Classification Models for Activity Recognition using Smart Phone Accelerometers (One Year Master Thesis in Statistics)

Biswas Kumar (biku0017)
Supervisor : Priyantha Wijayatunga
Umeå University

Spring 2022

Acknowledgments

I would like to express my sincere gratitude to my thesis supervisor Dr. Priyantha Wijayatunga for providing me with the opportunity to contribute towards the research in classification models for human physical activity recognition, with this thesis. His patience, guidance, enthusiasm, and suggestions helped me in all the time of research and writing this thesis.

Besides my supervisor, I would also like to thank course coordinators, Assoc. Prof. Lina Schelin and Dr. Maria Karlsson, for their guidance and the motivation they provided all along and especially during our personal meetings, to meet the goals of the thesis. I cannot imagine better mentorship for my thesis.

I am also thankful to my all classmates for providing me constructive feedback during our feedback meetings and seminars.

My sincere thanks go to all of my friends for their constant encouragement and love.

Finally, I would like to thank my parents, my wife, my brother, and the rest of my family for their unconditional love and support, without which I could not have enrolled in the course and started my thesis work.

Abstract

The huge amount of data generated by accelerometers in smartphones creates new opportunities for useful data mining applications. Machine Learning algorithms can be effectively used for tasks such as the classification and clustering of physical activity patterns. This paper builds and evaluates a system that uses real-world smartphone-based tri-axial accelerometers labeled data to perform activity recognition tasks. Over a million data recorded at the frequency 20Hz, was filtered and pre-processed to extract relevant features for the classification task. The features were selected to obtain higher classification accuracy. These supervised classification models, namely, random forest, support vector machines, decision tree, naïve Bayes classifier, and multinomial logistic regression are evaluated and finally compared with a few unsupervised classification models such as k-means and self-organizing map (SOM) technique built on an unlabelled dataset. Statistical model evaluation metrics such as accuracy-precision-recall are used to compare the classification performances of the models. It was interesting to see that all supervised learning methods achieved very high accuracy(over 95%) on labeled datasets as against 65% by unsupervised SOM. Moreover, they registered very low similarity (23%) among themselves on unlabelled datasets with the same selected features.

Abstrakt

Title: Klassificeringsmodeller för aktivitetsigenkänning använder sig av Accelerometrar för smarta telefoner

Den enorma mängd data som genereras av accelerometrar i smartphones skapar nya möjligheter. för användbara datautvinningstillämpningar. Machine Learning-algoritmer kan effektivt användas för uppgifter som klassificering och klustring av fysiska aktivitetsmönster. Detta papper bygger och utvärderar ett system som använder verkliga smartphone-baserade tri-axiala accelerometrar märkta data till utföra aktivitetsigenkänningsuppgifter. Över en miljon data inspelad vid frekvensen 20Hz, fil- bearbetad och förbehandlad för att extrahera relevanta funktioner för klassificeringsuppgiften. Funktionerna var väljs för att erhålla högre klassificeringsnoggrannhet. Dessa övervakade klassificeringsmodeller, nämligen slumpmässig skog, stödvektormaskiner, beslutsträd, na ıve Bayes-klassificerare och multinomial logistisk regression utvärderas och jämförs slutligen med några oövervakade klassificeringsmod- els såsom k-means och self-organizing map (SOM) teknik byggd på en omärkt dataset. Statistisk modellutvärderingsstatistik som precision-precision-recall används för att jämföra klassificeringsprestanda för modellerna. Det var intressant att se att alla övervakade lärandet metoder uppnådde mycket hög noggrannhet (över 95%) på märkta datamängder jämfört med 65% av oöverträffad vised SOM. Dessutom registrerade de mycket låg likhet (23%) sinsemellan på omärkta datauppsättningar med samma valda funktioner.

Popular scientific summary

Nowadays, smartphones are so common that almost every other person in the world owns one. These smartphones come with many sensors in them, that are capable of tracking, storing, and transmitting data related to the physical movement of the holder. One such sensor is called an accelerometer which measures the acceleration of a moving body, the holder in this case. These acceleration based data can be processed to find hidden movement patterns of the holder, if they are unknown. So-called machine learning-based algorithms are used to learn models that can recognize these patterns. Machine learning is a type of artificial intelligence method, that allows software applications to perform more accurate predictions of an outcome of interest for a given set of features of a system without being explicitly programmed to do so.

These accelerometer data-based machine learning studies are useful in various sectors such as healthcare, automobile and consumer electronics. One such common example of the healthcare sector is a fitness application wherein the mobile app can track users' movements using accelerometers and can generate health reports. They can be used for further personalized training advice to the user. Also, a patient's movement at a hospital can be tracked to apply the personalized treatment. Low levels of physical activity in older patients during hospitalization have been linked to loss of functional ability. The progress of such elderly patients can be tracked and used by doctors for their betterment.

