

# Human Activity Recognition Using Wearable Body Sensor By Machine Learning Approach

by

Sadia Tangim Promi

15301017

Md. Zahidur Rahman

15101122

Moumita Mostafa

15201023

Sarah Bintay Harun

14101067

A thesis submitted to the Department of Computer Science and Engineering  
in partial fulfillment of the requirements for the degree of  
B.Sc. in Computer Science and Engineering

Department of Computer Science and Engineering  
Brac University  
December 2019

© 2019. Brac University  
All rights reserved.

# Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

## Student's Full Name & Signature:

---

Sadia Tangim Promi  
15301017

---

Md. Zahidur Rahman  
15101122

---

Moumita Mostafa  
15201023

---

Sarah Bintay Harun  
14101067

# Approval

The thesis/project titled “Human Activity Recognition Using Wearable Body Sensor By Machine Learning Approach” submitted by

1. Sadia Tangim Promi(15301017)
2. Md. Zahidur Rahman (15101122)
3. Moumita Mostafa (15201023)
4. Sarah Bintay Harun (14101067)

Of Fall, 2019 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science and Engineering on December 25, 2019.

## Examining Committee:

Supervisor:  
(Member)

---

Dr. Md. Golam Rabiul Alam, PHD  
Associate Professor  
Dept. of Computer Science and Engineering  
BRAC University

Head of Department:  
(Chair)

---

Mahbubul Alam Majumdar, PHD  
Professor and Chairperson(CSE), Interim Dean,  
Department of Computer Science and Engineering  
Brac University

# Abstract

The prevalence of electronics devices and the increase in computer resources, like networking, storage, accessibility and sensor capacity, have significantly improved the lives of humans. Now a days most smart devices have a number of strong sensing equipment, such as sensors for movement, position, connection and direction. Basically, movement or motion tracking sensors are commonly been using to classify the physical activities of humans. This has opened entryways for a wide range of and intriguing applications with regards to a numerous zones, for example, human healthcare well being and transportation, security system. In this point of view, this research gives a complete, best in class audit of the present circumstance of human activity recognition (HAR) approaches with regards to inertial sensors in electronic portable smartphone devices. Our research started by analyzing the principles of human activities and the entire historical events based on electronics deices such a smartphone, which demonstrate the development in this area over the past few years. Our approach concentrates on the introduction of the means of HAR arrangements with regards to sensors. We propose a methodology which incorporates traditional signal processing techniques with deep learning tools to robustly classify activities from wearable body sensor data. Our proposed methodology achieves a validation accuracy of 96.26% in the WISDM Dataset and is able to recognize human activity from wearable body sensor data robustly.

**Keywords:** Human Activity Recognition; Machine Learning; ; Data set; Accelerometer; Prepossessing; convoluntional Neural Network;

# Dedication

We would like to dedicate this thesis to our loving parents, friends and everyone that helped us with this thesis.

## **Acknowledgement**

We would like to thank Dr. Md. Golam Rabiul Alam for agreeing to supervise us with our thesis. His patience and confidence in us have been a source of encouragement and this thesis would not have been possible without the great support and influence of him. We believe his dedication to this paper deserves to be reciprocated with great gratitude. A special thanks to the Thesis committee for taking the time to review and evaluate our thesis as part of our undergraduate program.

# Table of Contents

Declaration	i
Approval	ii
Ethics Statement	iii
Abstract	iii
Dedication	iv
Acknowledgment	v
Table of Contents	vi
List of Figures	viii
List of Tables	x
Nomenclature	xi
<b>1 Introduction</b>	<b>1</b>
1.1 Thoughts behind the Prediction Model . . . . .	1
<b>2 Related Work</b>	<b>3</b>
<b>3 Methodology</b>	<b>6</b>
3.1 Overview . . . . .	6
3.1.1 Python and Libraries . . . . .	8
3.2 Workflow . . . . .	9
3.3 Accelerometer . . . . .	10
3.4 Data Set Description . . . . .	11
3.5 Activities . . . . .	13
3.6 Research Development Using Python . . . . .	13
3.6.1 Data Pre Processing . . . . .	13
3.6.2 Plotting . . . . .	16
3.6.3 Data Pre Processing . . . . .	19
<b>4 Signal Processing for Feature Extraction</b>	<b>24</b>
4.1 Features . . . . .	24
4.2 Removing gravitational contribution from the signal . . . . .	25
4.3 Average value of Signal . . . . .	26

4.4	RMS value of Signal . . . . .	26
4.5	Autocorrelation Features . . . . .	26
4.6	Periodogram . . . . .	28
4.7	Welch's Power Spectral Density . . . . .	29
4.8	Spectral Peak Features . . . . .	30
4.9	Spectral Power Features . . . . .	30
4.10	Summary of Extracted Features . . . . .	31
<b>5</b>	<b>Exploratory Data Analysis</b>	<b>32</b>
5.1	Time Series Visualization . . . . .	32
5.2	Dimensionality Reduction . . . . .	35
5.2.1	PCA . . . . .	35
5.2.2	t-SNE . . . . .	37
5.2.3	LDA . . . . .	38
5.2.4	Summary of dimensionality Reduction plots . . . . .	39
<b>6</b>	<b>Proposed Pipeline</b>	<b>40</b>
6.1	Fusion of Features with Time-Series Data . . . . .	40
6.2	Convolutional Neural Network Architecture . . . . .	42
6.2.1	Inspiration . . . . .	42
6.2.2	Architecture Design . . . . .	42
6.3	Training and Results . . . . .	44
6.3.1	Loss Metric . . . . .	44
6.3.2	Optimizer . . . . .	44
6.3.3	Training . . . . .	45
6.3.4	Confusion Matrix . . . . .	46
<b>7</b>	<b>Conclusion</b>	<b>47</b>
7.1	Conclusion . . . . .	47
7.2	Future Scope . . . . .	47
	<b>Bibliography</b>	<b>51</b>



# List of Figures

3.1	Workflow Diagram . . . . .	9
3.2	Smartphone and accelerometer axis . . . . .	10
3.3	Data Set Text View . . . . .	12
3.4	Activity . . . . .	13
3.5	Parser Error . . . . .	13
3.6	semicolon separated line . . . . .	14
3.7	A line with 11 column . . . . .	14
3.8	Semicolon separated line . . . . .	14
3.9	Assigning column name in data frame . . . . .	15
3.10	Data Value type . . . . .	15
3.11	Walking Activity X,Y,Z Value Plot . . . . .	16
3.12	Jogging Activity X,Y,Z Value Plot . . . . .	16
3.13	Going Down Stair Activity X,Y,Z Value Plot . . . . .	17
3.14	Going Up Stair Activity X,Y,Z Value Plot . . . . .	17
3.15	Sitting and Standing Activity X,Y,Z Value Plot . . . . .	18
3.16	Data frame with out time stamp and user id . . . . .	19
3.17	Code for Reduced Data frame . . . . .	20
3.18	Reduce Data frame . . . . .	20
3.19	Data Frame with Label Value . . . . .	21
3.20	Standardized Data . . . . .	21
3.21	Frame Generation from time series data . . . . .	22
3.22	Taking 80 samples of data . . . . .	23
4.1	Extracting features from signal . . . . .	25
4.2	Removing gravitational contribution from signal . . . . .	26
4.3	Autocorrelation for the activity: Walking . . . . .	27
4.4	Autocorrelation Comparison between Walking Upstairs and Downstairs . . . . .	28
4.5	Spectral Peak Features . . . . .	30
5.1	Visualization of activity: Walking . . . . .	32
5.2	Workflow of Exploratory Data Analysis . . . . .	33
5.3	Visualization of activity: Jogging . . . . .	34
5.4	Time Series Visualization of activity: Upstairs . . . . .	34
5.5	Visualization of activity: Downstairs . . . . .	34
5.6	Visualization of activity: Sitting . . . . .	35
5.7	Visualization of activity: Standing . . . . .	35
5.8	PCA on x axis data . . . . .	36
5.9	PCA on y axis data . . . . .	36
5.10	PCA on z axis data . . . . .	37

5.11	t-SNE Reduction on x axis data . . . . .	37
5.12	t-SNE Reduction on y axis data . . . . .	38
5.13	t-SNE Reduction on z axis data . . . . .	38
5.14	LDA Reduction on x axis data . . . . .	38
5.15	LDA Reduction on y axis data . . . . .	39
5.16	LDA Reduction on z axis data . . . . .	39
6.1	Overview of Proposed Pipeline . . . . .	40
6.2	Feature Extraction from Frames . . . . .	41
6.3	Combining Features . . . . .	41
6.4	Combining Features . . . . .	43
6.5	Training Accuracy . . . . .	45
6.6	Training Loss . . . . .	46
6.7	Confusion Matrix without Normalization . . . . .	46

# List of Tables

4.1	Summary of Extracted Features . . . . .	31
6.1	Model Summary . . . . .	42

# Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

$\epsilon$  Epsilon

$v$  Upsilon

*HAR* Human Activity Recognition

# Chapter 1

## Introduction

### 1.1 Thoughts behind the Prediction Model

The necessity of recognizing human activities is increasing immeasurably now a day. Recognition of human activity plays an important role in communication between human beings and interpersonal relations. Since it contained information on a individual's behavior, temperament, and psychological state, it is hard to extract. Commonly it also called as Human Activity Recognition or in short form 'HAR'. It is basically a process of predicting what a human is doing by using few sensors or camera. There are now so many fields where this process can be applied to get expected result better outcome with prediction. For example we can implement Human activity recognition in health care, elderly health monitoring [36], day care center [18], rehabilitation center [26], human computer interaction [10], security surveillance [33], cognitive disorder recognition [43], ubiquitous computing. For example like elderly health monitoring wearable body sensor can support caretakers of nurse or doctor to record and monitor the patients or elderly people always and submit report dynamically whenever and bad movement or abnormal behavior is identified. Other application like survey on human activity and detecting location can also be benefited from this research.

To gain the better precision or efficiency in Human activity recognition with less computation expenses is a major challenge. To overcome this obstacle, the HAR group is starting to learn deep learning to replace excellently-established analytical methods that depend on hand-made extraction of features. Earlier, hand-made methods were commonly restricted to mathematical and statistical features like finding mean and time domain variance, and fast Fourier transforms coefficient in frequency domain. But now a days there are so many machine learning approaches are used to recognize and predict the human activity . Recently, human activity recognition has been solved by using CNN. CNN is a deep learning method. CNN combine the feature extraction and classification in an end-to-end approach.

Smartphones devices are perhaps the most valuable tools in our everyday lives. Through advanced technology, they are more able to fulfill the demands and expectations of consumers every day. Developers and designers are adding new modifications and modules to the devices to make these products more usable and efficient.

Sensors equipment play an important role in getting smartphone devices further usable efficient and Conscious of the climate. This is how smartphones devices become more embedded with different kinds of sensor and equipment's. This sensor and equipment's make easier to collect huge amount of data and information from human's daily routine and movement. Accelerometer is one the very common and useful sensor of it.

In almost all smartphone developers and engineers, the accelerometer sensor was a basic and common hardware. The title suggests, the accelerometer calculates the velocity shift; not the velocity itself. Data obtained from the accelerometer sensor can be extracted to identify rapid motion shifts. We use accelerometer data to develop our research and finding out the accuracy result.

Many people use compass application on their android or IOS smartphone. This particular application by some means or another detect that which direction the android devise is. There is another feature in the phone that is landscape mode or portrait mode. If enabling landscape mode is on in the phone when user tilt the phone from vertical to horizontal position, the device's screen also rotate accordingly. Android or IOS smartphone use a special technology to recognize its orientation or direction. This technology is called accelerometer. Accelerometer is a tiny device with huge utility.

In our research we used smartphone accelerometer data. At first we pre processed our data. Then we make 3 dimensional shape of our data. After that we visualize our data using machine learning dimensional reduction algorithm. Then we use MAT LAB signal processing feature extraction technique. We made fusion among features from MAT LAB approach and earlier created 3 dimensional frame . After that we feed the fusion features into the CNN model. Then we got our prediction result from this approach.

# Chapter 2

## Related Work

Human activity recognition is now a large research topic [28] connected to the productive understanding of smart ubiquitous worlds. It is the mechanism whereby the actions of an individual and its atmosphere are tracked and evaluated in order to predict the activities. Human activity recognition of behavior includes pattern classification, Modeling of action, surrounding tracking, the collection of information.

There has been huge development of microelectronics, computers systems in the recent times, which have enabled more robust sensors and mobile devices to easily capture multi-modal data from the surrounding environment. This has led to the idea of Ubiquitous sensing, which is an active research area. The goal of this research is to extract knowledge from data acquired by these sensors [19]. Recognition of human activities from sensors has become a topic of particular interest. Human activity recognition has huge applications, especially in medical, military and security fields. In the healthcare sector, for example diabetes patients are obliged to follow exercise routines for treatment purpose [16]. Recognition of different activities like walking, running, exercise etc. is essential to provide feedback regarding to the patient's behavior. Also, patients with dementia or other mental disorders could be brought under monitoring to detect anomalous activities and behaviour. This has the potential to prevent undesirable outcomes [13]. HAR also has the potential to contribute in military applications. Information on the soldiers' activity, location and health could provide crucial to monitoring the performance and ensuring their safety. Insights gained from such research could also contribute to support strategic decision making.

The earliest work on Human Activity Recognition date back to 1999 by Foerster et al. [5]. He used Statistical distances to predict activity from piezoelectric sensors mounted on the bodies of test subjects. Different statistical as well as machine learning approaches towards this task have been proposed over time. Kim et al. used Hidden Markov Models to classify human activity from sensor data [17]. Smart devices like smartphones saw wide adoption in recent years. A public domain dataset collected from smartphone sensors for activity recognition [24] saw applications of machine learning over smart phone sensor data to tackle this problem. In more recent times, deep learning techniques like convolutional neural networks have been applied to the task of HAR from smartphone sensors with exceptional success [34].

Usually, Human activity behavior and activity detection mechanisms include several major parts. A minimal level detecting control system that continually collects use full data through microphone, accelerometer, light device practices. A system for processing and selecting features that turn existing or raw device information or data to features which enable differentiate among activities. In the classification part, the system uses the features to decide which behavior or activity of a person or a bunch of people is participating in. Such as sitting, standing, walking, jogging, etc. There are many model for human activity recognition. Few are given below

Vision-Based Activity Recognition [37] : This requires equipment for vision detection for example Camera. This monitoring systems to track the actions of an individual or shifts in his or her surroundings. This system has several portion. The First one is tracking an individual or a group of people. Then comes the individual's behavior detection. After the recognition of activity of particular individual or a bunch of people. Finally the evaluating or analysis of large-level activities. Some certain approaches to science using various methodologies including: stereo camera, depth camera, thermal camera, 3D camera, infrared or night vision camera, network or ip camera, action camera, dome camera etc are used to detect and Catch the sense of operation. Some times multiple camera are being used to capture and detect any 3D shape of individual. Evaluating an image has several steps. First one is cutting the human shape or object shape from it's back ground. An particular algorithm is implemented to remove the background. That also works with if there is any surroundings changes also.

