

# **HUMAN ACTIVITY RECOGNITION USING SMARTPHONE DATASET**

Project report submitted in partial fulfillment of the requirement for the  
degree of Bachelor of Technology  
in  
**Computer Science and Engineering/Information Technology**

By

Anchal Sharma (171291)  
Chirag Arora (171273)

Under the supervision of  
Prof. Arvind Kumar

To



Department of Computer Science & Engineering and Information  
Technology

**Jaypee University of Information Technology,  
Waknaghat, Solan-173234, Himachal Pradesh**

## Candidate's Declaration

I hereby declare that the work presented in this report entitled “ **Human activity recognition using smartphone dataset**” in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering/Information Technology** submitted in the department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology, Wanknaghat is an authentic record of our own work carried out over a period from January 2021 to June 2021 under the supervision of **Prof. Arvind Kumar** (Associate Professor, Department of computer science and engineering).

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

  
(Student Signature)


Anchal Sharma, 171291



(Student Signature)

Chirag Arora, 171273

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

  
(Supervisor Signature)

Prof. Arvind Kumar

Associate Professor

Department of computer science and engineering

Dated: 18.05.2021

## **Acknowledgement**

We have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. We would like to extend our sincere thanks to all of them.

We are highly indebted to Prof. Amit Kumar for his guidance and constant supervision as well as for providing necessary information regarding the project & also for the support in completing the project.

We would like to express our gratitude towards our parents for their kind co-operation and encouragement which help me in completion of this project.

I would like to express my special gratitude and thanks to lab attends for giving me such attention and time.

My thanks and appreciations also go to my colleague in developing the project and people who have willingly helped me out with their abilities.

# Contents

<b>Title</b>	<b>Page No</b>
List of Abbreviations	iv
List of Figures	v
List of Graphs	vi
List of Tables	vi
Abstract	vii
Chapter 1 Introduction	1-3
Chapter 2 Literature survey	4-9
Chapter 3 System development	10-19
Chapter 4 Performance Analysis	20-25
Chapter 5 Conclusions	26
References	27-30
Appendix	31-37

## **List of Abbreviations**

- ❖ PCA- Principal Component Analysis
- ❖ UCI- University of California, Irvine
- ❖ HAR- Human Activity Recognition
- ❖ NA- Not Applicable
- ❖ EDA- Exploratory Data Analysis
- ❖ CSV- Comma separated values
- ❖ SVM-Support Vector Machine
- ❖ KNN-K Nearest Neighbor
- ❖ WRT-With Respect to

## List of figures

<b>Title</b>	<b>Page No</b>
Figure 1: HAR development Process	4
Figure 2: Protocol of activities for the HAR Experiment	5
Figure 3: Axis of angular rotation for gyroscopes	6
Figure 4: Flow of System Design of Human Activity Recognition	10
Figure 5: Snapshot of cleaned Dataset	12
Figure 6: A sample tree of the forest constructed	16

## List of graphs

<b>Title</b>	<b>Page No</b>
Graph 1: Proportions of Variances wrt Principal Component	14
Graph 2: Cumulative Proportions of Variances	14
Graph 3: Error rate measurements w.r.t to Number of Trees	15
Graph 4: Variation of error rate w.r.t value of K	17
Graph 5: Variation of error rate w.r.t number of nodes in hidden layer	19
Graph 6: Variation of success rate in Random Forest Model	21
Graph 7: Variation of success rate in K Nearest Neighbor	22
Graph 8 Variation of success rate in Support Vector Machine	24
Graph 9: Variation of success rate in Artificial Neural Network	25

## List of tables

<b>Title</b>	<b>Page No</b>
Table 1: Confusion matrix of data tested with Random Forest Model	20
Table 2: Confusion matrix of data tested with K Nearest Neighbor Model	22
Table 3: Confusion matrix of data tested with Support Vector Machine Model	23
Table 4: Confusion matrix of data tested with Artificial Neural Network	24



## **Abstract**

This project depicts recognition of Human activities using data generated from user's Smart phone. We have used data available at University of California Machine Learning repository to recognize six human activities. These activities are Standing, Sitting, Laying, Walking, walking upstairs and Walking downstairs. Data is collected from embedded accelerometer, gyroscope and other sensors of Samsung Galaxy S II Smart phone. Data is randomly divided into 7:3 ratios to form training and testing data set respectively. Dimensionality reduction is done using Principal Component Analysis technique. Activity classification is done using Machine Learning models namely Random Forest, Support Vector Machine, Artificial Neural Network and K-Nearest Neighbor. We have compared accuracy and performance of these models using confusion matrix and random simulation.

# Chapter 1 INTRODUCTION

## 1.1 Introduction

Research in Human Activity Recognition is in massive demand due to its applications in Health care domain, Computer Vision, Household safety and Robot Learning. A huge amount of money can be saved if sensors collect and monitor data of patients. System can automatically send reports to doctor in case of any abnormal behavior.

We have used sensors of low-cost Smartphones to identify human activities. Extensive growth in popularity, accessibility and computation power of smartphones makes it ideal candidate for non-invasive body attached sensor .

Smartphones have become irreplaceable part of human life. People carry Smartphones throughout the day. This enables smartphone sensors to collect data and hence lets system to detect human activity.

Human Activity Recognition (HAR) aims to identify the actions carried out by a person given a set of observations of him/her and the surrounding environment. Recognition can be accomplished by exploiting the information retrieved from various sources such as environmental or body-worn sensors. Our Aim is to classify the given activities in the form of a dataset into six labels namely sitting, standing, walking, climbing up, climbing down and laying. We present analysis of method for classifying activities, such as walking upstairs or standing, using data from a gyroscope and accelerometer. Analysis is informed by a visualization of the data. We analyze the differences in error rates between different methods.

## 1.2 Problem Statement

Identifying human activities from Smartphone dataset has proved to be complex task due to large dimensions of dataset. Various Machine learning techniques have been used previously to identify human activities. We have proposed a system that reduces dimensions of Smartphone dataset and uses Machine learning algorithms in an optimised manner to produce efficient result.

## 1.3 Objectives

- a. To clean the given dataset.
- b. To study the dataset and reduce its dimensions using PCA.
- c. To identify suitable machine learning algorithm to analyze given dataset.
- d. To analyze the performance of different Machine Learning algorithms using following process-
  - Apply Machine learning algorithms to the cleaned dataset and generate the confusion matrix.
  - Calculate error rate from the confusion matrix.
  - Calculate time taken to train the model.
  - Compare error rate of different Machine Learning algorithms.
  - Compare time taken by different Machine Learning algorithms.

## **1.4 Methodology**

We have collected data from University of California Machine Learning repository [3]. Data is imported, cleaned and normalized. In order to increase correctness and performance of our system we have reduced dimensions of our original dataset using Principal Component Analysis (PCA) technique. Reduced data is then processed through various supervised Machine Learning algorithms like Random Forest, Support Vector Machine, Artificial Neural Network and K-Nearest Neighbor to classify data into six categories namely Sitting, Standing, Laying, Walking, Walking Upstairs and Walking Downstairs. Correctness of the system is determined by generating confusion matrix and by random simulations.

## **1.5 Organization**

Chapter 2 describes previous works associated with our work. Methodology and System Design have been described in Chapter 3. Performance of different Machine Learning Algorithms have been compared in Chapter 4 and Chapter 5 deduces conclusion from the analysis performed.