In this paper, the sample of raw data from accelerometers was used for building machine learning models that can be used for physical activity recognition. These raw data mainly come from contests of six types of physical activities, namely, jogging, lying down, sitting, stairs, standing and walking. These activities have a different set of movement patterns i.e., different acceleration in different directions, and the data is fed to the model to learn and recognize the same. Once, the models have learned from data where activity patterns are known; they can be applied in similar contexts where activity patterns are unknown to predict them. The best-performing model is used to predict in cases of new data for their activity recognition. These are examples of supervised machine learning models.

There is another category of models which are trained on data where activity patterns are unknown. They are called unsupervised machine learning models. These models and methods only say which activity patterns are similar, thus group them and which activity patterns are dissimilar, thus separate apart.

This paper implemented both types of machine learning models (supervised and unsupervised) on the accelerometer dataset and compares in contexts of new data for the recognition of unknown activities. It was interesting to see how much supervised and unsupervised learning models agree to each other using the similarity table.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Background

According to the research from Strategic Analytics [1] in June 2021, half of the world's entire population now own smartphones. These smartphones are equipped with many diverse and powerful sensors. These sensors include Accelerometers, GPS, Magnetometer, Proximity sensors, Ambient light sensors and Temperature sensors. The sensors continuously collect real-time data and provide exciting opportunities in the field of data mining for human activity recognition, etc. that can have many types of health applications, elderly care, etc. The topic in this thesis deals with the evaluation of the system while classifying datasets obtained from accelerometer towards activity recognition.

1.2 Problem

The data mining and machine learning algorithms can be effectively used to process the data from accelerometer into meaningful output. These outputs can be constructively utilized in monitoring health-related issues etc.

Statistical Problem: Prediction of physical activity Table 1. from tri-axial data recorded by accelerometer and comparing different prediction models and methods

Table 1: Prediction (6 Classes)

1. Jogging	2. LyingDown	3. Sitting
4. Stairs	5. Standing	6. Walking

1.3 Goal

The goal of the project is to evaluate the supervised and unsupervised learning models and compare the results using various combinations of training and testing datasets . Furthermore, it will be interesting to see how and to what extent different unsupervised learning methods agree with different supervised prediction methods or models.

1.4 Motivation

A healthy lifestyle is on the priority list of the most people around the world. Healthy people are the foundation of healthy economies and societies [2] as they fall less sick and do not pressurize the health care system of the country. The health benefit of physical exercise is an open secret. When it comes to working out, most of us keep it old-school and simple. Around 70% of people worldwide say their primary form of fitness is walking. Running was the second-most popular activity [3]. It is therefore, human physical activity recognition algorithms have gained major attention in the healthcare industry and especially in fitness applications. Since smartphones are really common these days which comes with movement tracking sensors such as accelerometer and GPS, the data from the accelerometer gives a huge opportunity to plunge in data mining for the human physical activity recognition.

Many fitness platforms and projects have started using activity recognition techniques such as pose estimation and jogging to promote a healthy lifestyle. The importance of activity recognition algorithms gave me a strong reason to research on the underlying models and share my findings towards further improvements in the field.

Numerous models have been built for classification of the accelerometer dataset for activity recognition tasks, and most of them have used supervised machine learning algorithms to train the models. This project intends to build both supervised and unsupervised models and compare the performance on the same dataset and extracted features.

2 Literature Review

Development and analysis of classification models using accelerometers based on collected data is a very interesting topic that attracts many statisticians and researchers all over the world. The work has been done in different aspects of sensor-based data collection and various model generation.
Jennifer et al. [4] collected labelled accelerometer data supervised by WISDM [5] team members as they record daily activities performed by twenty-nine users such as walking, jogging, climbing stairs, sitting, and standing. They transformed the raw time series data into segments and to accomplish this. They divided the data into 10-second segments and then generated features that were based on the 200 readings contained within each 10-second segment. The generated features were mainly related to average acceleration, standard deviation, average resultant acceleration, time between peaks and binned distribution (10 equal-sized bins). The model was generated for these time-series data that provides knowledge about the habits of millions of users passively—just by having them carry cell phones in their pockets. The work on data was carried out by the supervised learning models and the accuracy level for different models varied for different activities. The Neural Network produced the best average result with 91.7% accuracy.