Sensor-Based Activity Recognition [44] : This utilizes sensor system solutions to track the actions of an individual and his / her surroundings. Sensors are remained linked to individual's body in this situation. Utilizing data analysis or artificial intelligence methods and machine learning algorithm sensor device collects data creating Structures of behavior and identification of action [41]. Through this the identification of activities like walking jogging running sitting standing and any gesture or movement is recognized throughout these algorithmic system. Because of the volume or battery capacity, these portable devices may not be suited for real world applications. Throughout this method, external sensors devices or devices connected to things can be utilized. Hidden Markov Model (HMM) algorithm is a very common algorithm for this kind of approaches. By this algorithm activity of individual can easily be identified and recognized. On the other hand another method is CRF method which stands for Conditional Random Field method. This is basically a statistical method for any statistical or classification approach.

Human-Sensing Taxonomy : Categorize the system of obtaining any data on human activity in any surroundings underneath the broad category of human sensing It mostly explains the spatiotemporal properties (STPs) inference .there are few lower level components and issue based on the position and history of human in an surroundings. More specifically:

- 1) Presence: Presence of a person is one of the most common and important part of human activity recognition. It is most common and used in any real world application or software. Presence of a person can be determined by a very important



and particular sensor like motion detection sensor or motion detector alarm or any kind of proximity sensor. Proximity sensor can easily identify any object or people nearby with out any touch contact or connection. Even IR (inferred sensor) or light sensor can be used to identify the presence of any human or object.

2) Count: So another task is to count how many person or human are present in an environment or place . This task can be done by also some important sensor.those sensor are likely to count how many human or object are presented in a particular surroundings. This kind of sensor can cover an entire or particular area or environment to count. it also count how many people are coming and going into a particular place from a gate. Even how many people are presented or working in a particular place for job or other purpose can be counted by this kind of sensor. Inferred Sensor or thermal image capturing sensor can be used here.

3) Location: [45] Another important task is localization or locate any particular person or object. This task can be done by GPS and GPS tracking sensor. now a days there are so many application and sensor which can be used for this type of operation. Position can be identify from indoor and outdoor position of a human.For example like indoor position there are method like WiFi based approaches which has several parts like Angle of Arrival, Time of Arrival, Received Signal Strength Indicator etc. Another examples are like visible light based indoor localization [42]. It is approaches that is measure the position of a particular person or object with the basis of LED luminaries.

4) Identity: Who is a that particular individual can also be identify by using those approaches . This can be achieved from a person's previous history or information is or not saved in the data base behind that object or not. It is basically a more extension of a tracking system . Information can be gather about the person from the database or recent cash information etc.

# Chapter 3

## Methodology

### 3.1 Overview

To develop our research and we go through several processes. Our first task was gathering the data. We use accelerometer data to develop our research . First we work with the data pre processing part. Where We standardize our data and make frame . Then we use several algorithm for data visualization and analysis.After that we use signal processing algorithm for feature extraction and after that we do fusion between our feature frame and early created frame. Then we feed our fusion frame into Convolutional Neural Network algorithm . We used several machine learning algorithm here for data analysis and data visualization part . Those algorithms short description are given below :

LDA (Linear Discriminant Analysis) : LDA is a mainly used as a technique for reduction, which usually applies in pre-processing stage for classifying patterns and machine learning programs. Technically, the less dimension there will be, the less time it will take to complete the whole process. Hence, LDA is used to deduct the dimensions in each dataset. In our research work, we have used LDA to reduce the number of dimensions for each and induvial axis. This helped us compress the total dataset as well as retain enough information that we need to complete the process. During our research, LDA was also applied on 532 frames which were generated from time-series data. Hence, each frame was reduced into two components and scatter plots were produced from the reduction.

PCA (Principal component analysis) : PCA is a statistical process which is used to convert a set of extracted data which are correlated variables, into a set of linear values which are not corelated variables using orthogonal transformation. In some cases, it's hard to detect the variables that we will be removing in the reduction process. On the top of that, some variables are dependent on each other which makes it even complicated to reduce the total number of variables. PCA uses a special feature extraction technique which separated the most important variable from the least one and detects the dependency as well in any given linear model. In our research work, we have used PCA reduction to detect the correlation and dependency among the frames. We were left out with 532 frames after we have applied the reduction algorithm on the dataset. Each frame comes with 80 different samples of acceleration data from all three separate channels that we have used in

our research. We had to run PCA separately on data that we received from all the three channels, as PCA is not meant to deal with multi-dimensional dataset. Each channel was reduced into two components which are direction and magnitude for the ease of visualization.

t-SNE (t-Distributed Stochastic Neighbor Embedding) : t-SNE is a widely used non-linear process for reducing dimensions which is specially works well for the visualizing high dimensional datasets. This algorithm is most popular for tasks like image processing, speech processing, natural language processing, genomic data processing etc. t-SNE starts working by calculating the probability of similar points in high dimensional space as well as the probability of similarity of points in low-dimensional space. Then it compares both probability and tries to minimize the difference to present a prefect representation of low- dimensional data. In our research work, we have applied t-SNE on the frames which were extracted using Python. To be more specific, we used Multi-core t-SNE using the Barnes-Hut algorithm for this process. During our work reduction algorithm was applied on all the x, y and z acceleration channels. Later on, the reduced t-SNE components were plotted using the python library named plotly express.

CNN (Convolutional Neural Network) : CNN is mainly a deep learning algorithm which grabs image or part of image as input, then it assigns weights and biases to different segments of the image so that it can detect and differential one image from another. CNN requires less pre-processing than other classification algorithms that exist. However, in primary stage this algorithm is hand-engineered. If the algorithm is trained enough it can learn complex filtering and classifications. The design of CNN is very similar to the pattern of Human Brain which is inspired by Visual Cortex. In our research work, we have applied CNN as the core classification algorithm. Before feeding the data to CNN, we have categorized the data into two separate subsets for the purpose of training and validation. In order to make the learning more impractical, 80 percent data was used to train CNN and remaining 20 percent for validation. As this process requires high computing power, it was done by high end GPU. The network was trained for 400 epochs while training and each epoch was validated by cross checking the test data. Finally, the model with the highest validation accuracy is selected as our desired model. Further training will definitely improve the accuracy and performance of the model.

### 3.1.1 Python and Libraries

Before we go depth into describing this coding's developing method here are the description it's several phases. Moreover, we should acknowledge about the basic knowledge theories and information used in here to get a better result for every steps of the whole coding development

We wrote the program with Python which is a general purpose yet highly sophisticated and interpreted programming language. So needless to say the first bases to cover are the modules libraries like any other programming languages. The lines of codes are simply to utilize these resources to process the input data analyze it and show the result. Here the following python dependencies are used

Tensorflow, Keras, Numpy, Pandas, Matplotlib, Scikit-learn, SciPy stats

## 3.2 Workflow

The workflow of our research and development is given below

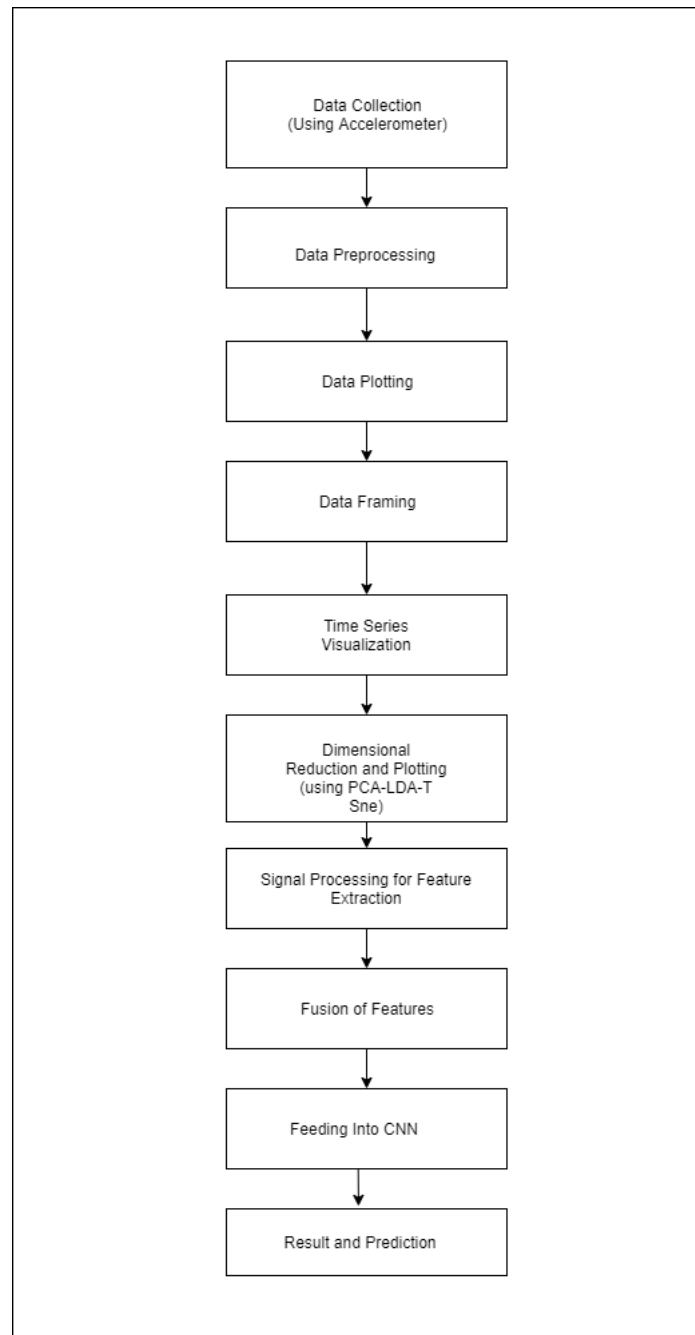


Figure 3.1: Workflow Diagram

### 3.3 Accelerometer

Accelerometers [49] [47] in cell phones are utilized to identify the direction or to trigger the android or IOS based smartphone acceleration. This device estimates direct speeding up of development. The perusing incorporates three tomahawks whose bearings are predefined as in Fig underneath. The speeding up is the crude information stream from the accelerometer. A lot of vectors speaks to the crude information An accelerometer will evaluate the directional advancement of a contraction yet won't have the option to determine its parallel direction or tilt during that development precisely.



Figure 3.2: Smartphone and accelerometer axis

The three direction readings are joined with a period stamp. So as to test the recurrence, the vast majority of existing accelerometers give a UI with the goal that the client could pick a superior examining rate through examinations. Single and multi-pivot accelerometers identify the consolidated extent and heading of direct, and also identify the speed of gravitation and the speed of rotation [8]. They can be utilized to give constrained movement detecting usefulness. For instance, a gadget with an accelerometer can identify pivot from vertical to even in a fixed state Location. Therefore, accelerometers are basically utilized for basic movement detecting applications in purchaser gadgets, for example, changing the screen of a cell phone from representation to scene direction. Its prominence is because of the way that it straightforwardly gauges the subject's physiology movement status. For instance, if a client changes his/her action from strolling to running, it will think about the sign state of the quickened perusing along the vertical pivot there will be an unexpected change in the sufficiency. Also, the quickening data that what demonstrate the motion towards the a designated time frame, which is helpful for the in the convoluted the activity concession The manner in which how accelerometer functions

- 1.static force : this force is works on the sensor or the device as the orientation changes or the gravitational pull
2. Force/acceleration exerted upon the sensor movement/force detection

Acceleration is the time-divided calculation of the velocity shift or speed. For example a bus is accelerating from its start position to 30ms in 5 second. So its acceleration is  $30/5$  equal to 6 meter per second.

In Many huge electronic gadget accelerometer appears to be a basic and minor part. Accelerometer has numerous discernable segments. Those parts works various approaches to play out the diverse capacity. The two-fundamental and significant pieces of an accelerometer is piezoelectric impact and the capacitance sensor. Piezoelectric Effect is the capability of certain substances in response to applied mechanical worry to deliver a voltage. While vibrating the accelerometer, a power is delivered that works on the piezoelectric part and the motor mass. As a result of the piezoelectric impact, this movement or effect creates a voltage yield proportionate to the power applied. At that point the accelerometer utilize the voltage to evaluate the direction and speed.

### 3.4 Data Set Description

The Data was collected by the WISDM Lab[24]. Dr. Gary Weiss is the team leader of the research and development of this lab. He teaches in information and computer science department of Fordham University, New York, United States of America. This lab has so many members who are Fordham university graduates. The WISDM Lab works for collecting the sensor data from mobile phone and electronic devices. They also work for data mining and big data. Their research and works has been accredited by Fordham University and it's other Undergraduates College, the National Science Foundation and Google Inc.

For collecting data, it was important to carry and android based smart phone by a large number of smart phone user. They did a few common activities for collecting data. Data was collected from 29 different people. They carried an android mobile phone and performed few common activities. These people carried the android mobile phone with their pant's pocket. After that they perform those certain activities. Those activities are walking, jogging, going upstairs, getting downstairs and finally sit and stand for a certain time duration.

A particular smart phone application was built for data collecting. The application was installed in the smartphone and was being used for data collection. The application was used in collecting the by accelerometer of the android device. The data were collected very frequently and the accelerometer collected the data in every 50ms. That means 20 samples per second.