# CHAPTER 2 LITERATURE SURVEY

## 2.1 Human Activity Recognition

The purpose of human activity recognition is to determine the actions of a person from a given set of data about him or her and his or her environment. Much research is being done in the field of human activity recognition where human behavior is interpreted by diminishing factors found in motion, location, body signals and information from nature etc. The environment and the senses worn by a person produce information that is used to interpret the work. Good accuracy can be found in the nerves worn on the waist, wrist, chest, and thighs. But these sensors do not feel right and cannot provide long-term solutions. [22]

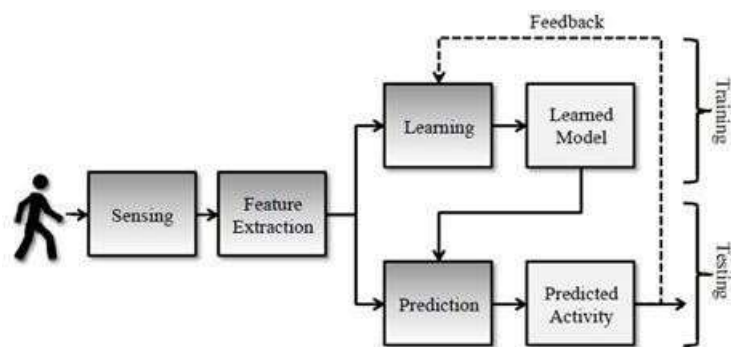


Figure 1: HAR development Process

Smartphones have brought great research opportunities to people-centered programs. The latest smartphones come with a variety of sensors installed such as microphones, cameras, accelerometers, gyroscopes, etc. These available devices provide automatic monitoring and uninterrupted monitoring of daily life activities without the use of telephone services. [1] HAR is part of a larger concept known as the computer that knows the context or the universe

computer use. Computing everywhere does its job in the same way as HAR, which is to collect data from users and help them.

No.	Static	Time (sec)	No.	Dynamic	Time (sec)
0	Start (Standing Pos)	0	7	Walk (1)	15
1	Stand (1)	15	8	Walk (2)	15
2	Sit (1)	15	9	Walk Downstairs (1)	12
3	Stand (2)	15	10	Walk Upstairs (2)	12
4	Lay Down (1)	15	11	Walk Downstairs (1)	12
5	Sit (2)	15	12	Walk Upstairs (2)	12
6	Lay Down (2)	15	13	Walk Downstairs (3)	12
			14	Walk Upstairs (3)	12
			15	Stop	0
				<b>Total</b>	<b>192</b>

Figure 2: Protocol of activities for the HAR Experiment

HAR is widely used in the fields of nursing, military, entertainment, and everyday life. For example, HAR can be used to assist a soldier with reports of his or her actions, unusual personal reporting to hospital staff, etc. Integrated movement sensors are used to provide important information to athletes, which will help them improve their performance. Obviously HAR becomes an important part of our daily lives.

HAR using smartphones has many benefits such as device portability, comfort and uninterrupted sensitivity. Reversing this method consumes and shares services and other applications on the phone which may be problematic for low-powered devices.

## 2.2 Sensors used

### 2.2.1 Accelerometer

An accelerometer is a device used to measure static or dynamic speed. We can determine the angle at which the device is tilted by determining the vertical speed due to gravity. And we can see the direction of travel by measuring the dynamic speed. Accelerometer can work on a piezoelectric effect or by sensing changes in capacitance.

Other less commonly used methods use a piezoresistive effect, hot air bubbles, and light. Piezoelectric accelerometers have small crystals that generate voltage power when accelerated. [24]

The Accelerometer on a smartphone contains silicon earthquake rotation. The quantity changes its shape depending on the device.

### 2.2.2 Gyroscope

Gyroscopes are small and cheap devices that are used to measure rotational angular velocity movements. Degrees per second ( $^{\circ} / s$ ) or changes per second (RPS) units of angular velocity. [25]

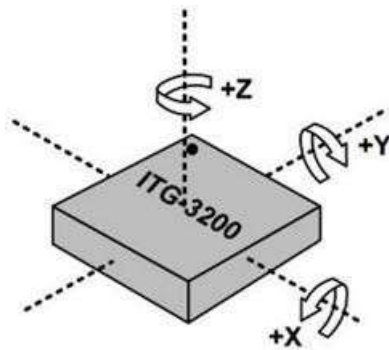


Figure 3: Axis of angular rotation for gyroscopes

The triple axis of the MEMS gyroscope can measure rotation with 3 axes: x, y, and z. gyroscopes with one or two axes are becoming increasingly unattractive as three-axis gyroscopes are cheaper and smaller. Low-level electrical signals are produced, which are later amplified, when the magnitude of the gyroscope repetition changes its position due to rotational motion. MEMS gyroscopes work in mA or sometimes micro-ampere range. [25]

### 2.3 Wearable sensor placement

The placement of the wearable sensors means that the attached nerves are attached where and when. It has a direct impact on the perception made in a person.

Good posture is still the subject of debate. The nerves can be connected to different parts of the body but the sternum, lower back and buttocks are the most common areas. Next to the placement of the nerves in the center of the weight, it is better to represent the movement. The main motivation for researchers is to reduce the number of nerve implants in the body, to maintain a good position and to maintain a high degree of accuracy.

Cleland et al. [30] placed 6 sensors in the chest, hip, wrist, thigh, foot and lower back to determine which area was best suited to collect accurate recognition data. They found that a waist sensor accurately indicates activity. Chamroukhi et al. [31] narrowed their study that the accuracy of detection can be significantly increased by placing accelerometers on both the upper and lower extremities. He also found that the system of 3 sensors placed on the chest, thigh and ankles gave at least some flaws. Also, various experiments have shown that the accuracy of detection increases with the number of nerves attached to a person.

## **2.4 Signal processing**

Tri-axial line acceleration and angular velocity were taken by the accelerometer and gyro of the smartphone at a sample rate of 50Hz. To reduce noise the signals are transmitted through a medium filter and a filter suitable for butter at a suitable cutting frequency of 20Hz as 99% power is limited to 15Hz. Captive signals have two body-generated components and gravity. The gravitational component has a 0.3 Hz corner frequency calculated from previous tests. [22]

## **2.5 Pre-processing**

Data processing is an integral part of data mining. The purpose of pre-processing is to filter data, replace corrupted values and extract or select features. In window technology the sensory signals are divided into smaller parts of the time. Used for debugging. After this, in each separate window, a partition and algorithm layout is used. There are 3 types of windows strategies: (a) sliding (b) event description (c) job description. The most useful process is the delivery windows for real-time applications [29].



## **2.6 Principal Component Analysis (PCA)**

When a database has a large number of symbols, we may encounter the following conditions-

- Many related variables can be found.
- If we decide to use the model in all data it can lead to a higher level of errors.
- We can start using a new method to extract some important attributes.

We have used Key Analysis to reduce the number of features in our database. Key Asset Analysis (PCA) is designed to reduce the size of a large data set that contains a large number of consistent variables and to maintain as much variance as possible in the data set [26]. The core component analysis works with a variable converter to the data provided in a new set of variables, the main components (PCs). Key elements are unrelated and ordered according to the variations that exist in all the original variables. In this collection of key items ordered, the first few items contain many variations that were present in the actual database [26]. This behavior is indicated by the decrease in Eigen's number of data variables [27] [28].

## **2.7 Machine learning algorithms**

Different monitored, supervised and targeted algorithms can be used to solve real-time recognition problem. A different algorithm has proven useful for different systems, which is why there is no clear difference as to which algorithm is more suitable than the others. The use of power and memory and the difficulty of the problem vary and form a set of conditions that determine which algorithm is most suitable for a particular problem. [23]

### **2.7.1 Random forests**

The Random Forest is a collective learning mechanism for classification, retreat and other activities [33].

It works by building many decision-making trees during training and providing the classroom which is a descriptive prediction of decision-making trees. Random forest inhibits data equilibrium [32]. The combination of bagging method and randomization improves algorithm performance.

It seems that the more data there is with the label the more algorithm works. In [34], the unplanned forest surpassed the SVM and Naïve Bayes in separating body movements in cars, trains and trails. [29]

### **2.7.2 K-Nearest Neighbor**

KNN is an example-based separator. It applies to the principal that the classification of unknown situations can be done by decorating unknown events in a known pattern on the basis of a particular task [35,36]. This function is the same as the distance performance. Foerster et al. [37] were the first to use the k-NN algorithm to separate 9 tasks.

### **2.7.3 Support Vector Machine (SVM)**

Support Vector Machines are based on decision-making that defines decision limits. The decision plane separates the two items with different class memberships. Support Vector Machine aims to increase the boundary of resolution between hyperplanes. Krause et al. [38] found that frequent domain features provided better results than time domain features and the use of SVM to split 8 tasks. [29]

### **2.7.4 Artificial Neural Networks**

The neural implant network mimics the activity of neurons in the bloodstream. Finds the relationship between input signals and output signals. [39] The most common method of data training is the retrospective method. This way the error in exit is determined and then distributed back to the network.