Maurer et al. [6] used “eWatch” as a multi-sensor platform worn on different body positions such as on the belt, shirt pocket, trouser pocket, backpack, and neck to recognize six different activities. They considered only time-domain features and avoided frequency domain features on the ground of costly computation requirements. It may be noted that a time-domain features are those that change with time, such as correlation coefficient, whereas a frequency-domain feature represents much of the signal lies within each given frequency band, such as maxima. The correlation-based feature selection method from the WEKA toolkit was used to find feature sets containing features that are highly correlated within the particular class but are uncorrelated with each other. During the user study the sensors were sampled with a frequency of 50Hz and later down-sampled to lower frequencies. They found that the recognition accuracy increases with higher sampling rates, and with the accelerometer features the accuracy then stabilizes between 15 to 20Hz, and was only improved marginally with higher sampling rates. To maintain some of the high frequency components information and to reduce the computational complexity significantly, no low pass filter was used for down-sampling the data. They performed decision trees, k-nearest neighbor, naïve Bayes, and Bayesian network classifiers with five-fold cross-validation and found decision Tree as the best performing model. They further wish to extend their activity recognition classifier to other activities and investigate how the activity classification can support the recognition of the user’s location.

Bayat et al. [7] examined tri-axial accelerometer data-set from two most common phone positions: smartphone in hand and smartphone in the pants pocket. They developed a model that was capable of recognizing six sets of physical activities such as slow walking, fast walking, running, stairs-up, stairs-down, and dancing (aerobic) under real-world conditions. Six different classifiers and combinations of the same were analyzed. For in-hand phone position, the combination consisting of multilayer perceptron, logitBoost and SVM classifiers yields the highest accuracy (91.15%) whereas the best combination is of multilayer perceptron, random forest and simple logistic with 90.34% accuracy were obtained for the in-pocket phone position. It was concluded that a combination of classifiers can provide a better result with the phone held in hand rather than in the pocket.

Weiss et al. [8] researched smartwatch and smartphone-based activity recognition, and found that smartwatches are capable of identifying specialized hand-based activities, such as eating activities, which cannot be effectively recognized using a smartphone. Two types of models (personal and impersonal) for eighteen different activities were investigated. These activities include general activities (hand and not hand-oriented) and eating activities (hand-oriented). The personal models utilize data from only the intended user while impersonal models were built using training data from everyone but the intended user. Overall average accuracy of models for smartphone-based data was 72.6% and for smartwatch was 91.9%. Five different models such as random forest, decision tree, IB3 instance-based, naïve Bayes and multi-layer perceptron were used for the purpose. The personal models outperform impersonal models by over 47% and 27% for smartphones and smartwatches respectively. It was concluded that commercial smartwatches can recognize a wide variety of activities with relatively good accuracy.

Lockhart et al. [9] further extended their study to implement a self-training capability into their Actitracker smartphone-based activity recognition system and they believe personal models can also benefit other activity recognition systems as well. The smartphone data for fifty-nine users located in their pockets were studied for six different activities. This time they induced three types of models: impersonal, personal, and hybrid. Impersonal models use training data from a panel of users that will not subsequently use the model (thus the training and test sets have no common users) whereas the personal models use training data from only the user for whom the model is intended (the training and test data come from the same user, but contain distinct examples). Hybrid models are a mixture of impersonal and personal models. The training set has data from both the test subject and other users, but the test set's examples are distinct. Different algorithms such as decision trees, random forest, instance-based learning, neural networks, rule induction, naïve Bayes, and logistic regression were performed and it was observed that personal models performed the best with over 97.4% accuracy for all models followed by hybrid models with 88.7%. Impersonal models performed the lowest with 70.9% accuracy. Random forest classifier worked best among all. They concluded that hybrid models can make effective use of the personal data in the data set, even though only a small fraction of the data (on average 1/59) is personal data. This means that the classification algorithms can effectively identify the movement patterns of a particular user from among a host of users.

Girija et al. [10] present a novel data analytic scheme for intelligent Human Activity Recognition using wireless body sensors and smartphone inertial sensor. They used opportunity body sensor activity recognition database raw database [11] and pre-processed by applying noise filters and then sampled with fixed-width sliding windows of 2.56 sec and 50% overlap to extract features. From each window, a vector of 17 features was obtained by calculating variables from the accelerometer signals in the time and frequency domain. They used naive Bayes, decision tree, random forest, random committee, lazy IBk classifiers and K-means clustering. The random forest classifier performed the best with 96.30% accuracy while K-means clustering performed worst at 60%.

The summary of different supervised and unsupervised models that has been used in various studies above is highlighted in appendix B. as Table 7. for quick references. There are multiple studies conducted with supervised learning models, whereas only a few implemented unsupervised techniques. This study aims to compare both types of learning models on the same dataset with extracted time and frequency domain features, and to check to what extent different models agree to each other.