In this data set we have three accelerometer data for every activity and one accelerometer data in three dimension. The dimensions are x, y and z. The data set has also user id, activity name and time stamps in millisecond. There are total Number of examples were 1098297. Number of attributes are walking, jogging, upstairs, down stairs, sitting, and standing. And distribution of classes. This type of data basically used for classification problem. In this data set the data follow a particular format

The format is:

```

33,Jogging,49105962326000,-0.6946377,12.680544,0.50395286;
33,Jogging,49106062271000,5.012288,11.264028,0.95342433;
33,Jogging,49106112167000,4.903325,10.882658,-0.08172209;
33,Jogging,49106222305000,-0.61291564,18.496431,3.0237172;
33,Jogging,49106332290000,-1.1849703,12.108489,7.205164;
33,Jogging,49106442306000,1.3756552,-2.4925237,-6.510526;
33,Jogging,49106542312000,-0.61291564,10.56939,5.706926;
33,Jogging,49106652389000,-0.50395286,13.947236,7.0553403;
33,Jogging,49106762313000,-8.430995,11.413852,5.134871;
33,Jogging,49106872299000,0.95342433,1.3756552,1.6480621;

```

Figure 3.3: Data Set Text View

[user], [activity], [timestamp], [x-acceleration],  
[y-acceleration], [z-acceleration];

And the line representative example is:

```
33, Jogging, 49105962326000,-0.6946377, 12.680544, 0.50395286;
```

Here timestamp value is numeric. It is generally the phone's uptime in nanoseconds. X acceleration is basically numeric, floating-point values. . Acceleration in X direction was taken by the smartphone's accelerometer. The acceleration recorded includes gravitational acceleration toward the center of the Earth, so that when the phone is at rest on a flat surface the vertical axis will register +-10. Same as Y and Z is basically the acceleration in y and z direction. There are total Number of examples were 1098297. And the walking attributes data count is 424,400. Which is 38.6 percent. Jogging attributes data count is 342,177. Which is 31.2 percent. Upstairs attributes data count is 122,869. Which is 11.2 percent. Downstairs attributes data count is 100,427. Which is 9.1 percent. Walking attributes data count is 424,400. Which is 38.6 percent



## 3.5 Activities

In our research we use the six regular and common activity of human. They are sitting, standing, upstairs, downstairs, jogging and most common one walking. For our research purpose we comes up with this ‘Actitracker’ data set from WISDM – Wireless Sensor Data Mining Lab of because the activities record that are included in the data set are performed regularly by many people in our daily life . Those activities include motions. And these motions take place for a particular time duration

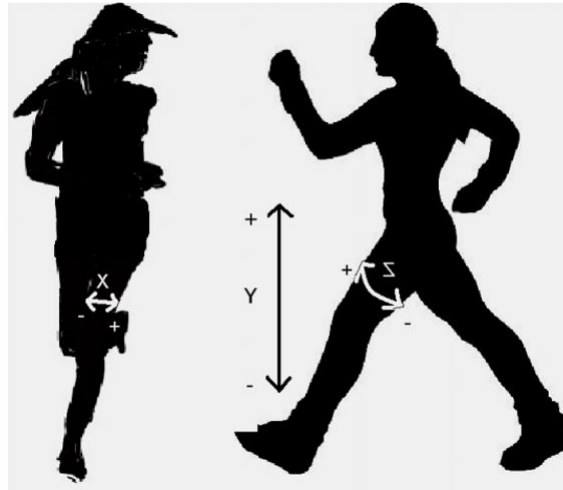


Figure 3.4: Activity

That’s what makes it easier to recognize. In addition those sets of activities include repetitive motions. These repetitive motions helps to recognize activity a lot easier. When the data’s were recorded they were recorded in three axis. These axis are X, Y, Z. In Z direction or Z axis records the forward motion of the leg. In Z direction or Z axis records the forward motion of the leg. Y direction or Y axis records the up and down motion. X direction or X axis records the horizontal motion of the leg.

## 3.6 Research Development Using Python

### 3.6.1 Data Pre Processing

Our first task is loading the data set. We are using here pandas data frame to load the data set. However, while loading the dataset from the csv file to a pandas dataframe, we faced an error. The error occur because there are some parser error. The error message is shown that the error occurred in a particular line.

```
ParserError: Error tokenizing data. C error: Expected 6 fields in line 134634, saw 11
```

Figure 3.5: Parser Error

That means each line supposed to have 6 column. For example the figure below:

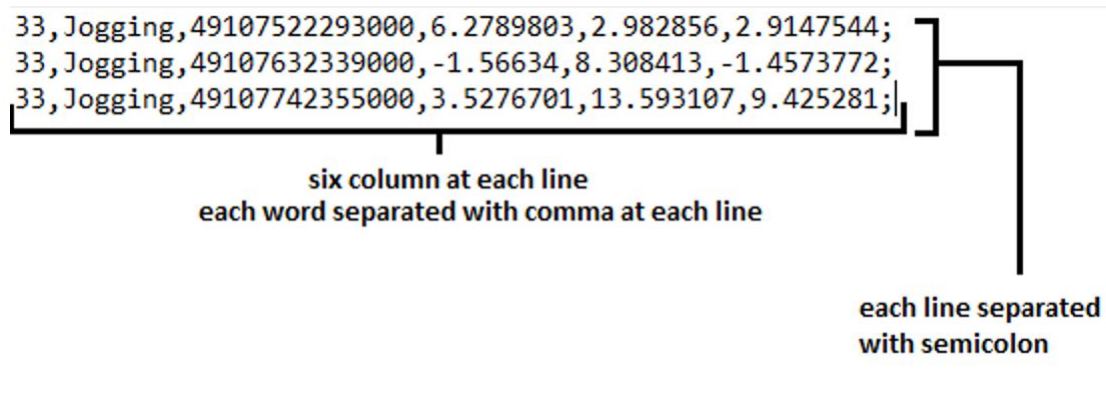


Figure 3.6: semicolon separated line

But that line has 11 column.



Figure 3.7: A line with 11 column

If one line is fixed we may get this similar error in other line because the whole data set is not pre processed properly. So rather than loading into pandas data frame we need to load it into native python file application opener. So we are loading the file into native python file opener with python file opener file location function. Now we are going to read the file line by line. We read the file into a variable so that we can enumerate the data set line by line. Now we are going to separate each of the word in the line. Each word in the line were separated by comma ‘,’. And the sixth column of a line is a semicolon ‘;’. Every line is separated from other line using the semicolon ‘;’. We remove the comma. Semi colon and white space from the lines.

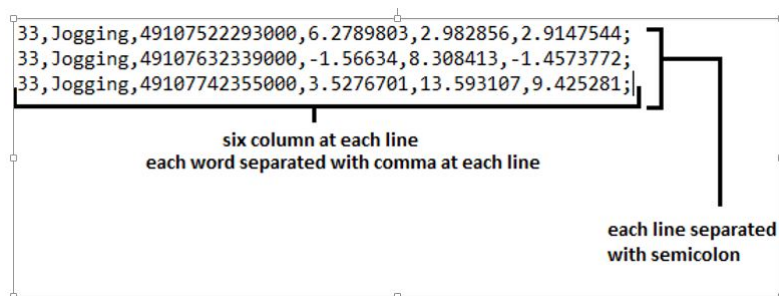


Figure 3.8: Semicolon separated line

We are separate each word from every line and save them into temporary array. And from the array we append every value of these into a global array list.

After that we pass our global array list into a pandas 2 dimensional data frame. Then we assign column name into the pandas data frame at the data frame top head. The columns are user Id, activity name, x axis magnitude value, y axis magnitude value, z axis magnitude value. For example

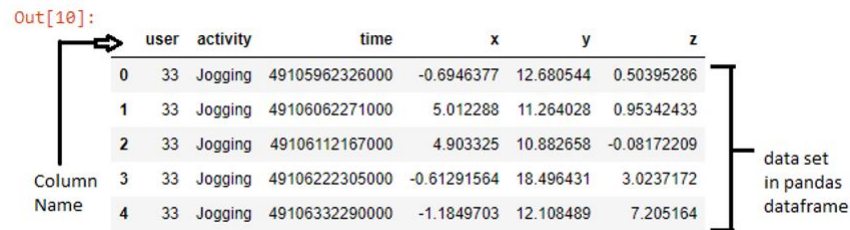


Figure 3.9: Assigning column name in data frame

If we look at the shape of the data frame by using python shape and size function, we can see that it is 2 dimensional data and there are total 2060496 data with 6 column and 343416 lines.

So then we use python data.info() function to find out what type of data were those. We can see that all the data here are string type or object type

```

RangeIndex: 343416 entries, 0 to 343415
Data columns (total 6 columns):
user          343416 non-null object
activity      343416 non-null object
time          343416 non-null object
x             343416 non-null object
y             343416 non-null object
z             343416 non-null object

```

Figure 3.10: Data Value type

But for machine learning algorithm we need numerical data. So we had to convert all the time, x, y, z data into numerical form .So we need to convert all the string values into floating values. So we use astype() function to convert all the string value of x, y, z data into floating point data. So now if we see the data using data.info() function we can see that all the x, y, z data are converted to floating point data. We don't need to convert other data column like user, activity and time because those data column will not used in the machine learning process. Now we have to check if there is any null value in data set. So by using data.isnull().sum() we checked that there is no null values in our data. And we can see that there is no null value is present in the data

### 3.6.2 Plotting

Now we are going to plot the X, Y, Z data for few seconds. If we plot the data we need to plot them few seconds. If we look at the file of data set description we can see that the sampling rate is 20Hz. That's mean the data has been processed in 20Hz. Now we are going plot the data for each activity. Here we are going to plot 10 second activity for each of the data.

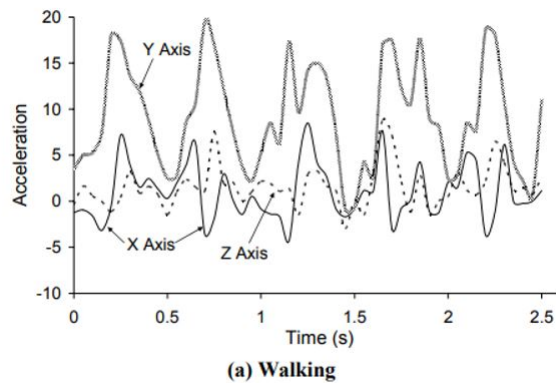


Figure 3.11: Walking Activity X,Y,Z Value Plot

If we look the graph plot of walking activity we can see that here, the peaks for the Y axis are high. Time duration interval is 0.50 second. On the other hand peaks shown for the z- axis or z direction acceleration represents the peaks with low spaces or magnitude. The peak distance of z axis and y axis demonstrates the time duration of one pace. X axis magnitude demonstrates also low measurement. There are much variation in x, y, z data.

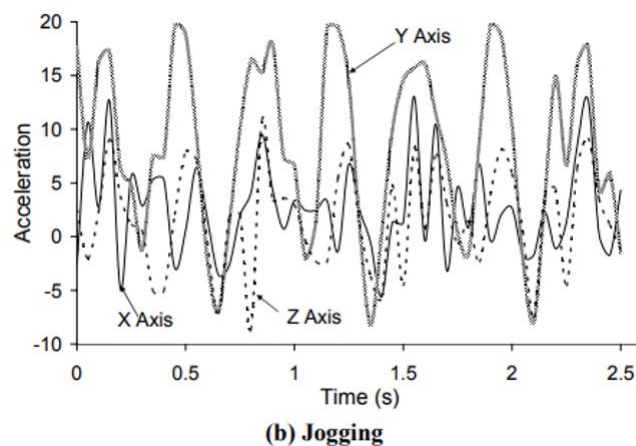


Figure 3.12: Jogging Activity X,Y,Z Value Plot

If we look at the graph plot of Jogging activities distinguishable pattern are represents for the y axis and z axis. Nonetheless the time duration between the peaks

of this activities are less which is around 0.25 second. The span area of Y axis acceleration for this activity is bigger than walking. But the motion shifting is more visible and distinguishable in the negative axis or direction. So for this activity we can see that there are much variation in x, y, z data

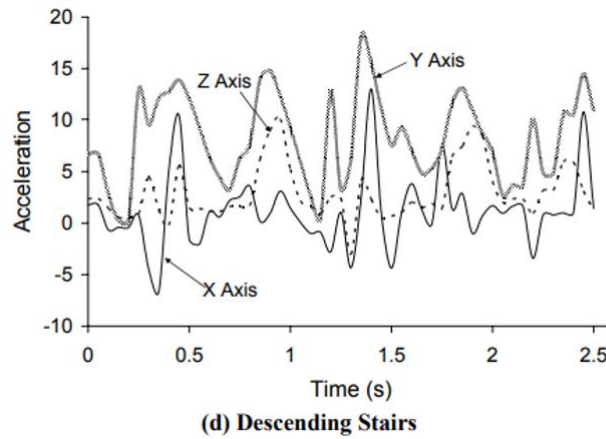


Figure 3.13: Going Down Stair Activity X,Y,Z Value Plot

Now by looking the graph plot for the going downstairs activities, the peaks of Y acceleration quitter lower. The time duration interval for this is 0.50 second. Every peak with lower pace demonstrates the motion of a single step of going downstairs. The magnitude of Z axis also demonstrate the indistinguishable pattern with negative acceleration. Which consider the movement of going every steps stair. And the X axis data magnitudes demonstrate the small peaks. It's pace or plot runs between the positive and negatives values as well. It has very less changes in the acceleration.

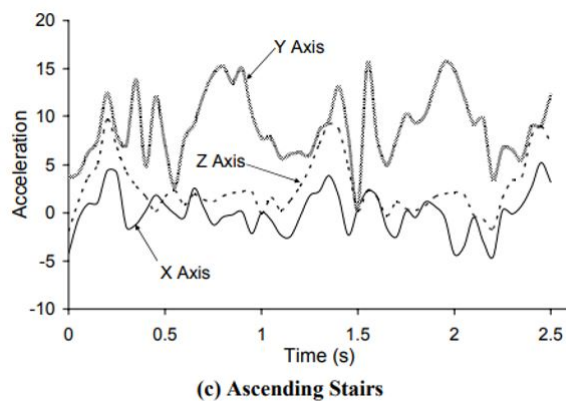


Figure 3.14: Going Up Stair Activity X,Y,Z Value Plot

Now for going upstairs activity we can see the peaks of Z axis and Y axis magnitudes are spaced according to tome interval 0.75 second apart. It is taking longer time as

it indicating the activity which is going upward to the stairs. This activity's data variation is less various from the walking and jogging

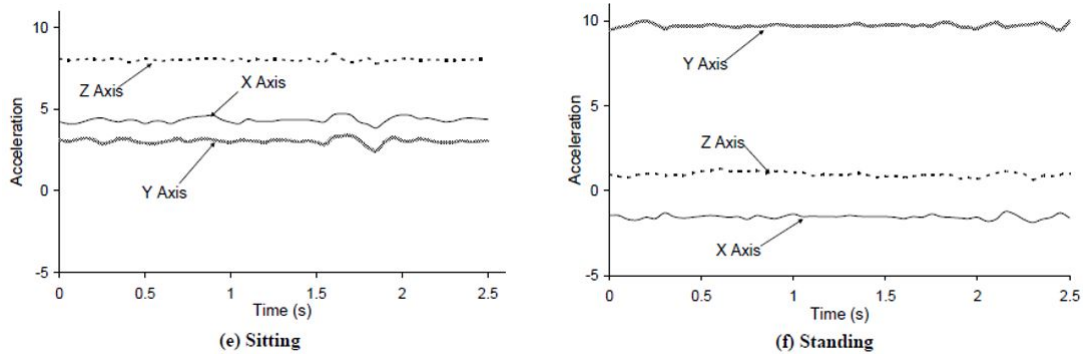


Figure 3.15: Sitting and Standing Activity X,Y,Z Value Plot

On the other hand the rest activities are sitting and standing. Sitting and standing do not manifest any kind of systematic periodic recurrent motion. These happen because the acceleration magnitudes are comparatively similar for every axis. We can see from the graph plot that x, y, z magnitudes are almost linear and peak for x and y axis are comparatively same and relative. For this reason it is easily recognizable and use for easy for classification. The plot of this data are almost flat and there is no much changes in the data.

So we can see that most of the changes of acceleration are taking place in the Walking and Jogging. And less changes of acceleration are taking place in standing and sitting.