## Chapter 3 System Development

### 3.1 Introduction

In this chapter, we will describe the step-by-step process used to design our system. The machine learning problem we face falls into the category of segregation problems. We used R Studio version 3.2.5 on windows 10 platform to do all the research.

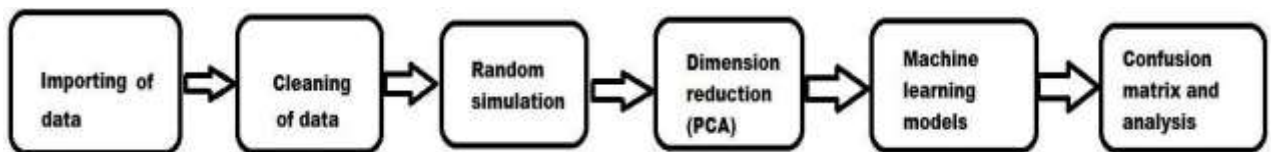


Figure 4: Flow of System Design of Human Activity Recognition

### 3.1 Importing and Cleaning Data

Data were collected from a survey conducted by a team of 30 volunteers. Each person had to perform six tasks (Walking, walking up, walking down, Laying, Standing and sitting). They used the embedded sensors of the Samsung Galaxy SII Smartphone to collect data. The database is uploaded to R studio as a comma-separated file. The imported database has 586 features.

The following refers to the actual features of the database. The database contains:

- \* A 561 vector with a variety of background and quantity.
- \* Label for its work.
- \* The subject index that made the test

Feature names have been changed and used as headers. Missing data and NA values are entered for the purpose of that particular feature.

Data was normalized using the following formula:

$$z_i = \frac{x_i - \bar{x}}{sd}$$

Where  $x_i$  is an instance of data,  $\bar{x}$  is mean of that feature column and  $sd$  is standard  $x$  deviation of that feature column

	tBodyAcc.mean...X	tBodyAcc.mean...Y	tBodyAcc.mean...Z	tBodyAcc.std...X	tBodyAcc.std...Y	tBodyAcc.std...Z	tBodyAcc.med...X	tBodyAcc.med...Y	tBodyAcc.med...Z	tBodyAcc.max...X	tBodyAcc.max...Y	tBodyAcc.max...Z
2	0.27841083	-0.016410568	-0.123520190	-0.998245280	-0.975300220	-0.990321990	-0.99880719	-0.974914370	-0.957688220	-0.9430675100		
7	0.27945388	-0.019640778	-0.110022150	-0.996921040	-0.967185930	-0.985117830	-0.99700268	-0.9680967100	-0.983116270	-0.9408866300		
10	0.28058589	-0.0099602983	-0.109095160	-0.994803440	-0.972758400	-0.986243870	-0.99540462	-0.9736632200	-0.985841930	-0.9400275100		
11	0.27688827	-0.0127218858	-0.103498320	-0.994815110	-0.973078920	-0.985357020	-0.99550827	-0.9738479600	-0.985172470	-0.9400275100		
12	0.27622817	-0.0214413828	-0.108202340	-0.998245958	-0.987213760	-0.992726590	-0.99825127	-0.9858965400	-0.983181880	-0.9439057800		
13	0.27845700	-0.0204147610	-0.112751720	-0.999134880	-0.984880040	-0.996274240	-0.99907654	-0.9829370200	-0.986410310	-0.9439057800		
15	0.28794572	0.0270929800	-0.061668123	-0.988640790	-0.816898800	-0.901906530	-0.98895795	-0.794284220	-0.888014600	-0.9259766800		
18	0.28813490	-0.0139169518	-0.106370480	-0.997694928	-0.987515670	-0.990407440	-0.99801432	-0.9879544800	-0.982198120	-0.9420275800		
22	0.27715238	-0.0179833288	-0.106801170	-0.997765220	-0.989957270	-0.996586670	-0.99829882	-0.9898887000	-0.986790430	-0.9414724200		
23	0.27587630	-0.0212642340	-0.110801220	-0.997862110	-0.990990760	-0.994592570	-0.99833345	-0.9894726800	-0.984484510	-0.9445667200		
24	0.27920020	-0.0177144278	-0.109161350	-0.998388280	-0.987307840	-0.990831590	-0.99888852	-0.9867713100	-0.988637390	-0.9436371800		
39	0.23715407	0.0078251224	-0.122857910	-0.979953588	-0.866193410	-0.968290470	-0.98017878	-0.8823158700	-0.968996710	-0.9396369300		
41	0.28159539	-0.0184348598	-0.111392870	-0.995468848	-0.984330220	-0.990995140	-0.99597751	-0.9823893700	-0.980283570	-0.9397887700		
42	0.27843225	-0.0198543828	-0.107970290	-0.994380280	-0.984582780	-0.991985780	-0.99505023	-0.9845219500	-0.986170550	-0.9397887700		
44	0.27985615	-0.0162606700	-0.112815850	-0.995018770	-0.978448310	-0.992661220	-0.99481764	-0.9851487800	-0.980246390	-0.9417154100		
48	0.27748121	-0.0174918188	-0.106359450	-0.996428438	-0.984673120	-0.990682660	-0.99673775	-0.9831586700	-0.988938010	-0.9385725700		
52	0.48347433	-0.0180744848	-0.118167390	-0.914811158	-0.885231120	-0.891748110	-0.91769888	-0.9246235400	-0.885894780	-0.7851038800		
59	0.28028640	-0.0183962800	-0.107488630	-0.996474938	-0.994988760	-0.991861310	-0.99729491	-0.9943383800	-0.993438800	-0.9394526400		
61	0.27672941	-0.0172095488	-0.103657970	-0.994788118	-0.991031320	-0.993187220	-0.99688888	-0.9966248800	-0.992794740	-0.9334585100		
64	0.27937115	-0.0176449538	-0.108181110	-0.995238838	-0.995810090	-0.994490450	-0.99551468	-0.9953381400	-0.993719970	-0.9382213100		
65	0.01981615	-0.0070373588	-0.028333368	-0.661299818	-0.713352570	-0.701155170	-0.68994813	-0.7069575100	-0.755421840	-0.8912835800		
66	0.34020850	-0.0364528848	-0.108258190	-0.958896488	-0.988855820	-0.984669480	-0.95685496	-0.8964938500	-0.987138760	-0.8912835800		
72	-0.27786634	-0.6840969800	0.346657720	-0.596410470	0.024882958	-0.180404490	-0.63181958	-0.0688567320	-0.235159390	-0.8662981700		
84	0.25548822	0.0212198638	-0.048949431	-0.224538698	0.022312942	-0.113196240	-0.25062407	-0.0219882870	-0.089186333	-0.0734780840		
87	0.31283404	-0.0283677498	-0.130951210	-0.383088888	-0.017382158	-0.127887880	-0.39527374	-0.0543888800	-0.085783859	-0.2903871800		
88	0.27891540	-0.0354862108	-0.080569176	-0.262041838	0.111277810	-0.212700270	-0.30868024	0.0898867980	-0.233259880	-0.0888828658		
90	0.38899599	-0.0137888888	-0.180161970	-0.239028858	0.072865343	-0.247204440	-0.28254588	0.0088078837	-0.261873890	-0.0912851788		

Figure 5: Snapshot of cleaned Dataset

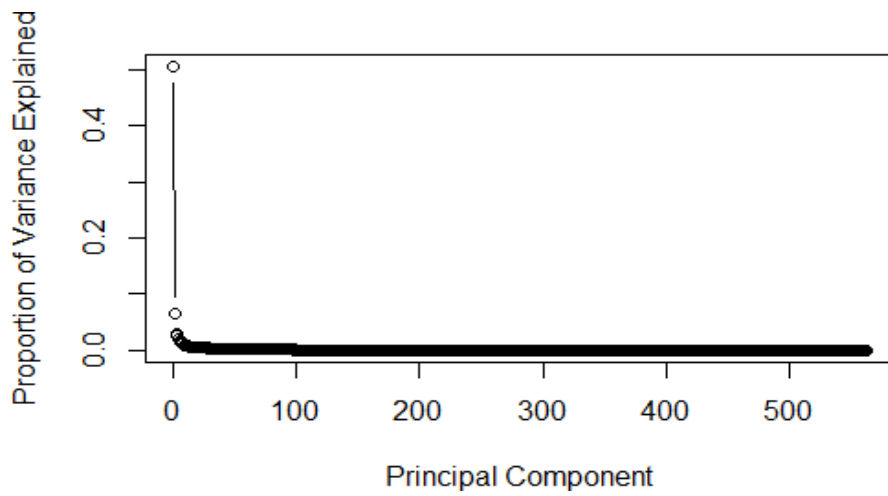
## 3.2 Random Simulation

Random simulation is used to check the accuracy of speculation models and to prevent overstatement and input of data. This approach involves random data classification in training and assessment set at scale 7: 3. All simulations are repeated 50 times to improve model accuracy according to Central Limit theorem statistics [10] [11]. The test database provides us with real-time data limitations and provides us with a way to test the stability of our model in real-life situations.