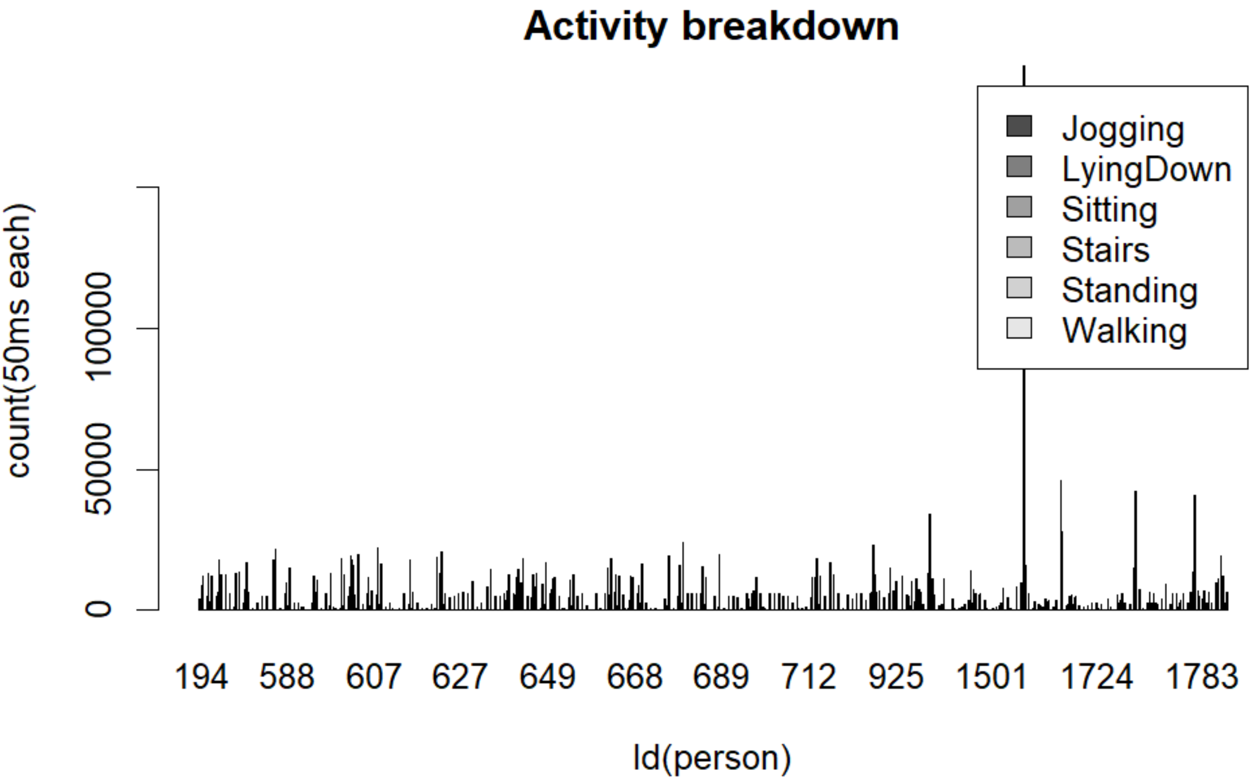
3 Data and Tools

The real-world time-series data was collected by the Wireless Sensor Data Mining (WISDM) Lab and is available for research. It was captured with their Actitracker system, which is available online at http://actitracker.com and is available free of cost in the Google Play store [12]. The raw data consists of labelled and un-labelled records of 225 users for 6(six) attributes and is available for research purposes. The dataset, detailed collection process and ethical considerations are enclosed as appendix A.

The accelerometer data was collected for every 50ms, so it had 20 observations per second that translates to 20Hz frequency. Since the data set is huge, this project used 500 randomly sampled labelled data from each label for testing and training purposes. This was to ensure that the model was trained to recognize each label irrespective of its imbalance in the original data set. Overall, 3,000 labeled sampled data set was used for further processing. Furthermore, the random sample (500 data) was extracted from the unlabeled dataset for prediction and comparison between supervised and unsupervised classifiers.

The contribution of few user towards each activity is represented by the activity distribution chart in Figure 1. The labeled raw dataset and unlabelled raw dataset is summarized in Table 2 and is represented by Figure 2.