Machine learning algorithm will work on this. If there are much activity and changes into acceleration machine learning algorithm will classify it into jogging or walking and if there is less changes in acceleration and less activity it will classify in sitting or standing.

The accelerometers sensor of smartphone capable to regulate and decide direction regarding the Earth's gravitational field. So if the smartphone or direction or orientation may changes or shifts in user pockets it will automatically compensate or correct its accelerometers orientation or direction. For deep learning network it will easier to detect all the pattern and to classify it accurately.

### 3.6.3 Data Pre Processing

Now we go back to data pre processing steps again

Now we are going to remove the time stamp and the user id value of data from the data frame. As those two column won't be needed for further processing. So we use `drop()` function to drop the column of time stamp and user id . So and we copy the new data frame into a new data frame. In the new data frame there is only activity, x, y and z axis column for further processing.

activity	x	y	z
Jogging	-0.694638	12.680544	0.503953
Jogging	5.012288	11.264028	0.953424
Jogging	4.903325	10.882658	-0.081722
Jogging	-0.612916	18.496431	3.023717
Jogging	-1.184970	12.108489	7.205164

Figure 3.16: Data frame with out time stamp and user id

As we already mentioned in the data set description the different activities and their distribution are given in the data set. Those were highly imbalanced data. For example, the walking activity data count is 424,400. Which is 38.6 percent. Jogging attributes data count is 342,177. Which is 31.2 percent. Upstairs attributes data count is 122,869. Which is 11.2 percent. Downstairs attributes data count is 100,427. Which is 9.1 percent. Walking attributes data count is 424,400. Which is 38.6 percent. So here we can see that the data distribution is not same for different activity. If we use different number of data for different activity for machine learning approach our end result might be highly over fitted and eschewed to walking and jogging so we needed to distribute all the different activity data equally. The lowest data count in standing activity which is 3555. So we need to balance all the data . So we select 3555 data from all other activity

So in order to reduce all the data we need to reduce down every data for each activity to 3555 data and make different data frame for each the activity. Then we append all the data frame together so we can make a combination of all data frame where every activity has 3555 data.

```
Walking = df[df['activity']=='Walking'].head(3555).copy()
Jogging = df[df['activity']=='Jogging'].head(3555).copy()
Upstairs = df[df['activity']=='Upstairs'].head(3555).copy()
Downstairs = df[df['activity']=='Downstairs'].head(3555).copy()
Sitting = df[df['activity']=='Sitting'].head(3555).copy()
Standing = df[df['activity']=='Standing'].copy()
```

making dataframe for each activity

---

```
balanced_data = pd.DataFrame()
balanced_data = balanced_data.append([Walking, Jogging, Upstairs, Downstairs, Sitting, Standing])
balanced_data.shape
```

appending dataframe for each activity into a big dataframe

Figure 3.17: Code for Reduced Data frame

Now if we check the data frame to see how many data are presented for each activity we can see that there are 3555 data for each activity. So this is a balanced data set. By using this balanced data set we are going for further process.

```
balanced_data['activity'].value_counts()
```

```
Jogging      3555
Standing     3555
Walking      3555
Downstairs   3555
Upstairs     3555
Sitting      3555
Name: activity, dtype: int64
```

Figure 3.18: Reduce Data frame

Now we can see in the activity column the data of activity variable are in string format. For machine learning approach we need to convert the string variable of activity column into categorical variable. In order to convert the string to categorical variable we need to import label encoder from scikit-learn library. We use label.fit transform () function to convert the string activity data to categorical data. We pass the data frame and pass the activity column value through the parameter.



So through this fit transform we encode the activity column values. So after encoding we can find that all the activity column values encoded categorical value in label column.

	activity	x	y	z	label
597	Walking	0.844462	8.008764	2.792171	5
598	Walking	1.116869	8.621680	3.786457	5
599	Walking	-0.503953	16.657684	1.307553	5
600	Walking	4.794363	10.760075	-1.184970	5
601	Walking	-0.040861	9.234595	-0.694638	5
602	Walking	2.492524	8.730643	-1.457377	5
603	Walking	0.531194	9.888372	-1.225831	5

**Encoded categorical values**

Figure 3.19: Data Frame with Label Value

So by this we find the categorical value for every activity. For example for walking categorical value is 5, for going upstairs categorical value is 4, for standing categorical value is 3, for sitting categorical value is 2, for jogging categorical value is 1, for going downstairs categorical value is 0.

So in the diagram no. x, y, z is feature space and label is predictor which will be our output as target variable.

Here we can see that x, y, z are not in a same variance. So we need to make the X, Y, Z data into same variance. Because while doing machine learning algorithm there will be more or less spreading of X, Y, Z values and we cannot get the appropriate result out of it. So we need to standardize our data. So to standardize our data we need to bring them under same variance. We take x, y, z balanced data value in x and the label values in y. then we use StandardScaler () and fit transform() function to standardize our data. After standardize our data we put the standardized data into a data frame named scaled X.

	x	y	z	label
0	0.000503	-0.099190	0.337933	5
1	0.073590	0.020386	0.633446	5
2	-0.361275	1.588160	-0.103312	5
3	1.060258	0.437573	-0.844119	5
4	-0.237028	0.139962	-0.698386	5
5	0.442677	0.041644	-0.925081	5
6	-0.083546	0.267510	-0.856263	5

Figure 3.20: Standardized Data

So we can see the x, y, z values has been scaled and brought down into a single variance also. So there are no much variance shift in x , y and z

Now we are going to make every data frame into a 3D shape for prediction. We need to make those data into a shape because for a particular person who is doing any activity he is doing that for some time. For example a person is going to do that activity for 2 second, 3 second or 4 second. So we are going to take first four second of the data. We cannot make prediction only three values of X, Y, Z what a person is doing. So we are taking 4 second of a data as it will be considered as single activity. Similarly we take next four second data. Thus we take every four second data to make a prediction

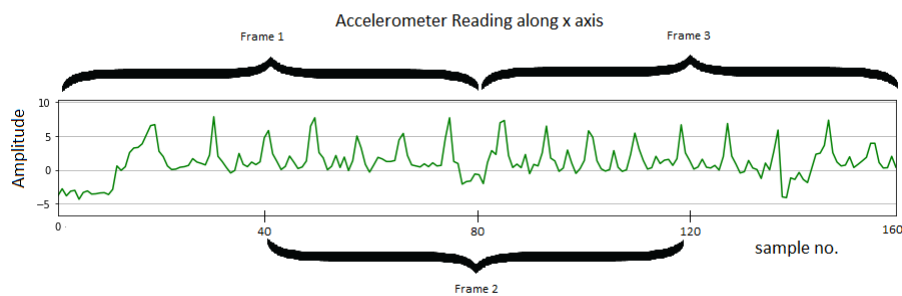


Figure 3.21: Frame Generation from time series data

Now we are approaching for the frame preparation process. we import Scipy stats library. We already we know we took 20 data samples of X, y, z values for per second. So our Fs will be 20. And we are taking data for 4 second. So there will be 20 into 4 equal to 80 samples we are going to take for making a prediction about a person human activity.

So after that we increase our hop size of taking data sample by 2 second. So for 2 second our total sample of data we are going to take in next phase is 2 into 20 Fs equal 40. So our hop size will be increased by 40 data sample.


In the get frame function we will pass the data frame and the frame size and the hop size. Then we read here our frame size and hope size in the for loop. After that we

will create a range for our time frame in the for loop that which will be advancing with the size of hope size. Thus we will make a data frame of 4 second by using each sample of X, Y, Z data for the whole data sets. That means we are preparing a data sample of four second here. But sometimes two activity's data samples may

be brought in 4 second data. So in the label variable we are considering the activity for the data frame which comes most number of time of in the data samples. We will use Stats.mod function to decide which activity number comes or occur most of the time. Then we will append X, Y, Z data frame into frame list and label into a

labels list. Then we will convert the list into a numpy array and Then we will return

	x	y	z	label
0	0.000503	-0.099190	0.337933	5
1	0.073590	0.020386	0.633446	5
2	-0.361275	1.588160	-0.103312	5
3	1.060258	0.437573	-0.844119	5
4	-0.237028	0.139962	-0.698386	5
5	0.442677	0.041644	-0.925081	5
6	-0.083546	0.267510	-0.856263	5
7	0.245343	0.490718	-0.686242	5
8	0.574233	0.708612	-0.880552	5
9	-0.441670	0.820217	-0.868408	5
10	-0.174904	-0.309113	-0.334055	5
11	-0.317423	-0.428689	-0.342151	5
12	-0.101818	0.073531	1.228521	5
13	-0.204139	-0.247996	1.260906	5



we are going to take  
80 samples on ward  
for prediction

Figure 3.22: Taking 80 samples of data

frame and labels numpy array. Frame is feature space and labels is target variable. So in this function we will pass our standardize data frame scaled X and our frame size and hop size. By this we will get a particular shape of data in 3 dimension. We get the data frame which of standardize X data frame into a variable X which size is (532,80,3) and the label values in Y variable which size is (532) . Y consists of the label values

# Chapter 4

## Signal Processing for Feature Extraction

In the field of research and development, the use of signals is omnipresent. It is necessary to process, analyze and extract information from time domain data for any predictive application. [20]. Signals act as the raw data to any machine learning application. Signal processing is necessary to aid in the decision making process.

Consolidated signal processing methods are used in our workflow of extracting significant features from the signal. There are also some other features like Principal Component Analysis (PCA) [15], Autoregressive Model (AR) [11] and Haar filters [14]. Our methodology uses statistical and spectral features of the feature obtained from consolidated signal processing techniques. An overview of our workflow for extracting features is shown in Figure 4.1. Different signal processing methods are used to generate a fairly small number of highly-descriptive features which are then used to aid in the prediction of activity from accelerometer sensor data. We use MATLAB[39] to extract different features mentioned in the following sections. The generated features help our machine learning architecture, which is a convolutional neural network, to deliver high-performance results with minimal training data.

### 4.1 Features

The following features are extracted from the time-series data frames using MATLAB.

- Average value for acceleration
- RMS value for acceleration
- Auto-correlation features for all three acceleration components
- Spectral peak features
- Spectral power features

Except the first feature, all other features employed the use of a digital filter to remove gravitational contributions with digital filter.

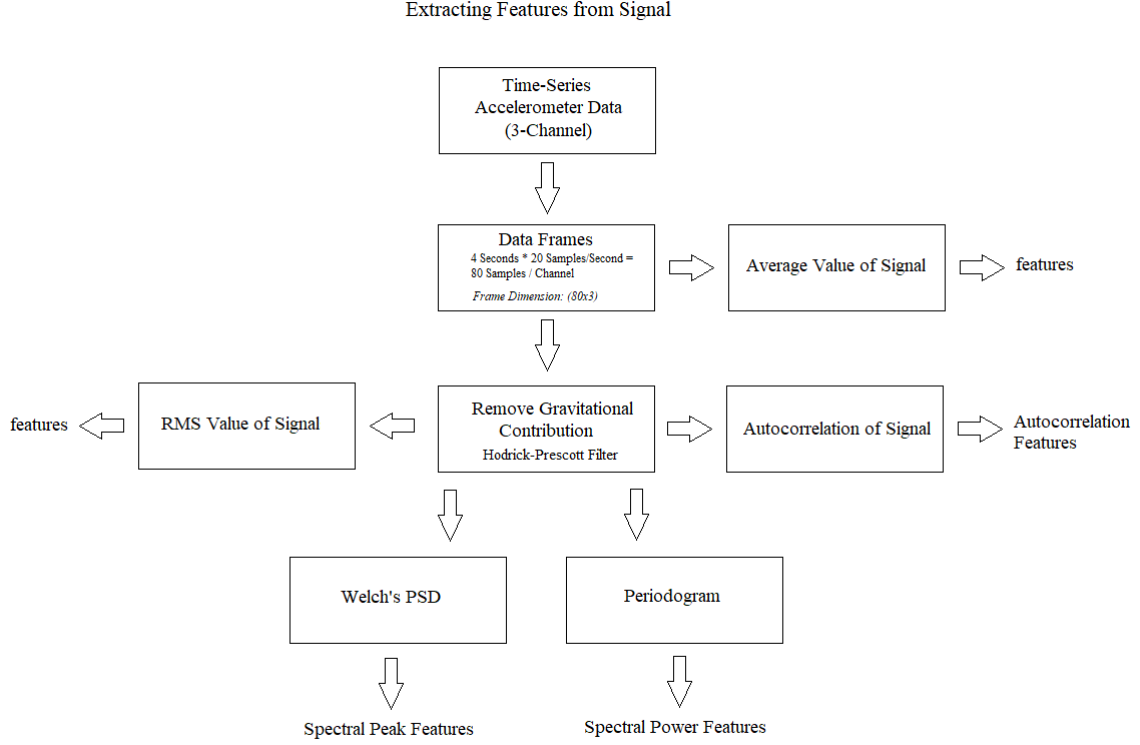


Figure 4.1: Extracting features from signal

## 4.2 Removing gravitational contribution from the signal

Since we use accelerometer sensors for recording activities, gravitational acceleration could contribute to the sensor reading [27] as a constant DC offset.

In order to filter out the gravitational components, we use a Hodrick-Prescott filter[3], [7] for trend and cyclical components. The time series  $y_t$  is separated into two components by the Hodrick-Prescott filter. The two components are a trend component  $T_t$  and a cyclical component  $C_t$ , where  $y_t = T_t + C_t$  [2]. This filtering corresponds a cubic spline smoother where  $T_t$  corresponds to the smoothed portion.

Let,  $m$  be the number of samples and  $\lambda$  be the smoothing parameter. The objective function of the filter is as follows -

$$\sum_{i=1}^m C_i^2 + \lambda \sum_{i=2}^{m-1} ((T_{i+1} - T_t) - (T_t - T_{i-1}))^2 \quad (4.1)$$

The main challenge here is to minimize this objective function over  $T_1, \dots, T_m$ . The first sum is used to minimize the time-series and trend component difference, whereas the second sum is responsible for minimizing the second-order difference of the trend component. These two functions are respectively analogous to minimizing the cyclical component and the second derivative of the trend component.