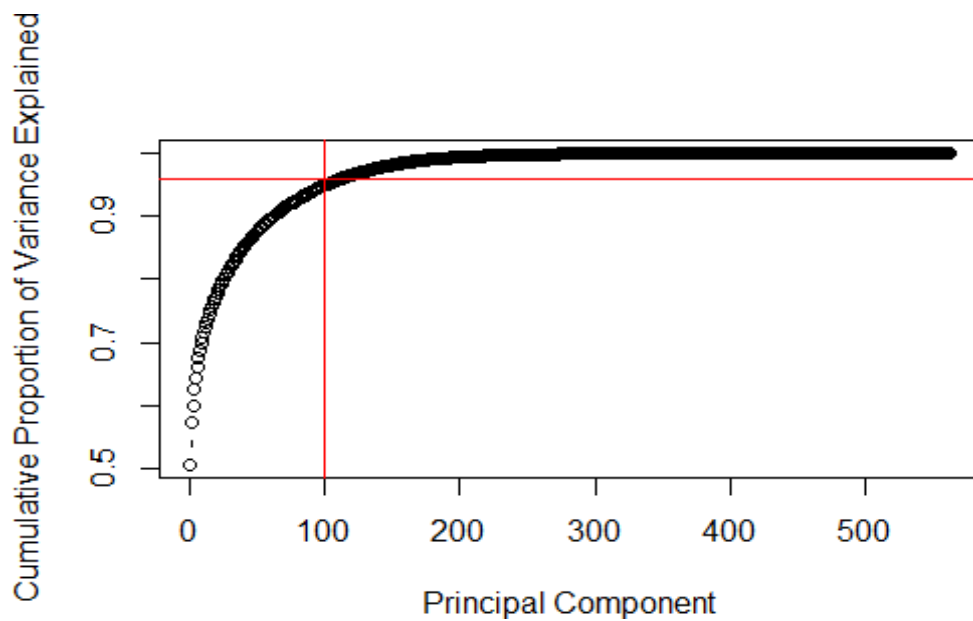
## 3.3 Dimensionality Reduction

We have used Key Analysis to reduce the number of features in our database. Key Asset Analysis (PCA) is designed to reduce the size of a large data set that contains a large number of consistent variables and to maintain as many variations as possible in the data set [41].

The core component analysis works with a variable converter to the data provided in a new set of variables, the main components (PCs). Key elements are unrelated and ordered according to the variations that exist in all the original variables. In this collection of main items ordered, the first few items contain many variations that were present in the actual database [41]. This behavior is indicated by the decrease in Eigen's rate of data variability [11] [12]. The following figure shows this change.



Graph 1: Proportions of Variances wrt Principal Component



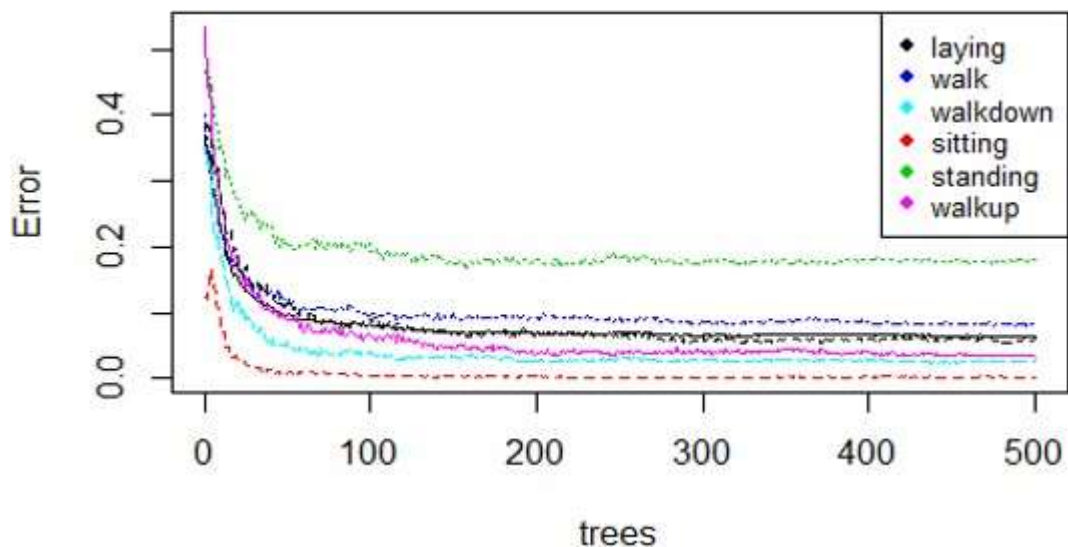
Graph 2: Cumulative Proportions of Variances

It is observed from the figure that first 100 Principal Components account for more than 95% of the variance in dataset, so we choose only first 100 components and disregard

other components. The changes were made in training dataset and exact transformation was applied to testing dataset.

### 3.4 Random Forest

The Random Forest is a collective learning mechanism for segregation, retreat and other activities [14]. It works by building many decision-making trees during training and providing the classroom which is a descriptive prediction of decision-making trees. The random forest inhibits the equilibrium of data [13]. In the R-system environment, 500 decision trees are built automatically. Through our efforts, we have seen that there is no need for the construction of 500 decision trees. Our model will work the same way and will produce the same results if we build only 80 trees as shown in the picture below.



Graph 3: Error rate measurements w.r.t to Number of Trees

Above figure shows that error rate is constant after construction of 80 trees, so we need only 80 decision trees.



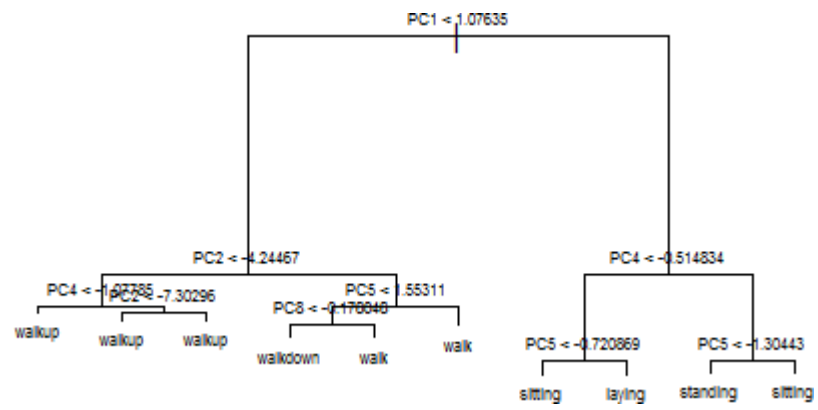
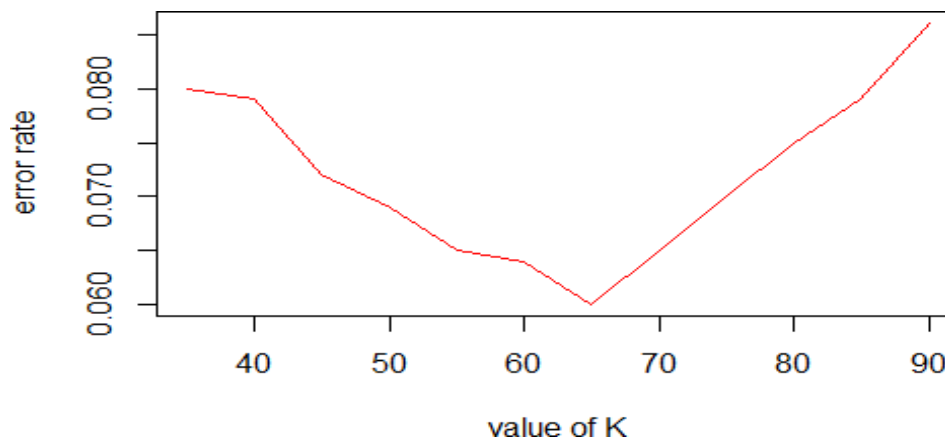


Figure 6: A sample tree of the forest constructed

### 3.5 K-Nearest Neighbor (KNN)

KNN is an example-based separator. It applies to the principal that the classification of unknown situations can be done by decorating unknown events in a known pattern on the basis of a particular task [15,16]. This function is the same as the distance performance.

