean And Standard Deviation.

[17] Davide Anguita, D., Ghio, A., Parra, X. and L.Reyes-Ortiz, J., 2013. A Public Domain Dataset For Human Activity Recognition Using Smartphones. Bruges (Belgium).

[18] Chen, Y. and Shen, C., 2017. Performance Analysis Of Smartphone-Sensor Behavior For Human Activity Recognition. china: IEEE.

[19] Rosati, S. and Balestra, G., 2018. Comparison Of Different Sets Of Features For Human Activity Recognition By Wearable Sensors. Torino, Italy.

[20] Ahmad, T., Rafique, J., Muazzam, H. and Rizvi, T., 2015. Using Discrete Cosine Transform Based Features for Human Action Recognition. Journal of Image and Graphics, 3(2).

[21] Afzal Hossan, M., Memon, S. and Gregeory, M., 2014. A novel approach for MFCC feature extraction. [online] p.81. Available at: <https://ieeexplore.ieee.org/document/8186158> [Accessed 13 September 2020].

[22] Lisowski, E., 2020. Machine Learning Techniques And Methods. [online] Addepto. Available at: <https://addepto.com/machine-learning-techniques-and-methods/> [Accessed 14 September 2020].

[23] Brownlee, J., 2020. Supervised And Unsupervised Machine Learning Algorithms. [online] Machine Learning Mastery. Available at: <https://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms/> [Accessed 14 September 2020].

[24] Medium. 2020. A Brief Introduction To Supervised Learning. [online] Available at: <https://towardsdatascience.com/a-brief-introduction-to-supervised-learning-54a3e3932590> [Accessed 14 September 2020].

[25] Bartlett P. and Shawe-Taylor J., "Generalization performance of support vector machine and other pattern classifiers", In C. ~Burges B. ~Scholkopf, editor, "Advances in Kernel Methods-Support Vector Learning", pp. 43–55, MIT press, 1998.

[26] Burges C., "A tutorial on support vector machines for pattern recognition", In "Data Mining and Knowledge Discovery". Kluwer Academic Publishers, Boston, 1998, (Vol. 2), pp. 1–43.

[27] Sutton, O., 2012. Introduction To K Nearest Neighbour Classification And Condensed Nearest Neighbour Data Reduction.

[28] Bhalla, D., 2020. K Nearest Neighbor : Step By Step Tutorial. [online] ListenData. Available at: <https://www.listendata.com/2017/12/k-nearest-neighbor-step-by-step-tutorial.html> [Accessed 14 September 2020].

[29] Qi, W., Su, H., Yang, C., Ferrigno, G., De Momi, E. and Aliverti, A., 2019. A Fast and Robust Deep Convolutional Neural Networks for Complex Human Activity Recognition Using Smartphone. Sensors, 19(17), p.3731.

[30] Nair, A., 2019. HUMAN ACTIVITY RECOGNITION USING ACCELEROMETER DATA WITH MULTI CLASS SVM. International Journal of Engineering Applied Sciences and Technology, 04(04), pp.41-48.

[31] Badshah, M., n.d. Sensor - Based Human Activity Recognition Using Smartphones.

[32] Analog.com. 2020. [online] Available at: <https://www.analog.com/media/en/analog-dialogue/volume-40/number-4/articles/capacitance-sensors-for-human-interfaces-to-electronics.pdf> [Accessed 14 August 2020]. 1

[33] Bhalla, D., 2020. K Nearest Neighbor : Step By Step Tutorial. [online] ListenData. Available at: <https://www.listendata.com/2017/12/k-nearest-neighbor-step-by-step-tutorial.html> [Accessed 15 September 2020].

[34] Archive.ics.uci.edu. 2020. UCI Machine Learning Repository: Human Activity Recognition Using Smartphones Data Set. [online] Available at: <http://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones?spm=a2c4e.11153940.blogcont603256.23.333b1d6fYOsiOK> [Accessed 7 November 2020].