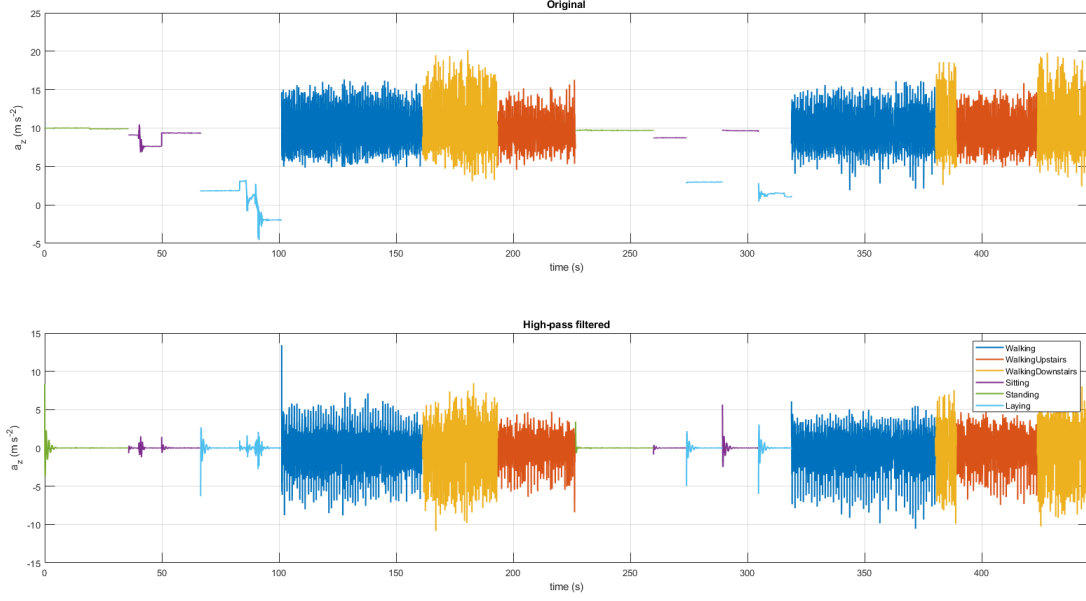


Figure 4.2: Removing gravitational contribution from signal

Figure 4.2 shows the signals before and after removing gravitational contributions using the Hodrick-Prescott filter in MATLAB. We can clearly see that after using the high pass filter, the DC offset due to gravitation is completely removed from the sensor data.

### 4.3 Average value of Signal

Average values of the signal were taken for all three acceleration components as a feature. This produced a total of three features, one from each axis.

### 4.4 RMS value of Signal

The root-mean-squared value of the signals along x, y and z axes after using the digital filter were also taken as three separate features.

### 4.5 Autocorrelation Features

Autocorrelation is an important feature for the function of frequency estimation. It can be used to effectively estimate low-pitch fundamental frequencies.

Cross correlation [6] is basically an estimate of correlation between two random processes,  $x_n$  and  $y_n$ . The true cross-correlation between these random processes is given by -

$$R_{xy}(m) = E\{x_{n+m}y_n^*\} = E\{x_ny_{n-m}^*\} \quad (4.2)$$

where  $-\infty < n < \infty$ . complex conjugation is denoted by the asterisk sign and  $E$  represents the expected value operator. Practically, the random process is of infinite length, of which a finite segment of one realization is available. Therefore, correlation is only able to estimate the sequence.

If  $c$  is the output vector, then -

$$c(m) = \hat{R}_{xy}(m - N), m = 1, 2, \dots, 2N - 1 \quad (4.3)$$

Auto-correlation on the other hand is a special case of correlation. When the correlation of a signal is calculated with a delayed copy of itself, this is termed auto-correlation. This result is also termed serial correlation [8]. In other words, auto-correlation corresponds to the similarity between a function and another version of itself which has a time lag has been applied to it.

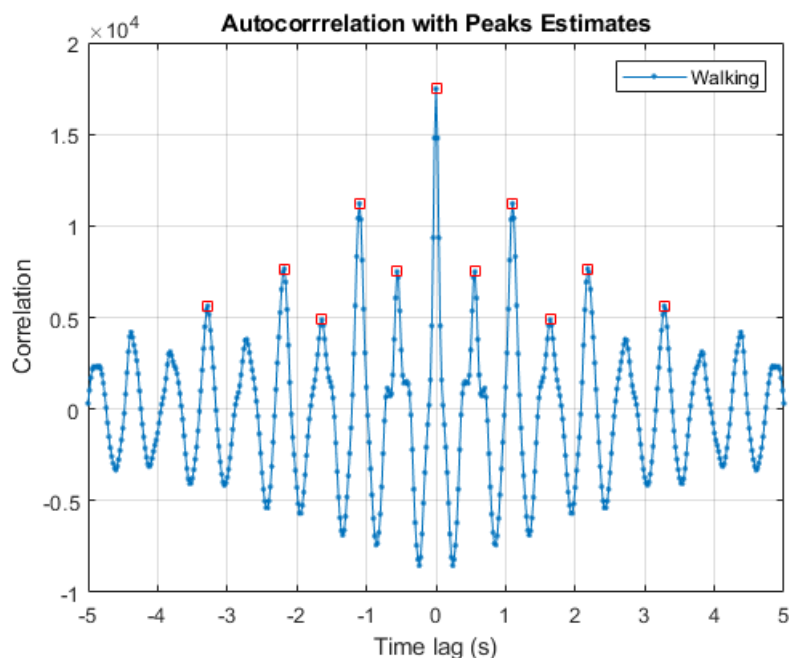


Figure 4.3: Autocorrelation for the activity: Walking

Figure 4.3 illustrates the autocorrelation for the activity of walking. In Figure 4.4 we can see a relative comparison between the two activities of walking upstairs and downstairs. It is evident from the figure that different activities can result in different autocorrelation features from the recorded acceleration data. In order to take autocorrelation as a feature, we took three features from the data of each axis. Which are -

1. The peak height at lag = 0
2. The position of the first peak
3. The height of the first peak

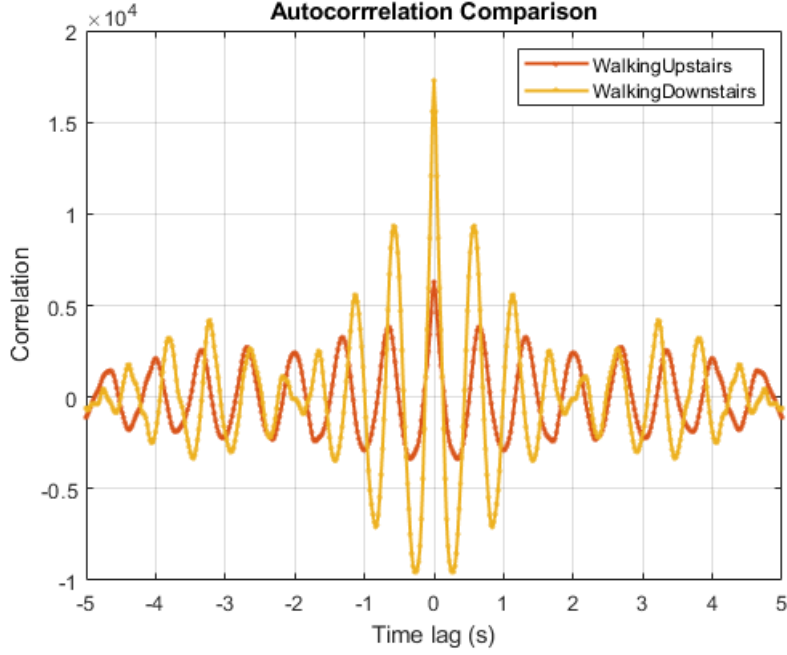


Figure 4.4: Autocorrelation Comparison between Walking Upstairs and Downstairs

## 4.6 Periodogram

If we have a wide-sense stationary random process, which is represented by a signal  $x_n$ , then the periodogram [9] of the signal is basically a nonparametric estimate of the power spectral density (PSD) of the signal. If we take the biased estimate of the autocorrelation of the sequence and take its Fourier Transform, we can obtain the periodogram. If the signal is sampled at  $F_s$  and  $\Delta t$  represents the sampling interval, the periodogram is represented by the following equation -

$$\hat{P}(f) = \frac{\Delta t}{N} \left| \sum_{n=0}^{N-1} x_n e^{-i2\pi f n} \right|^2, \quad -1/2\Delta t < f \leq 1/2\Delta t \quad (4.4)$$

The values at all frequencies are multiplied by 2 except at zero frequency and at the Nyquist frequency, which is  $1/2\Delta t$ . This is done to conserve the total power of the one-sided periodogram.

Representing the frequencies in radians/sample, the periodogram can be represented by -

$$\hat{P}(\omega) = \frac{1}{2\pi N} \left| \sum_{n=0}^{N-1} x_n e^{-i\omega n} \right|^2, \quad -\pi < \omega < \pi \quad (4.5)$$

The integral of the true PSD,  $P(f)$ , over one period,  $1\Delta t$  for cyclical frequency and  $2\pi$  for normalized frequency, is equal to the variance of the wide-sense stationary random process:

The variance of the signal can be calculated by Integrating the true power spectral density of the signal,  $P(f)$  over one period. If the period for cyclical frequency and normalized frequency are respectively  $1\Delta t$  or  $2\pi$ , this can be defined by -



$$\int_{-1/2\Delta t}^{1/2\Delta t} P(f)df \quad (4.6)$$

If the limits are replaced appropriately, for normalized frequencies we have -

$$P_{x_m, M}(\omega_k) = \frac{1}{M} |FFT_{N,k}(x_m)|^2 \triangleq \hat{P}(w) = \frac{1}{M} \left| \sum_{n=0}^{N-1} x_m(n) e^{-j2\pi nk/N} \right|^2 \quad (4.7)$$

The periodogram of the signal is used to generate the spectral power features.

## 4.7 Welch's Power Spectral Density

For a wide-range stationary process, a periodogram does not always provide a constant estimation and contains significant variance. An alternate technique is introduced to reduce the variance of the periodogram. This process, known as Welch's technique[1] uses overlapping method to break the periodogram into a series of time segments.

To produce a better estimate of the PSD, Welch's method calculates the periodogram for each segment. The results of each segment are combined to produce a better estimate of the PSD. This method is wide-sense stationary and utilizes spectral density estimates over distinct segments from the time series. As a result, the approximately uncorrelated estimates of the true spectral density is represented by this modified spectrogram. Thus, variability is reduced by this process.

The segments produced are overlapping by nature. Because of this overlapping nature, the data values at the beginning and at the end of the segment need to be tapered. A window function such as the Hamming Window is multiplied to the segment to obtain this effect. Thus, a modified periodogram is produced by averaging the segments through windowing. This sort of windowing prevents information loss due to overlapping.

In this method, the time signal  $x$  is divided into successive time blocks. Periodograms are calculated from each block and the modified periodogram is generated through averaging the blocks. If the window hop size and number of frames are respectively defined by  $R$  and  $K$  -

$$x + m(n) \triangleq w(n)x(n + mR), n = 0, 1, \dots, M - 1, m = 0, 1, \dots, K - 1 \quad (4.8)$$

We can define the periodogram of the  $m$  th block as follows -

$$P_{x_m, M}(\omega_k) = \frac{1}{M} |FFT_{N,k}(x_m)|^2 \triangleq \hat{P}(w) = \frac{1}{M} \left| \sum_{n=0}^{N-1} x_m(n) e^{-j2\pi nk/N} \right|^2 \quad (4.9)$$

Finally, we can define the Welch estimate of power spectral density as -

$$\hat{S}_x^W \triangleq \frac{1}{K} \sum_{m=0}^{K-1} P_{x_m, M}(\omega_k) \quad (4.10)$$

In simpler terms, Welch’s method averages the periodograms over time. If we use a rectangular window for  $w(n)$ , the non-overlapping successive blocks are used. For other windows like Hamming, Hanning etc., overlapping windows are used.

## 4.8 Spectral Peak Features

Welch’s Power Spectral Estimate is used to extract the spectral peak features. The Welch’s method is used to calculate the power spectral estimates of the signal. We locate the peaks of the spectral estimate and features are extracted from this information. The location and height of the first 6 peaks are taken as features for a single channel. This results in a total of 36 features, 12 from each channel.

Figure 4.5 illustrates the Welch’s Power Spectral density for the activity of walking. The peaks of the signal are also shown clearly in the figure.

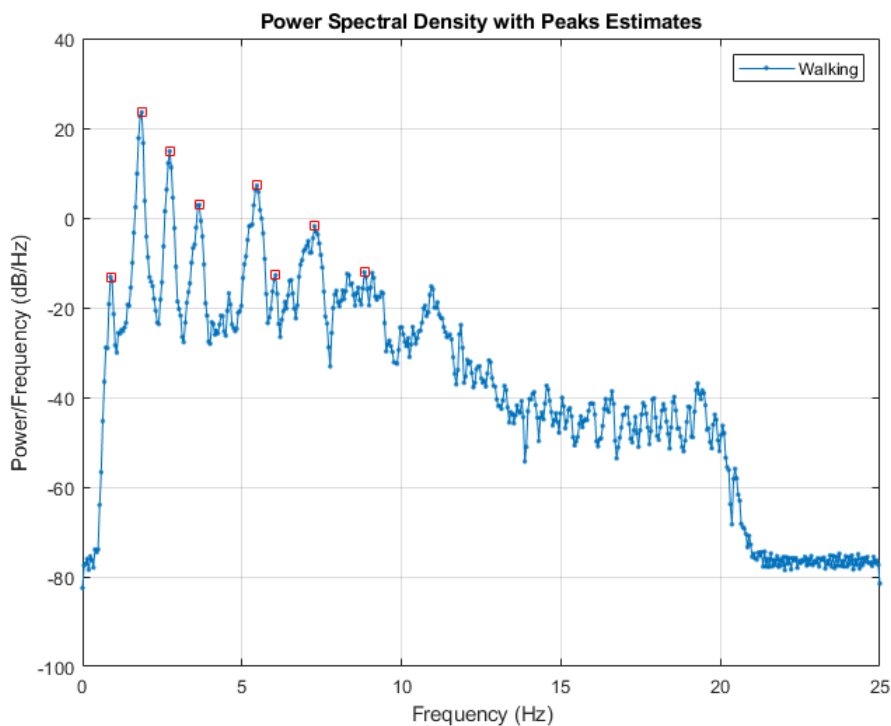


Figure 4.5: Spectral Peak Features

## 4.9 Spectral Power Features

The periodogram is used to calculate the Spectral Power Features of the signal. Spectral power features are calculated in five energy bands of the signal in the intervals of the frequency levels 0.5, 1.5, 5, 10, 15, 20 Hz from the Power Spectral Density. So we get five power spectral features for each channel. The result is a total of 18 power spectral features.

Feature	Features Per Channel	Total Features
Average Value of Signal	1	3
RMS Value of Signal	1	3
Auto-correlation Features	3	3
Spectral Peak Features (PSD)	12	36
Spectral Power Features (PSD)	5	15
Total	22	66

Table 4.1: Summary of Extracted Features

## 4.10 Summary of Extracted Features

A summary of all the extracted features is shown in 4.1

# Chapter 5

## Exploratory Data Analysis

In order to identify patterns or anomalies in the data, to test hypothesis or check assumptions, a wide range of statistical and graphical representations are useful. The process of performing these investigations is known as Exploratory Data Analysis. [40]. For exploratory data analysis on the provided dataset, we followed traditional approaches like time-series visualization and also dimensionality reduction techniques like PCA, t-SNE and LDA. The workflow for Exploratory Data Analysis is illustrated in Figure 5.2.