We used Euclidean grade work to balance our learning activity. We determined the value of K by setting the error value graph vs the value of K as shown in the figure below.



Graph 4: Variation of error rate w.r.t value of K

From the figure we had found that error rate decreases from K=40 to K=65 and reaches minima at K=65.

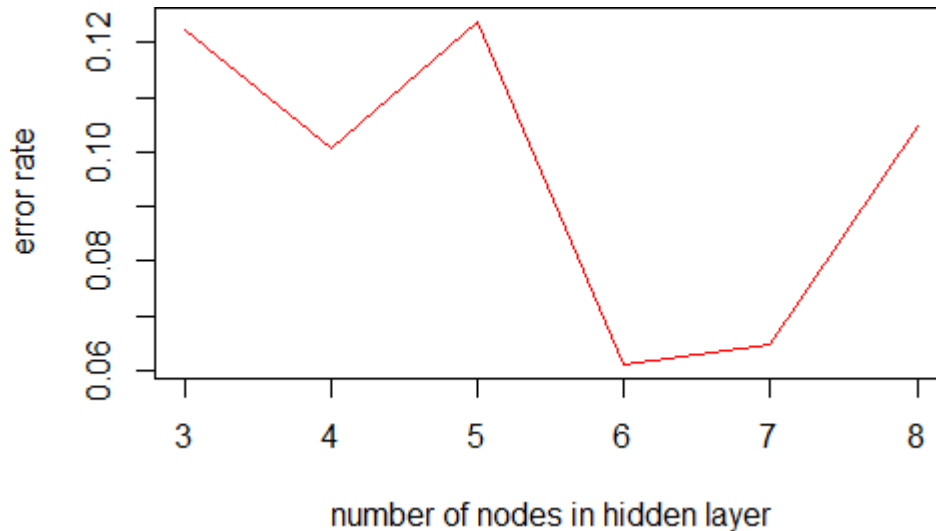
### **3.6 Support Vector Machine**

Support Vector Machines are based on decision-making that defines decision limits. The decision plane separates the two items with different class memberships. Support Vector Machine aims to increase the boundary of resolution between hyperplanes. We used the “e1071” SVM library in R to train our database [17]. The SVM algorithm takes kernel type, cost and gamma as parameters. We had selected the Gaussian (radical) kernel as a matching function.

The amount of cost is kept 1 as it makes the habit of staying longer and prevents data overload [18]. The gamma value determines the composition of the vector hyperplane. It is maintained as 0.013758, as the gamma value is equal to  $1 / (\text{the number of elements})$ .

### **3.7 Artificial Neural Network**

The Artificial Neural Network (ANN) models have a relationship between a collection of input signals and an output signal using a model based on our understanding of how the biological brain responds to input objects [40]. We have used the nnet package to train our database, which is used for single-layer feed-forward neural networks [20]. The nnet package trains the neural implant network using the retrieval method. This way the error in exit is determined and then distributed back to the network. To reduce the defect caused by each neuron, weights are updated [21]. The Linout parameter is set to False automatically as we face the problem of isolation [19].



Graph 5: Variation of error rate w.r.t number of nodes in hidden layer

From the above figure it is observed that error rate is minimum is at number of nodes in hidden layer equal to 6. Therefore, we have assigned value of size parameter as 6.

### 3.8 Prediction of Testing Dataset

After calculating the Training Key Features, we need to predict the test data using these Key Features. This seems very simple but, we need to understand a few points here:

1. Incorporating PCA into all data simultaneously in "reward" features of the training database in the test database and will affect the predictability of our model [12].
2. Incorporating PCA into separate datasets and training will result in vectors with different direction. Therefore, it will give negative results [12].

So, what are we doing now?

We will apply the change to tests set on the same side as the set training and the same facility with the measurement feature.

## Chapter 4 Performance Analysis

In this section we compared the performance of different machine learning models based on the matrix of confusion and the time taken to train the model. The time taken to train the model is calculated by taking 30 simulation methods.

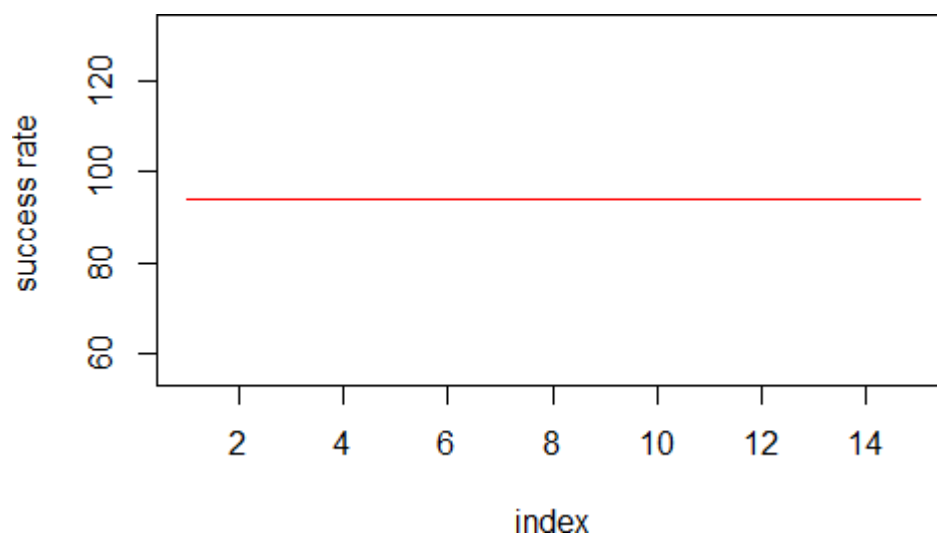
### 4.1 Random Forest

Table 1: Confusion matrix of data tested with Random Forest Model

Result/references	Laying	Sitting	Standing	Walk	Walk up	Walkdown
Laying	420	0	0	0	0	0
Sitting	14	326	52	0	0	0
Standing	0	35	373	0	0	0
Walk	0	0	0	368	3	2
Walk up	0	0	0	2	284	6
Walkdown	0	0	0	7	17	297

From the confusion matrix it is observed that percentage error rate is 6.25% and percentage success rate is 93.75%.

Time taken to train the model is 10.56 seconds.



Graph 6: Variation of success rate in Random Forest Model

From the above figure, it is observed that there is no variation in success rate of Random Forest Model.