### 5.1 Time Series Visualization

As discussed earlier, our provided dataset has activities spread into six separate classes. The dataset contained time-series readings of the accelerometer sensor sampled at a rate of 20 samples per second. The dataset was read from the comma-separated-values and loaded into a pandas dataframe. Time series visualizations for different activities were plotted against time. Visualizations for different activities are shown in the following figures.

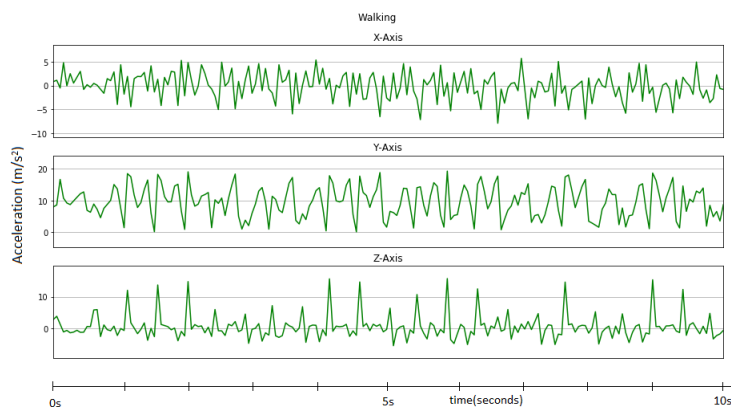


Figure 5.1: Visualization of activity: Walking

The time-series plot of the activity is taken along each axis. Figure 5.1 shows the plot of the activity walking. We can see prominent high frequency components present along the x axis of the signal due to the nature of movement during walking.



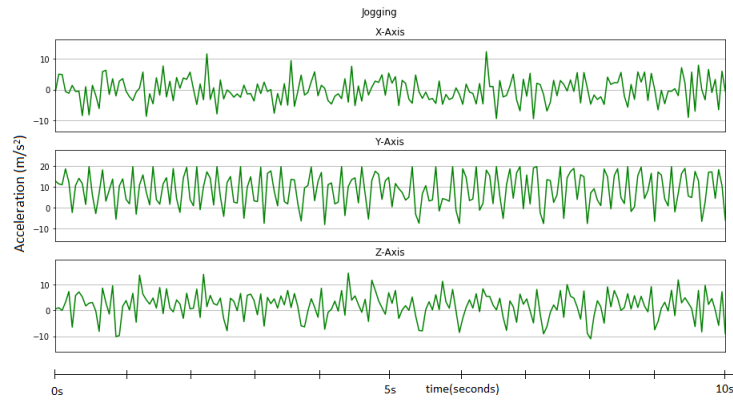


Figure 5.3: Visualization of activity: Jogging

Figure 5.3 shows the plot of the activity jogging. High frequency components are present in both the x and y axis of the signal. It can be seen that jogging can especially be identified by the nature of the y axis signal which shows a periodic component.

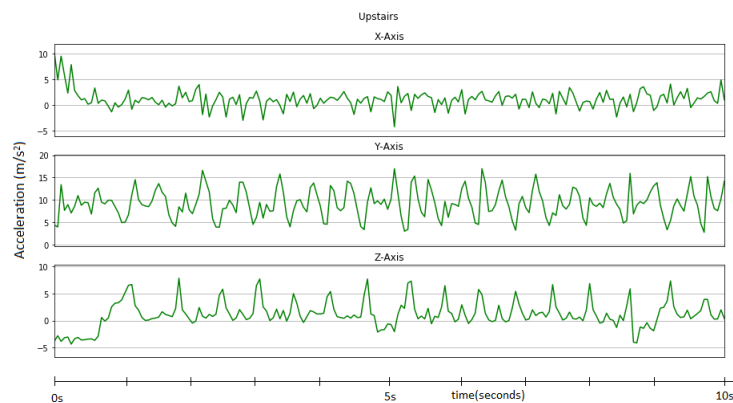


Figure 5.4: Time Series Visualization of activity: Upstairs

Figure 5.4 shows the plot of the activity of walking upstairs. Compared to the activity walking, a major difference can be seen in the z axis component which shows certain peaks in the signal. These peaks correspond to movements resulting from steps between stairs.

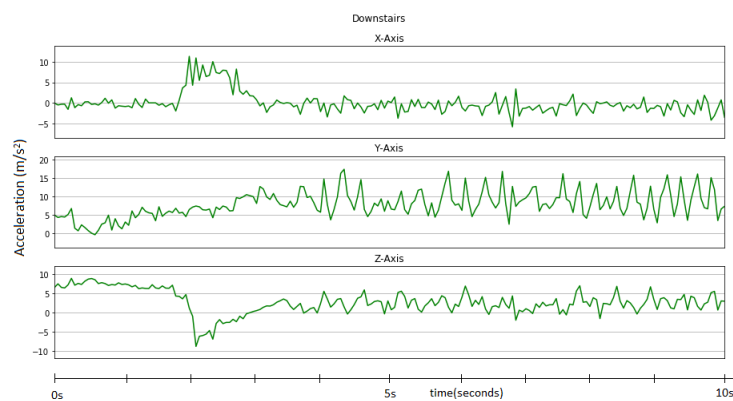


Figure 5.5: Visualization of activity: Downstairs

Figure 5.5 shows the plot of the activity of walking downstairs. The plot for waling downstairs is somewhat comparable to the activity of walking downstairs. However, the peaks resulting along the z axis are different in comparison.

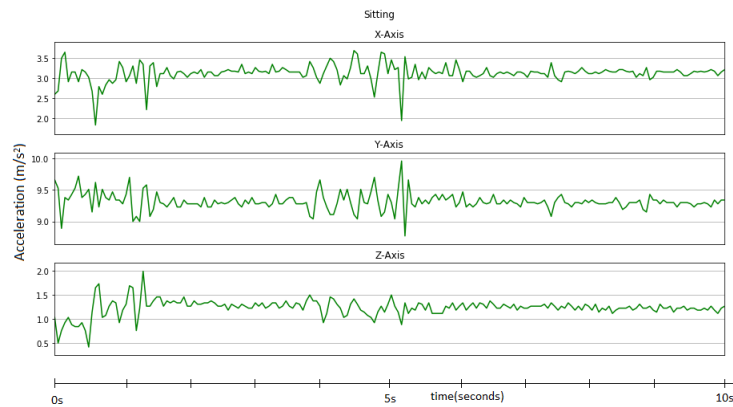


Figure 5.6: Visualization of activity: Sitting

Figure 5.6 shows the plot of the activity of sitting. This activity is relatively easy to differentiate since minor periodic components are present in the signal.

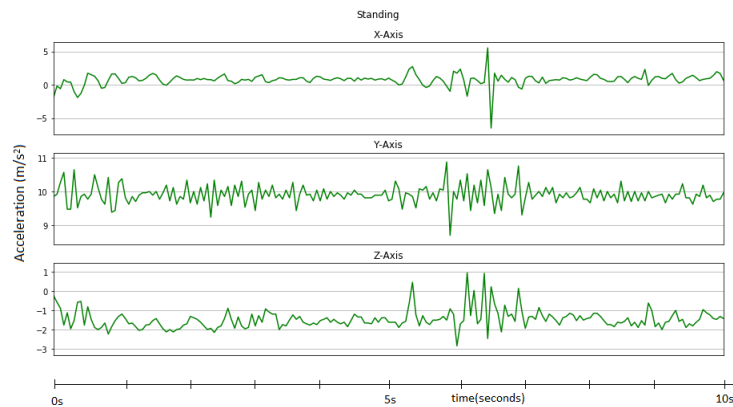


Figure 5.7: Visualization of activity: Standing

Figure 5.7 shows the plot of the activity of standing. This activity might be comparable to the activity standing since they both have minor periodic components. However, the slight variation in y axis might help us to differentiate the signals.

We can conclude from the time-series visualization that there is enough information present in the signal which will likely enable us to predict different activities from the time series data.

## 5.2 Dimensionality Reduction

### 5.2.1 PCA

Principal component analysis(PCA)[48] is a statistical process which allows higher dimensional data to be represented in lower dimension values. These values are known as principal components. On the assumption that the observations contain

correlated variables, PCA utilizes an orthogonal transformation which converts these variables into a set of linearly uncorrelated principal components.

PCA reduction was applied on the separated frames so that we could gain more insight of the correlation between the variables. After reducing the dataset to frames, we were left with 532 frames. Each frame consisted of 80 samples of acceleration data from three different channels. Since PCA is unable to deal with multi-dimensional data, we had to separately apply PCA on the data from the three channels of the sensor- the x channel, the y channel and the z channel. Each channel was reduced to two components for visualization.

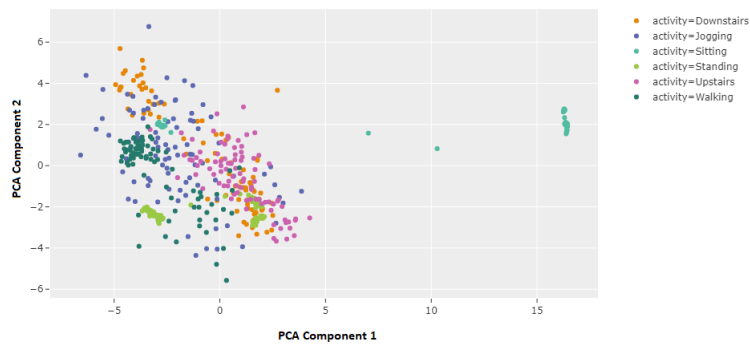


Figure 5.8: PCA on x axis data

Figure 5.8, 5.9 and 5.10 respectively show the PCA visualization on the x, y and z axis acceleration data.

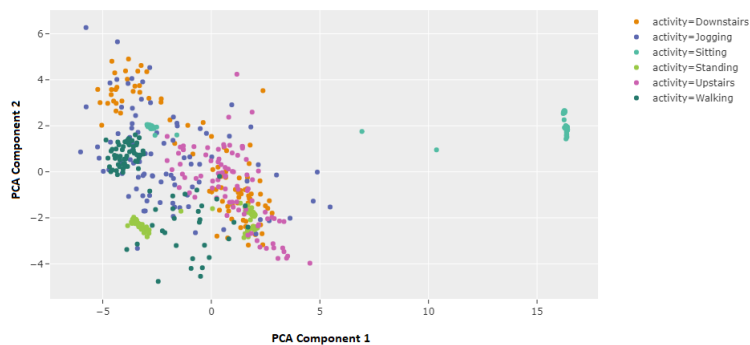


Figure 5.9: PCA on y axis data



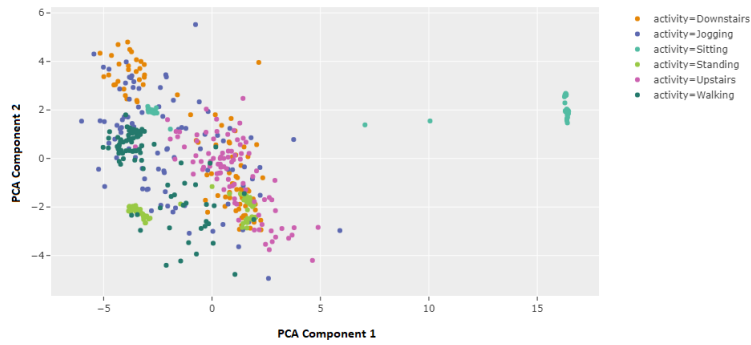


Figure 5.10: PCA on z axis data

## 5.2.2 t-SNE

t-Distributed Stochastic Neighbor Embedding (t-SNE)[12] is another statistical technique which reduces the dimensionality of data. Compared to PCA, t-SNE is more suited to perform on higher dimensional datasets where visualization is necessary. The Barnes-Hut approximations is used for t-SNE, making it practical for large real-world datasets.

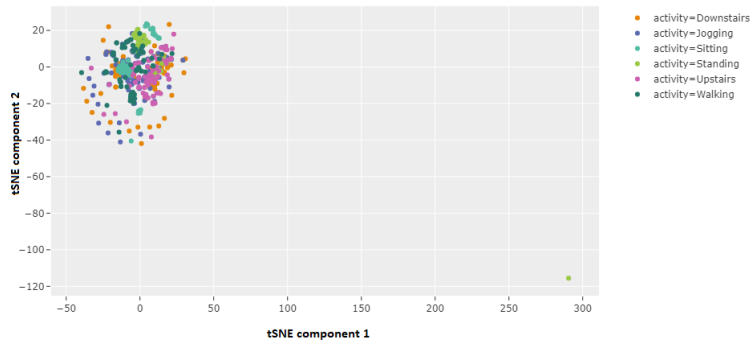


Figure 5.11: t-SNE Reduction on x axis data

We applied t-SNE on the extracted frames using the python implementation *Multi-core t-SNE* [35] using the Barnes-Hut algorithm [30]. The reduction algorithm was separately applied to x, y and z acceleration channels. The reduced t-SNE components were plotted using the python library *plotly express*. Results of the t-SNE reconstructions are illustrated in Figure 5.11, 5.12 and 5.13.

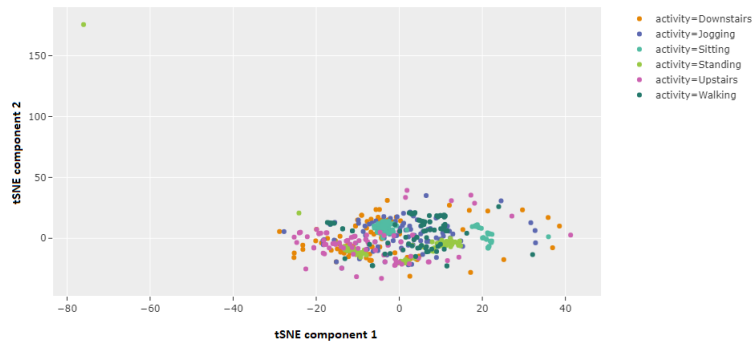


Figure 5.12: t-SNE Reduction on y axis data

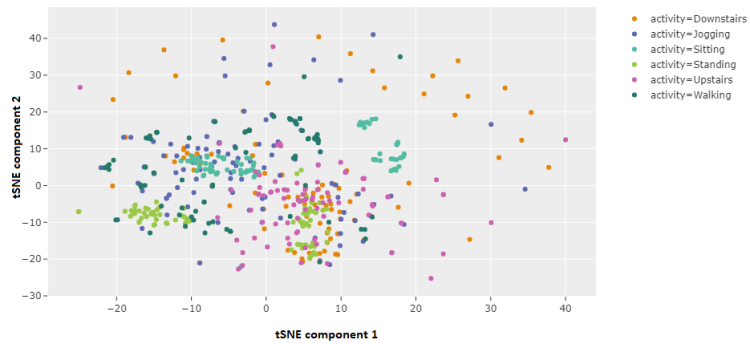


Figure 5.13: t-SNE Reduction on z axis data

### 5.2.3 LDA

Another dimensionality reduction technique which we utilized is Linear Discriminant Analysis (LDA)[46]. The function of LDA is also reducing the dimensionality of the dataset but at the same time it focuses on retaining relevant information.

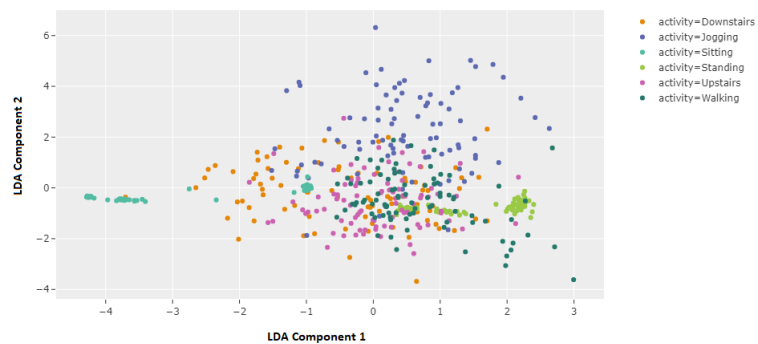


Figure 5.14: LDA Reduction on x axis data

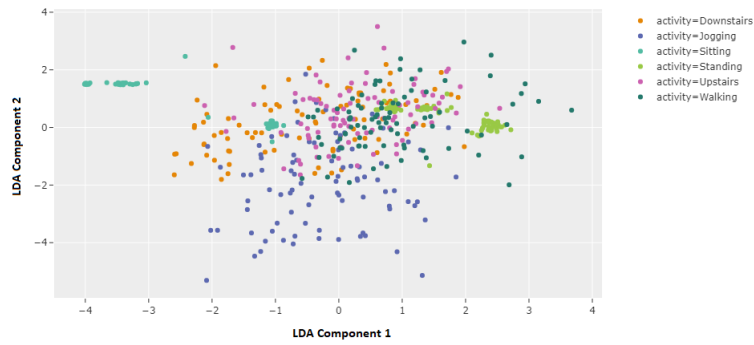


Figure 5.15: LDA Reduction on y axis data

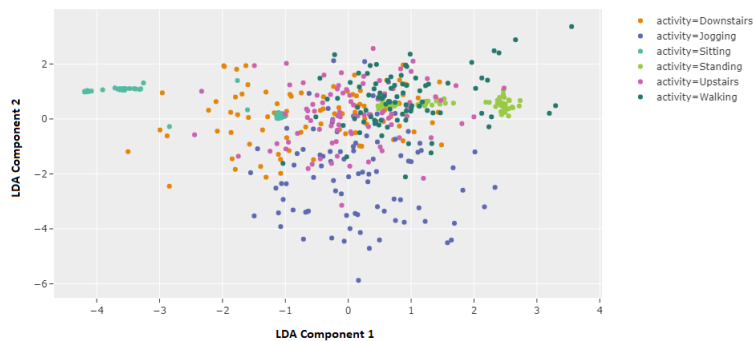


Figure 5.16: LDA Reduction on z axis data

LDA was also applied to the 532 frames generated from the time-series data. The frames were reduced to two-components and scatter plots were produced from the reduction. Figure 5.14, 5.15 and 5.16 respectively show the plots produced from LDA reduction

### 5.2.4 Summary of dimensionality Reduction plots

As we know, when any higher dimensional signal is projected to a lower dimension, the distinction between classes become more visible. PCA, t-SNE and LDA are all unsupervised learning techniques. The results of these techniques provide us valuable insights as to how good a classifier might perform on the select data. The outputs of the reduced dimensionality also help us choose appropriate classifiers for accurate classification of the data.

We can see from the plots of the frames that in most cases of PCA, LDA and t-SNE reduction of the image, the classes in the reduced dimensionality plots are not totally random, but moderately clustered. This suggests that it might be possible to classify the data robustly using the help of a classifier. However, the data is never clearly clustered. So we might conclude that it would be better to employ neural networks rather than statistical machine learning models to get a better classification performance on the data.

# Chapter 6

## Proposed Pipeline

As discussed in the earlier chapter, we extracted a number of features from the time-series accelerometer signals. In order to get a robust performance for the classifier, we decided to combine the time-series data and the features together and then run a machine learning classifier on the combined feature set. We use a Convolutional Neural Network in order to classify the feature set.

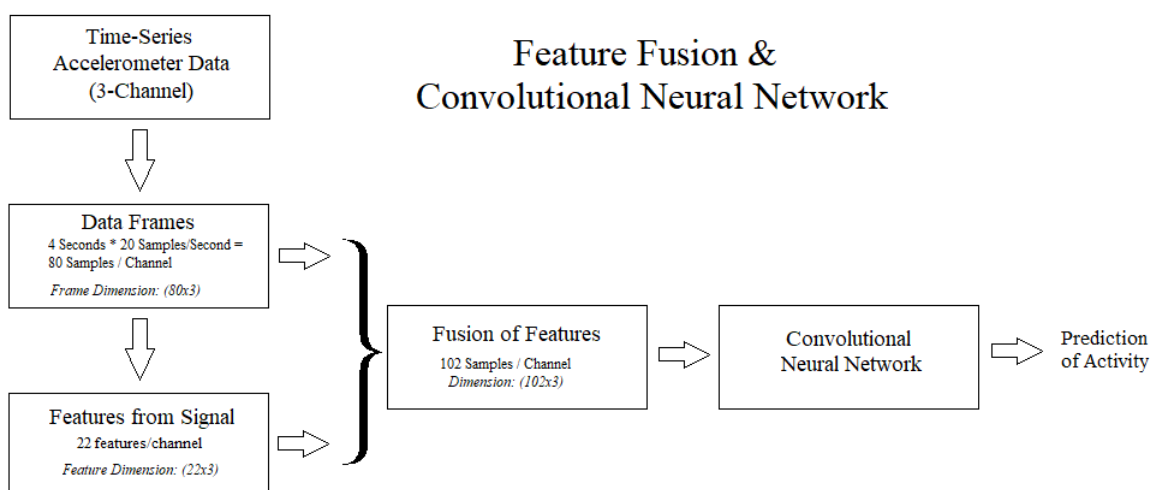


Figure 6.1: Overview of Proposed Pipeline

### 6.1 Fusion of Features with Time-Series Data

To preserve the time-series information of the signal, The balanced dataset containing 21330 samples was converted into frames having a sample size of 80. This resulted in a total of 532 frames, each having 3 channels for the three axis acceleration reading. The shape of each frame array is  $80 \times 3$ .

On the other hand, the features we extracted using signal processing techniques contained a vector of 66 features, 22 from each channel. The only way these two features could be combined preserving spatial and temporal information is if we concatenated the extracted feature of a single channel at the end of its time-series



## 6.2 Convolutional Neural Network Architecture

### 6.2.1 Inspiration

The combined feature set discussed in the previous section has a shape of  $102 \times 3$ . Since, the total number of frames is 532, the resulting data sample has a dimension of  $532 \times 102 \times 3$ . As we can see, the resulting data is multidimensional. The use of statistical machine learning classifiers in this sort of data would require flattening the multidimensional array which could cause a loss of spatial and temporal information of the data. The same issue is faced in image processing applications when a machine learning classifier is applied on a 2D image. In order to overcome this problem, 2D convolution [22] is used in image processing [23], [4] which preserves the spatial information present in image. We extend this idea to use 2D convolution in our classifier. Therefore, we decided to utilize a Convolutional Neural Network (CNN) as our classifier architecture.

### 6.2.2 Architecture Design

Our proposed convolutional Neural Network Architecture was designed using the python deep learning framework Tensorflow[32]. The Keras[31] Sequential model was used to design the architecture.

The model architecture is illustrated in Table 6.1 and Figure 6.4. The architecture contained two ( $2 \times 2$ ) 2D convolutional layers with 16 and 32 filters respectively. Both convolutional layers were followed by a dropout?? of 1% and 2% respectively. A flattening operation followed by a dense hidden layer succeeds the convolutional layers. A a dropout layer of 50% dropout is added after the dense layer. The dropout layers are added to reduce overfitting of the model.

Table 6.1: Model Summary

Layer	Properties	Shape (n=no of frames)	Param #
Input		( n, 102, 3)	
Conv2D	Filters = 16	( n, 101, 2, 16)	80
Dropout	Dropout = 0.1	( n, 101, 2, 16)	
Conv2D	Filters = 32	( n, 100, 1, 32)	2080
Dropout	Dopout = 0.2	( n, 100, 1, 32)	
Flatten		( n, 3200)	
Dense	Neurons = 64	( n, 64)	204864
Dropout	Dropout = 0.5	( n, 64)	
Dense	Neuurons = 6	( n, 6)	390

## CNN Architecture

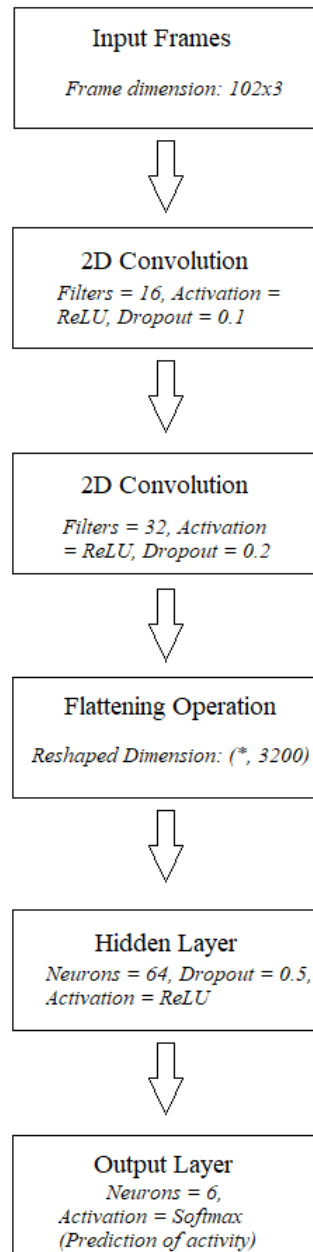


Figure 6.4: Combining Features

The final layer of the model is a dense layer containing six neurons corresponding to the six different activities which we intend to classify. Rectified Linear Unit (ReLU)[38] activation is used for the activation of all the layers except the final layer, where *softmax* activation is used for categorical classification.

## 6.3 Training and Results

### 6.3.1 Loss Metric

We use categorical crossentropy[50] as the loss function to train our model. Categorical crossentropy is applied when performing single label classification or tasks. This categorization task signifies that there is only a single category associated with each data point. One sample in such a dataset might only belong to a single class and no more than that. Categorical crossentropy is denoted by -

$$L(y, \hat{y}) = - \sum_{j=0}^M \sum_{i=0}^N (y_{ij} * \log(\hat{y}_{ij})) \quad (6.1)$$

where  $\hat{y}$  is the predicted value.

The objective of categorical crossentropy is to compare the distribution of the predictions with the true value of the distribution. The distribution of this data is required to have a typical representation where 0 corresponds to the negative classes and 1 corresponds to the true classes. This representation is known as one-hot encoding. The loss is closely related to the model's outputs. If the outputs are more similar to the true vector, the loss is lower.

### 6.3.2 Optimizer

We use the *Adam* [29] optimizer to optimize our convolutional neural network architecture. Classical stochastic gradient descent was once an established method of training any neural network i.e. updating the network weights over every iteration. In this process, a fixed learning rate  $\alpha$  is used throughout the training procedure and it is not updated. However, this fixed rate imposes certain shortcomings to the training of the network. Adam is an optimization algorithm which can be used as an alternate to the classical stochastic gradient descent algorithm. Instead of maintaining a static learning rate, Adam has an adaptive learning rate which is separately updated for each network parameter as the learning continues. Adam utilizes a more efficient optimization algorithm by utilizing the conveniences of the following two optimization algorithms:

RMSProp is another optimization technique which also utilizes per-parameter learning rates. However, these learning rates are updated with respect to the rate of change of the gradient. In other words, the change in magnitude of the gradients. RMSProp proves to be more efficient with online and non-stationary data.

- **Adaptive Gradient Algorithm** Adagrad [21] is a stochastic optimization technique which incorporates a per-parameter learning rate instead of a global learning rate. Adagrad is more effective in improving performance with problems which contain sparse gradients.
- **Root Mean Square Propagation** RMSProp [25] is another optimization technique which also utilizes per-parameter learning rates. However, these learning rates are updated with respect to the rate of change of the gradient. In other words, the change in magnitude of the gradients. RMSProp proves to be more efficient with online and non-stationary data.



As stated earlier, Adam brings together the advantages of both techniques. RMSProp utilizes the alpha of the network parameters based on the average first moment or the mean. However, Adam leverages the average second moments of the gradients instead of the mean, which is the uncentered variance of the data. The exponential moving average of the gradient and the squared gradients are computed in this process. Two rates  $\beta_1$   $\beta_2$  are used for controlling the decay rates of these moving averages.

For our implementation, we took the learning rate of the Adam optimizer 0.001 ,  $\beta_1 = 0.9$  and  $\beta_2 = 0.999$

### 6.3.3 Training

The combined feature frames discussed earlier are used to train the CNN architecture. Before training the network, the data is split into two subsets, first for training the model and the other for validating the CNN. The train-test ratio is 80-20 for the dataset.

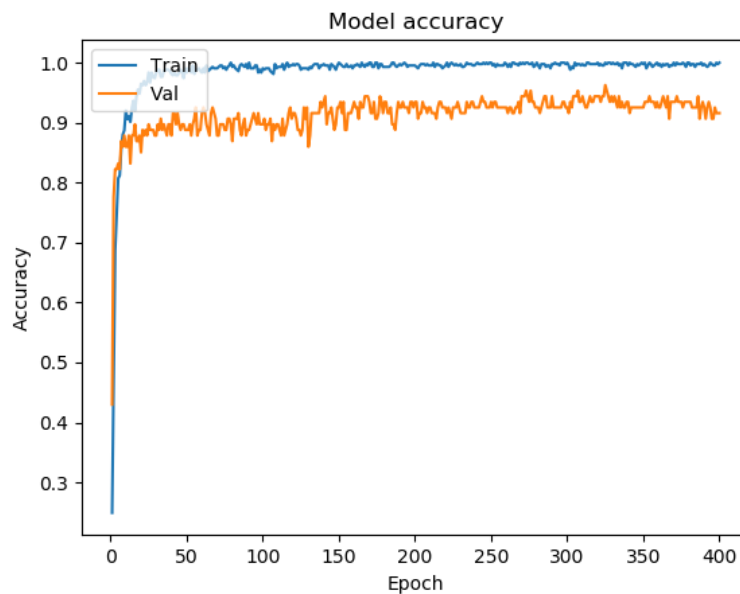


Figure 6.5: Training Accuracy

The neural network was trained using a high-performance desktop PC with GPU acceleration. The network was trained for 400 epochs on the training data and each epoch was validated using the test data. The model with the best validation accuracy is selected as our desired model.