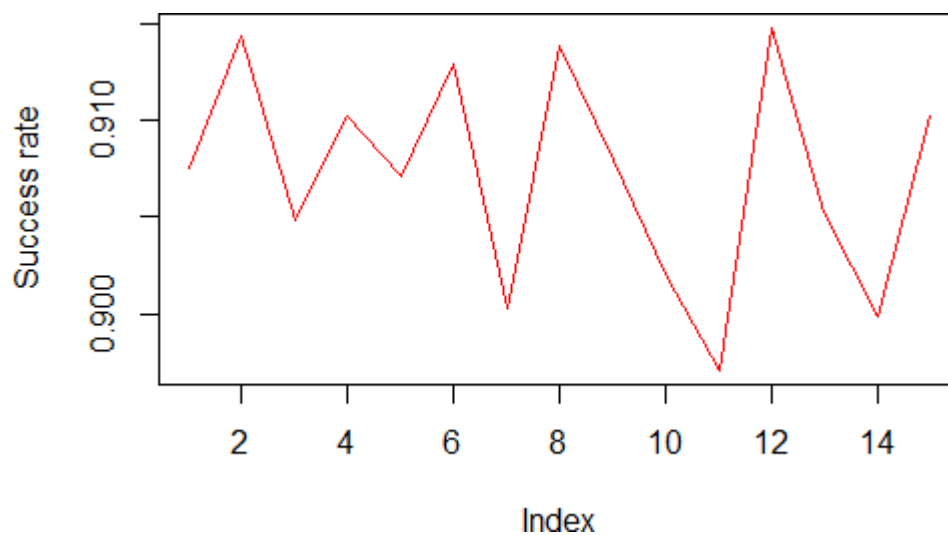
## 4.2 K-Nearest Neighbor

Table 2: Confusion matrix of data tested with K Nearest Neighbor Model

Result/references	Laying	Sitting	Standing	Walk	Walk up	Walkdown
Laying	421	4	0	0	0	0
Sitting	1	269	23	0	0	0
Standing	10	104	407	0	0	0
Walk	0	0	0	365	17	8
Walk up	0	0	0	7	259	5
Walkdown	2	0	0	1	16	287

From the confusion matrix it is observed that percentage error is 8.98 % and percentage success rate is 91.02%.

Time taken to train the model is 2.7 seconds.



Graph 7: Variation of success rate in K Nearest Neighbour

From the above figure, it is observed that success rate of K nearest neighbour varies from 91.5% to 89.7%.

### 4.3 Support Vector Machine

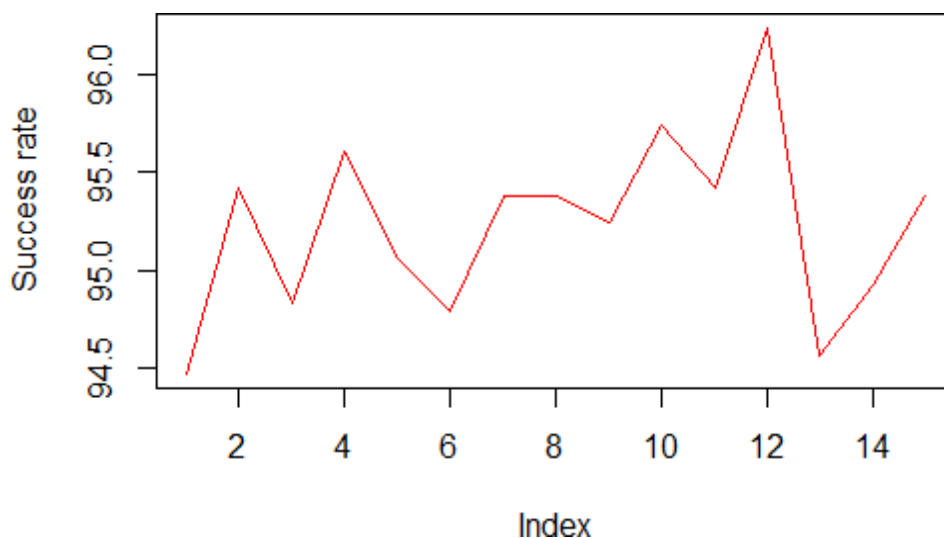
Table 3: Confusion matrix of data tested with Support Vector Machine Model

Result/references	Laying	Sitting	Standing	Walk	Walk up	Walkdown
Laying	429	0	0	0	5	0
Sitting	0	333	42	0	2	0
Standing	0	35	393	0	2	0
Walk	0	0	0	362	11	0
Walk up	0	0	0	0	292	0
Walkdown	0	0	0	0	5	295

From the confusion matrix it is observed that percentage error is 4.63 % and percentage success rate is 95.37%

Time taken to train the model is 12.9 seconds.





Graph 8 Variation of success rate in Support Vector Machine

From the above figure, it is observed that success rate of support vector machine varies from 94.5% to 96.4%.

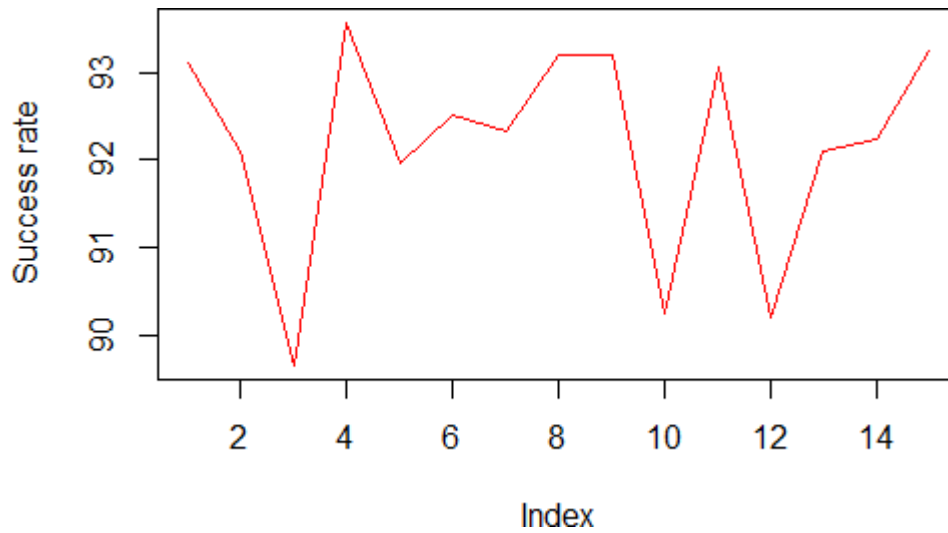
#### 4.4 Artificial Neural Network

Table 4: Confusion matrix of data tested with Artificial Neural Network

Result/references	Laying	Sitting	Standing	Walk	Walk up	Walkdown
Laying	432	0	2	0	0	0
Sitting	8	318	50	1	0	0
Standing	1	48	374	3	0	4
Walk	0	0	0	361	3	9
Walk up	0	1	0	4	279	8
Walkdown	0	0	0	2	5	293

From the confusion matrix it is observed that percentage error is 6.76 % and percentage success rate is 93.24%

Time taken to train the model is 5.2 seconds.



Graph 9: Variation of success rate in Artificial Neural Network

From the above figure, it is observed that success rate of artificial neural network varies from 93.5% to 89.6%.

## Chapter 5 Conclusion

We had read the Database Activity Activity database and studied the Random Forest Learning Algorithms, K-Nearest Neighborhood, Vector Support Machine and Artificial Neural Networks. We have reduced the size of our database from 586 features out of 100 features using Principal Component Analysis. We did a Data Analysis and found that the Support Vector Machine was highly effective (95.37%) in predicting Human Activity. SVM works very well because we used the Gaussian kernel for reduced data.

The minimum time to train the machine learning model was taken by K-Nearest Neighbor (2.7 seconds) because it is very small and uses Euclidian distance work. The maximum training time for the model was taken by the Support Vector Machine (12.9 seconds).

- Future Features: The model used can be used to predict People's Activity in real time. An app on Android Platform can be used to transfer ratings and implement a model on those ratings. This app has features in monitoring the health and performance of athletes etc.
- Innovative Idea: We found that PCA (Principal Component Analysis) data processing led to improved model design that led to improved performance.

## References

1. Adil Mehmood Khan, Young-Koo Lee, Sungyoung Y. Lee, and Tae-Seong Kim, A triaxial accelerometer-based physical -activity recognition via augmented-signal features and a hierarchical recognizer, *Information Technology in Biomedicine, IEEE transactions on information technology in biomedicine*, Volume 14, NO. 5, September 2010
2. Charles Arthur, "Smartphone explosion in 2014 will see ownership in India pass US.", accessed on 30/3/2017, <http://www.theguardian.com/technology/2014/jan/13/smartphone-explosion-2014-india-us-china-firefoxos-android>
3. "Human activity recognition using smartphone dataset", accessed on 1/11/2016, <https://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones>
4. Lester, Jonathan and Choudhury, Tanzeem and Borriello, Gaetano , "A practical approach to recognizing physical activities", accessed on 4/10/2016, [https://www.cs.cornell.edu/~tanzeem/pubs/JonathanLester\\_EDAS-1568973904.pdf](https://www.cs.cornell.edu/~tanzeem/pubs/JonathanLester_EDAS-1568973904.pdf)
5. T.B. Moeslund, A.Hilton, V.Kruger, "A survey of advances in vision-based human motion capture and analysis, *Computer Vision Image Understanding*", Volume 104, Issues 2–3 ,December 2006.
6. L. Bao and S. S. Intille, "Activity recognition from user-annotated acceleration data," *Pers Comput.*, *Lecture Notes in computer Science*, vol. 3001, pp. 1–17, 2004.
7. Kwapisz, Jennifer R and Weiss, Gary M and Moore, Samuel A, "Cell phone-based biometric identification", *Biometrics: Theory Applications and Systems (BTAS)*, 2010 Fourth IEEE International Conference, 10.1109/BTAS.2010.5634532, September 2010
8. Kwapisz, Jennifer R and Weiss, Gary M and Moore, Samuel A, "Activity recognition using cell phone accelerometers", accessed on 21/11/2016, <http://www.cis.fordham.edu/wisdm/includes/files/sensorKDD-2010.pdf>
9. Ramiro J.Caro, "Central Limit Theorem Simulator" , accessed on 15/11/2016, [https://rpubs.com/RamiroJC/CLT\\_Slides](https://rpubs.com/RamiroJC/CLT_Slides)
10. Central Limit Theorem ,accessed on 12/11/2016, [http://www.investopedia.com/terms/c/central\\_limit\\_theorem.asp](http://www.investopedia.com/terms/c/central_limit_theorem.asp)
11. " Principal component analysis" ,accessed on 10/11/2017, [https://en.wikipedia.org/wiki/Principal\\_component\\_analysis](https://en.wikipedia.org/wiki/Principal_component_analysis)
12. " Practical Guide to Principal Component Analysis (PCA) in R & Python", accessed on 5/11/2016,

<https://www.analyticsvidhya.com/blog/2016/03/practical-guide-principal-component-analysis-python/>

13. Leo Breiman , Adele Cutler, “Random forests”, accessed on 10/11/2016,

[https://www.stat.berkeley.edu/~breiman/RandomForests/cc\\_home.htm](https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm)

14. Tavish Srivastava, “Introduction to Random forest – Simplified”, accessed on 8/11/2016,

<https://www.analyticsvidhya.com/blog/2014/06/introduction-random-forest-simplified/>

15. “k-nearest neighbors algorithm”, accessed on 12/2/2017,  
[https://en.wikipedia.org/wiki/K-nearest\\_neighbors\\_algorithm](https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm)

16. Ola Soder, “knn classifiers”, accessed on 4/3/2017,

[http://www.fon.hum.uva.nl/praat/manual/kNN\\_classifiers\\_1\\_What\\_is\\_a\\_kNN\\_classifier\\_.html](http://www.fon.hum.uva.nl/praat/manual/kNN_classifiers_1_What_is_a_kNN_classifier_.html)

17. David Meyer , Evgenia Dimitriadou , Kurt Hornik , Andreas Weingessel , Friedrich Leisch , Chih-Chung Chang , Chih-Chen Lin ,”Department of Statistics ,Probability theory group,TU Wein”, accessed on 2/3/2017,

<https://cran.r-project.org/web/packages/e1071/index.html>

18. Yu-Wei, David Chiu,” Machine Learning with R Cookbook”,

19. Gary H.Y ,” Notes for nnet, tree”, accessed on 2/2//2017,

<http://statisticsr.blogspot.in/2008/10/notes-for-nnet.html>

20. Brian Ripley , William Venables,”Package nnet”, accessed on 20/1/2017,

<https://cran.r-project.org/web/packages/nnet/nnet.pdf>

21. Aarshay Jain,” Fundamentals of Deep Learning – Starting with Artificial Neural Network”, accessed on 15/1/2017,

<https://www.analyticsvidhya.com/blog/2016/03/introduction-deep-learning-fundamentals-neural-networks/>

22. Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra and Jorge L. Reyes-Ortiz, “A Public Domain Dataset for Human Activity Recognition Using Smartphones”, European Symposium on Artificial Neural Networks, 24-26 April 2013.

23. Jorge L. Reyes-Ortiz, Alessandro Ghio, Davide Anguita , Xavier Parra, Joan Cabestany, Andreu Catal, “Human Activity and Motion Disorder Recognition: Towards Smarter Interactive Cognitive Environments”, European Symposium on Artificial Neural Networks, 24-26 April 2013.

24. TONI\_K, “Accelerometer Basics”, accessed on 20/9/2016,  
<https://learn.sparkfun.com/tutorials/accelerometer-basics>
25. Toni\_K, “Gyroscope Basics” ,accessed on 20/9/2016,  
<https://learn.sparkfun.com/tutorials/gyroscope>
26. I.T. Jolliffe ,”Principal Component Analysis”, Second Edition, springer, April 2002
27. “ Principal component analysis” ,accessed on 10/11/2017,  
[https://en.wikipedia.org/wiki/Principal\\_component\\_analysis](https://en.wikipedia.org/wiki/Principal_component_analysis)
28. “Practical Guide to Principal Component Analysis (PCA) in R & Python”, accessed on 5/11/2016,  
<https://www.analyticsvidhya.com/blog/2016/03/practical-guide-principal-component-analysis-python/>
29. Physical Human Activity Recognition Using Wearable Sensors Ferhat Attal 1 , Samer Mohammed 1,\* , Mariam Dedabrishvili 1 , Faicel Chamroukhi 2 , Latifa Oukhellou 3 and Yacine Amirat 1 Received: 11 September 2015; Accepted: 8 December 2015; Published: 11 December 2015 Academic Editor: Vittorio M.N. Passaro
30. Cleland, I.; Kikhia, B.; Nugent, C.; Boytsov, A.; Hallberg, J.; Synnes, K.; McClean, S.; Finlay, D. Optimal Placement of Accelerometers for the Detection of Everyday Activities. *Sensors* 2013, 13, 9183–9200.
31. Chamroukhi, F.; Mohammed, S.; Trabelsi, D.; Oukhellou, L.; Amirat, Y. Joint segmentation of multivariate time series with hidden process regression for human activity recognition. *Neurocomputing* 2013, 120, 633–644.
32. Leo Breiman , Adele Cutler, “Random forests”, accessed on 10/11/2016,  
[https://www.stat.berkeley.edu/~breiman/RandomForests/cc\\_home.htm](https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm)
33. Tavish Srivastava, “Introduction to Random forest – Simplified”, accessed on 8/11/2016,  
<https://www.analyticsvidhya.com/blog/2014/06/introduction-random-forest-simplified/>
34. Breiman, L.; Friedman, J.; Stone, C.J.; Olshen, R.A. *Classification and Regression Trees*; CRC press: Boca Raton, FL, USA, 1984.
35. “k-nearest neighbors algorithm”,accessed on 12/2/2017,  
[https://en.wikipedia.org/wiki/K-nearest\\_neighbors\\_algorithm](https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm)
36. Ola Soder, “knn classifiers”,accessed on 4/3/2017,

37. Foerster, F.; Smeja, M.; Fahrenberg, J. Detection of posture and motion by accelerometry: A validation study in ambulatory monitoring. *Comput. Hum. Behav.* 1999, 15, 571–583.
38. Krause, A.; Ihmig, M.; Rankin, E.; Leong, D.; Gupta, S.; Siewiorek, D.; Smailagic, A.; Deisher, M.; Sengupta, U. Trading off prediction accuracy and power consumption for context-aware wearable computing. In *Proceedings of the 2005 Ninth IEEE International Symposium on Wearable Computers*, Osaka, Japan, 18–21 October 2005; pp. 20–26
39. Brett Lantz, “Machine learning with R”, Second edition, Packt publishing, July 2015
40. Brett Lantz, “Machine learning with R”, Second edition, Packt publishing, July 2015
41. I.T. Jolliffe, ”Principal Component Analysis”, Second Edition, springer, April 2002