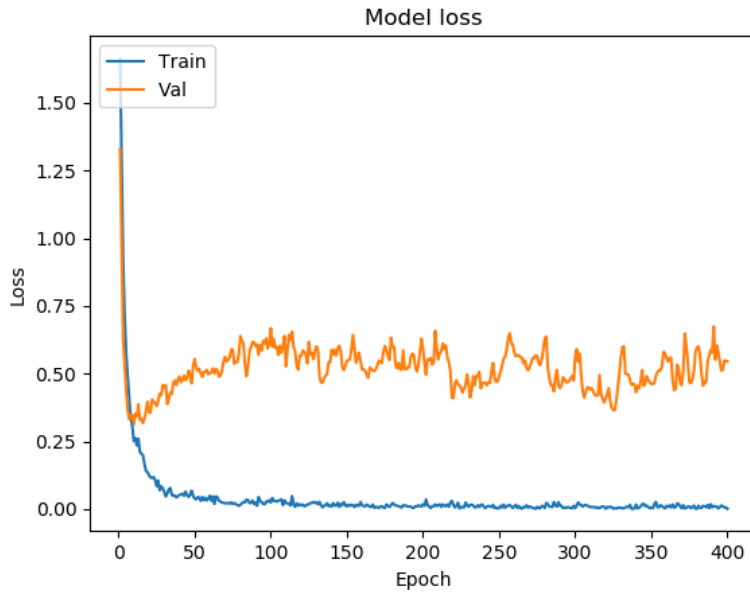


Figure 6.6: Training Loss

The training accuracy and training loss is shown in Figure 6.5 and 6.6 respectively. It is evident from the training curves that the model converges around 200 epochs of training. Further training has little improvement in model performance. The maximum validation accuracy for the recorded model is found to be 96.26%.

### 6.3.4 Confusion Matrix

The confusion matrix for the validation data is shown in Figure 6.7. As we can see from the plot of the confusion matrix, the model is performing robustly on all the classes.

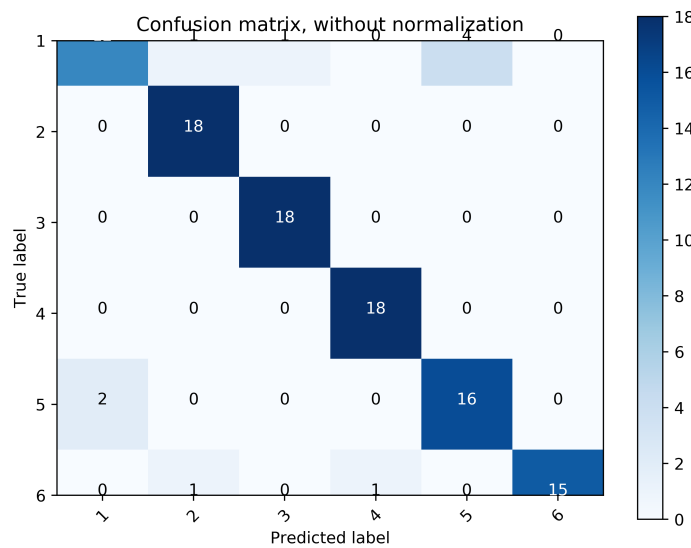


Figure 6.7: Confusion Matrix without Normalization

# Chapter 7

## Conclusion

### 7.1 Conclusion

Human activity recognition from wearable device sensors alone is a complicated task if only traditional signal processing techniques or only machine learning techniques are used. Since the data from the accelerometer sensors are temporal in nature, traditional machine learning classifiers which work with vectorized data points are not suitable for proper classification of this sort of data. At the same time, since the accelerometer sensors are one-dimensional electrical signals, existing signal processing techniques like signal mean, rms, spectral peak features and power spectrum estimation should add valuable information to the classification pipeline. This hypothesis is verified by our work for this thesis. Our approach incorporated a fusion of traditional signal processing features as well as advanced machine learning algorithms like convolutional neural networks. Combining both techniques, our classifier performed exceedingly well on the WISDM dataset and yielded a prediction accuracy of 96.26% on the testing set. Our model is also lightweight and takes less than a second to properly predict each frame of accelerometer sensor data. Therefore, our proposed pipeline is suitable for real-world applications where it can accurately predict human activity from the sensor from any easily accessible device like a smartphone.

### 7.2 Future Scope

Our prediction model is capable of robustly and efficiently classifying human activity from wearable body sensor data. This prediction methodology might find its way to a smartphone, smartwatch or even a wearable smart device which has an accelerometer sensor. This has the potential to contribute to various sectors, namely healthcare. Monitoring of elderly patients, infants, patients with disabilities or mental disorders is possible if this methodology is implemented on-device. This technology is not only capable to track and monitor patients with ongoing treatment but also it may have the potential to save lives by alerting about patients with disabilities or medical conditions in the case of an emergency. An example of this application might be fall detection of an elderly person or a patient. The military and security applications of this methodology are also very promising. Monitoring the status, activity and health of soldiers is important for military applications to ensure their safety and performance as well as aiding strategic decisions is possible through HAR. Our research work, if implemented properly, opens up endless possibilities.

# Bibliography

- [1] J. O. Smith, *Spectral Audio Signal Processing*. <http://ccrma.stanford.edu/~jos/sasp/>, accessed `{date}`, online book, 2011 edition.
- [2] T. Cogley and J. M. Nason, “Effects of the hodrick-prescott filter on trend and difference stationary time series implications for business cycle research”, *Journal of Economic Dynamics and control*, vol. 19, no. 1-2, pp. 253–278, 1995.
- [3] R. J. Hodrick and E. C. Prescott, “Postwar us business cycles: An empirical investigation”, *Journal of Money, credit, and Banking*, pp. 1–16, 1997.
- [4] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back, “Face recognition: A convolutional neural-network approach”, *IEEE transactions on neural networks*, vol. 8, no. 1, pp. 98–113, 1997.
- [5] F. Foerster, M. Smeja, and J. Fahrenberg, “Detection of posture and motion by accelerometry: A validation study in ambulatory monitoring”, *Computers in Human Behavior*, vol. 15, no. 5, pp. 571–583, 1999.
- [6] M. M. Daniel and A. C. Singer, *Computer explorations in signals and systems using MATLAB*. Upper Saddle River, NJ: Prentice Hall, 2002.
- [7] M. O. Ravn and H. Uhlig, “On adjusting the hodrick-prescott filter for the frequency of observations”, *Review of economics and statistics*, vol. 84, no. 2, pp. 371–376, 2002.
- [8] P. Stoica, R. L. Moses, *et al.*, “Spectral analysis of signals”, 2005.
- [9] S. A. Fulop and K. Fitz, “Algorithms for computing the time-corrected instantaneous frequency (reassigned) spectrogram, with applications”, *The Journal of the Acoustical Society of America*, vol. 119, no. 1, pp. 360–371, 2006.
- [10] J. Choi, Y.-i. Cho, T. Han, and H. Yang, “A view-based real-time human action recognition system as an interface for human computer interaction”, Sep. 2007, pp. 112–120. DOI: 10.1007/978-3-540-78566-8\_10.
- [11] Z.-Y. He and L.-W. Jin, “Activity recognition from acceleration data using ar model representation and svm”, in *2008 international conference on machine learning and cybernetics*, IEEE, vol. 4, 2008, pp. 2245–2250.
- [12] L. v. d. Maaten and G. Hinton, “Visualizing data using t-sne”, *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.
- [13] J. Yin, Q. Yang, and J. J. Pan, “Sensor-based abnormal human-activity detection”, *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 8, pp. 1082–1090, 2008.

- [14] Y. Hanai, J. Nishimura, and T. Kuroda, “Haar-like filtering for human activity recognition using 3d accelerometer”, in *2009 IEEE 13th Digital Signal Processing Workshop and 5th IEEE Signal Processing Education Workshop*, IEEE, 2009, pp. 675–678.
- [15] Z. He and L. Jin, “Activity recognition from acceleration data based on discrete cosine transform and svm”, in *2009 IEEE International Conference on Systems, Man and Cybernetics*, IEEE, 2009, pp. 5041–5044.
- [16] Y. Jia, “Dietetic and exercise therapy against diabetes mellitus”, in *2009 Second International Conference on Intelligent Networks and Intelligent Systems*, IEEE, 2009, pp. 693–696.
- [17] E. Kim, S. Helal, and D. Cook, “Human activity recognition and pattern discovery”, *IEEE pervasive computing*, vol. 9, no. 1, pp. 48–53, 2009.
- [18] S. Boughorbel, J. Breebaart, F. Bruekers, I. Flinsenbergh, and W. Kate, “Child-activity recognition from multi-sensor data”, Jan. 2010. DOI: 10.1145/1931344.1931382.
- [19] A. J. Perez, M. A. Labrador, and S. J. Barbeau, “G-sense: A scalable architecture for global sensing and monitoring”, *IEEE Network*, vol. 24, no. 4, pp. 57–64, 2010.
- [20] Y. Xue and L. Jin, “A naturalistic 3d acceleration-based activity dataset & benchmark evaluations”, in *2010 IEEE International Conference on Systems, Man and Cybernetics*, IEEE, 2010, pp. 4081–4085.
- [21] J. Duchi, E. Hazan, and Y. Singer, “Adaptive subgradient methods for online learning and stochastic optimization”, *Journal of Machine Learning Research*, vol. 12, no. Jul, pp. 2121–2159, 2011.
- [22] G. Hinton, L. Deng, D. Yu, G. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, B. Kingsbury, *et al.*, “Deep neural networks for acoustic modeling in speech recognition”, *IEEE Signal processing magazine*, vol. 29, 2012.
- [23] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks”, in *Advances in neural information processing systems*, 2012, pp. 1097–1105.
- [24] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, “A public domain dataset for human activity recognition using smartphones.”, in *Esann*, 2013.
- [25] A. Graves, *Generating sequences with recurrent neural networks*, 2013. arXiv: 1308.0850 [cs.NE].
- [26] D. Leightley, J. Darby, B. Li, J. Mcphee, and M. H. Yap, “Human activity recognition for physical rehabilitation”, Dec. 2013. DOI: 10.1109/SMC.2013.51.
- [27] V. T. Van Hees, L. Gorzelniak, E. C. D. Leon, M. Eder, M. Pias, S. Taherian, U. Ekelund, F. Renström, P. W. Franks, A. Horsch, *et al.*, “Separating movement and gravity components in an acceleration signal and implications for the assessment of human daily physical activity”, *PloS one*, vol. 8, no. 4, e61691, 2013.

- [28] A. Bayat, M. Pomplun, and D. A. Tran, “A study on human activity recognition using accelerometer data from smartphones”, *Procedia Computer Science*, vol. 34, pp. 450–457, 2014.
- [29] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization”, *arXiv preprint arXiv:1412.6980*, 2014.
- [30] L. Van Der Maaten, “Accelerating t-sne using tree-based algorithms”, *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 3221–3245, 2014.
- [31] F. Chollet *et al.*, *Keras*, <https://keras.io>, 2015.
- [32] Martin Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Y. Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng, *TensorFlow: Large-scale machine learning on heterogeneous systems*, Software available from [tensorflow.org](http://tensorflow.org), 2015. [Online]. Available: <http://tensorflow.org/>.
- [33] A. Taha, H. Zayed, M. Khalifa, and E.-S. El-Horbarty, “Human activity recognition for surveillance applications”, May 2015. DOI: 10.15849/icit.2015.0103.
- [34] C. A. Ronao and S.-B. Cho, “Human activity recognition with smartphone sensors using deep learning neural networks”, *Expert systems with applications*, vol. 59, pp. 235–244, 2016.
- [35] D. Ulyanov, *Multicore-tsne*, <https://github.com/DmitryUlyanov/Multicore-TSNE>, 2016.
- [36] Y. Hbali, S. Hbali, L. Ballihi, and M. Sadgal, “Skeleton-based human activity recognition for elderly monitoring systems”, *IET Computer Vision*, vol. 12, Aug. 2017. DOI: 10.1049/iet-cvi.2017.0062.
- [37] S. Zhang, Z. Wei, J. Nie, L. Huang, S. Wang, and Z. Li, “A review on human activity recognition using vision-based method”, *Journal of Healthcare Engineering*, vol. 2017, pp. 1–31, Jul. 2017. DOI: 10.1155/2017/3090343.
- [38] A. F. Agarap, “Deep learning using rectified linear units (relu)”, *arXiv preprint arXiv:1803.08375*, 2018.
- [39] *MATLAB version 9.5.0.944444 (R2018b)*, The Mathworks, Inc., Natick, Massachusetts, 2018.
- [40] P. Patil, *What is exploratory data analysis?*, May 2018. [Online]. Available: <https://towardsdatascience.com/exploratory-data-analysis-8fc1cb20fd15>.
- [41] A. Shdefat, A. Halimeh, and H. Kim, “Human activities recognition via smartphones using supervised machine learning classifiers”, *Prim Health Care*, vol. 8, no. 289, pp. 2167–1079, 2018.
- [42] T. Wenge, M. T. Chew, F. Alam, and G. Gupta, “Implementation of a visible light based indoor localization system”, Mar. 2018, pp. 1–6. DOI: 10.1109/SAS.2018.8336711.

- [43] D. Arifoglu and H. Bouchachia, “Detection of abnormal behaviour for dementia sufferers using convolutional neural networks”, *Artificial Intelligence in Medicine*, vol. 94, Feb. 2019. DOI: 10.1016/j.artmed.2019.01.005.
- [44] M. Badshah, “Sensor-based human activity recognition using smartphones”, 2019.
- [45] A. Farooq and S. Kamal, “Indoor positioning and tracking using sensors of a smart device”, Aug. 2019. DOI: 10.1109/ICAEM.2019.8853812.
- [46] C. Maklin, *Linear discriminant analysis in python*, Aug. 2019. [Online]. Available: <https://towardsdatascience.com/linear-discriminant-analysis-in-python-76b8b17817c2>.
- [47] D. Smith, *How does an accelerometer work in a smartphone? bill hammack, the engineer guy, explains [full text]*, Mar. 2019. [Online]. Available: <https://www.ibtimes.com/how-does-accelerometer-work-smartphone-bill-hammack-engineer-guy-explains-full-text-699762>.
- [48] Wikipedia contributors, *Principal component analysis — Wikipedia, the free encyclopedia*, [Online; accessed 9-December-2019], 2019. [Online]. Available: [https://en.wikipedia.org/w/index.php?title=Principal\\_component\\_analysis&oldid=928372596](https://en.wikipedia.org/w/index.php?title=Principal_component_analysis&oldid=928372596).
- [49] *Accelerometers: What they are how they work*. [Online]. Available: <https://www.livescience.com/40102-accelerometers.html>.
- [50] *Categorical crossentropy*. [Online]. Available: <https://peltarion.com/knowledge-center/documentation/modeling-view/build-an-ai-model/loss-functions/categorical-crossentropy>.