## Appendix

### Code 1

```
load("C:/Users/user/Downloads/samsungData.rda")
names(samsungData)
table(is.na(samsungData)) ## no NAs
table(samsungData$subject)
nameVec <- make.names(names(samsungData),unique=TRUE)
names(samsungData) <- nameVec
set.seed(123)
smp_size <- floor(0.70 * nrow(samsungData))
train_ind <- sample(seq_len(nrow(samsungData)), size = smp_size)
train <- samsungData[train_ind, ]
test <- samsungData[-train_ind, ]
par(mfrow=c(1,2))
plot(train[,1],train[,2], col=factor(train$activity),pch=19, cex=0.5)
plot(train[,2],train[,3], col=factor(train$activity),pch=19, cex=0.5)
legend("bottomleft",legend=unique(factor(train$activity)),
col=unique(factor(train$activity)), pch=19, cex=0.8)
pc <- prcomp(train[,-563], center=TRUE, scale=TRUE)
train.data<-data.frame(activity=train$activity,pc$x)
train.data<-train.data[,1:100]
## Construction of Tree
library(tree)
train.tree <- tree(factor(activity)~.,data=train.data)
```



```
par(mfrow=c(1,1))
```

```
plot(train.tree)
```

```
#Random Forest Constrcution
```

```
library(randomForest)
```

```
train.rf <- randomForest(factor(activity) ~., data=train.data, prox=TRUE, ntree=100)
```

```
train.rf$confusion
```

```
text(train.tree, cex=0.5)
```

```
#Transformation of testing data according to Principal Component values
```

```
test.data<-predict(pc,newdata=test)
```

```
test.data<-as.data.frame(test.data)
```

```
test.data<-test.data[,1:100]
```

```
result <-predict(train.rf,test.data)
```

```
test.data$activity=test$activity
```

```
references<-test.data$activity
```

```
table(references,result)
```

## Code 2

```
load("C:/Users/user/Downloads/samsungData.rda")
g=c()
ti=c()
for(i in 1:15){
nameVec <- make.names(names(samsungData),unique=TRUE)
names(samsungData) <- nameVec

  set.seed(123+10*i+100*i)
smp_size <- floor(0.70 * nrow(samsungData))
train_ind <- sample(seq_len(nrow(samsungData)), size = smp_size)
train <- samsungData[train_ind, ]
test <- samsungData[-train_ind, ]

pc <- prcomp(train[,-563], center=TRUE, scale=TRUE)

train.data<-data.frame(activity=train$activity,pc$x)
train.data<-train.data[,1:80]
library("nnet")
a<-Sys.time()
nmodel<-nnet(activity~.,data=train.data,size=7)
b<-Sys.time()
ti<-c(ti,b-a)
test.data<-predict(pc,newdata=test)
test.data<-as.data.frame(test.data)
```

```

test.data<-test.data[,1:80]
result<-predict(nmodel,test.data,type="class")

test.data$activity=test$activity
references<-test.data$activity

t<-table(references,result)
g<-c(g,(t[1,1]+t[2,2]+t[3,3]+t[4,4]+t[5,5]+t[6,6])/sum(t))
}
g<-g*100
plot(g,type = "l",col="red",y="success rate",main="variation of success rate")
gv<- var(g)
#mean time taken is 5.2 sec for training the model
#variance in success rate is 0.000149 approx

```

### Code 3

```

load("C:/Users/user/Downloads/samsungData.rda")
nameVec <- make.names(names(samsungData),unique=TRUE)
names(samsungData) <- nameVec
g=c()
ti=c()
for(i in 1:15){
  set.seed(123+10*i+100*i)
  smp_size <- floor(0.70 * nrow(samsungData))
  train_ind <- sample(seq_len(nrow(samsungData)), size = smp_size)
  train <- samsungData[train_ind, ]
  test <- samsungData[-train_ind, ]
  pc <- prcomp(train[,-563], center=TRUE, scale=TRUE)
  train.data<-data.frame(activity=train$activity,pc$x)

```

```

train.data<-train.data[,1:80]
library("e1071")
a<-Sys.time()
svm_model <- svm(activity ~ ., data=train.data)
b<-Sys.time()
ti<-c(ti,b-a)
test.data<-predict(pc,newdata=test)
test.data<-as.data.frame(test.data)
test.data<-test.data[,1:80]
result<-predict(svm_model,test.data,type="class")
test.data$activity=test$activity
references<-test.data$activity
t<-table(references,result)
aa<-t[1,1]+t[2,2]+t[3,3]+t[4,4]+t[5,5]+t[6,6]
g<-c(g,aa/sum(t)*100)
}
plot(g,type = "l",col="red")
vg<-var(g)
#mean time taken is 12.9 sec for training the model
#variance in success rate is 0.21 approx

```

#### **Code 4**

```

load("C:/Users/user/Downloads/samsungData.rda")
nameVec <- make.names(names(samsungData),unique=TRUE)
names(samsungData) <- nameVec
x=c()
y=c()
ti=c()
for(i in 1:15){

```

```

smp_size <- floor(0.70 * nrow(samsungData))
train_ind <- sample(seq_len(nrow(samsungData)), size = smp_size)
train <- samsungData[train_ind, ]
test <- samsungData[-train_ind, ]
pc <- prcomp(train[,-563], center=TRUE, scale=TRUE)
train.data<-data.frame(activity=train$activity,pc$x)
train.data<-train.data[,1:80]
test.data<-predict(pc,newdata=test)
test.data<-as.data.frame(test.data)
test.data<-test.data[,1:80]

library("class")
a<-Sys.time()

kmodel <- knn(train = train.data[,-1], test = test.data[,-80],cl = train.data$activity,
k=20+5*i)
b<-Sys.time()

t<-table(kmodel,test$activity)

aa<-t[1,1]+t[2,2]+t[3,3]+t[4,4]+t[5,5]+t[6,6]
x<-c(x,30+5*i)
y<-c(y,1-aa/sum(t))
ti<-c(ti,b-a)
}
plot(g,type="l",col="red",ylab="success rate",main="variation of success rate")
gv<-var(g)
#mean time taken is 2.737 sec for training the model
#variance in success rate is 0.0000318 approx

```

**JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT**  
**PLAGIARISM VERIFICATION REPORT**

Date: 17-06-2021 ✓  
 Type of Document (Tick):  PhD Thesis  M.Tech Dissertation/ Report  B.Tech Project Report  Paper  
 Name: Anchal Sharma <sup>Chirag Arora</sup> Department: Cse Enrolment No 171291 171273  
 Contact No. 9816243355 E-mail. 171291@juitsolan.in 171273@juitsolan.in  
 Name of the Supervisor: Prof.Arvind kumar  
 Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters): Human activity recognition

**UNDERTAKING**

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/ revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

**Complete Thesis/Report Pages Detail:**

- Total No. of Pages = 45
- Total No. of Preliminary pages = 5
- Total No. of pages accommodate bibliography/references = 3

*Chirag*

*Arvind*

(Signature of Student)

**FOR DEPARTMENT USE**

We have checked the thesis/report as per norms and found **Similarity Index** at .....<sup>28</sup>.....(%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

(Signature of Guide/Supervisor)

Signature of HOD

**FOR LRC USE**

The above document was scanned for plagiarism check. The outcome of the same is reported below:

Copy Received on	Excluded	Similarity Index (%)	Generated Plagiarism Report Details (Title, Abstract & Chapters)	
Report Generated on	<ul style="list-style-type: none"> <li>• All Preliminary Pages</li> <li>• Bibliography/Images/Quotes</li> <li>• 14 Words String</li> </ul>		Word Counts	
			Character Counts	
		Submission ID	Total Pages Scanned	
			File Size	

Checked by  
Name & Signature

Librarian

Please send your complete thesis/report in (PDF) with Title Page, Abstract and Chapters in (Word File) through the supervisor at [plagcheck.juit@gmail.com](mailto:plagcheck.juit@gmail.com)