Figure 1: Activity distribution chart



Each reading contained an x(horizontal movement), y(upward-downward movement), and z (forward-backward movement) value corresponding to the three axes/dimensions. The axes in

Table 2: original and sampled labelled and unlabelled data set

S.No.	Data-set	Type	Dimension	Sample	Uses
1.	Raw-data	Labelled	3,005,410 X 6	3,000	Train-Test dataset
2.	Raw-data	Un-labelled	38,209,771 X 5	500	Prediction and comparison

Figure 2: The original and sampled data set

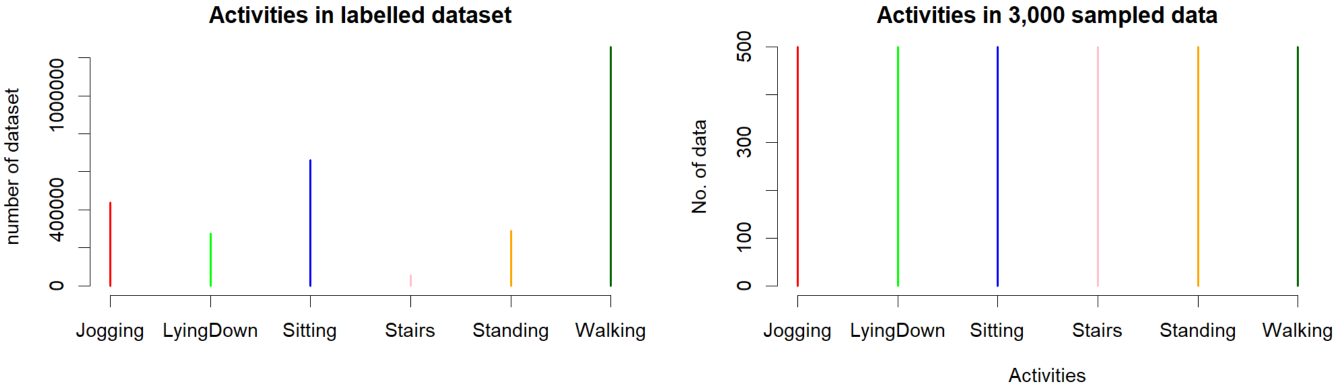


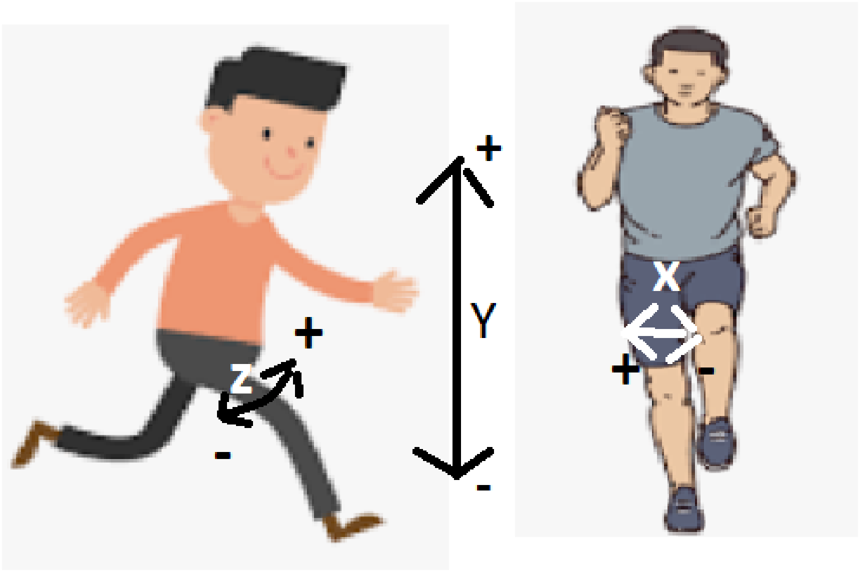
Table 3: Raw data attribute table

Attribute	Description
id	These represent the users on whom data was recorded for various activities.
action	It represents the activities that the user were performing.It is used to label. the dataset. There are altogether 6 (six) activities (Jogging, LyingDown, Sitting, Stairs, Standing, Walking) in the dataset.
timestamp	It represent the time at which the activity was recorded. The sampling rate is 20Hz (1 sample every 50ms).
x-acceleration (ax)	The acceleration in the x direction as measured by the android phone's accelerometer in m/s^2. 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.
y-acceleration (ay)	The acceleration in the y direction as measured by the android phone's accelerometer in $m/(s^2)$.
z-acceleration (az)	The acceleration in the z direction as measured by the android phone's accelerometer in $m/(s^2)$.

motion relative to user is represented by Figure 3.

This study has used R-studio and Weka [13] for data pre-processing, feature selection, manipulation and implementation of different clustering techniques and building supervised as well as unsupervised machine learning models for prediction.

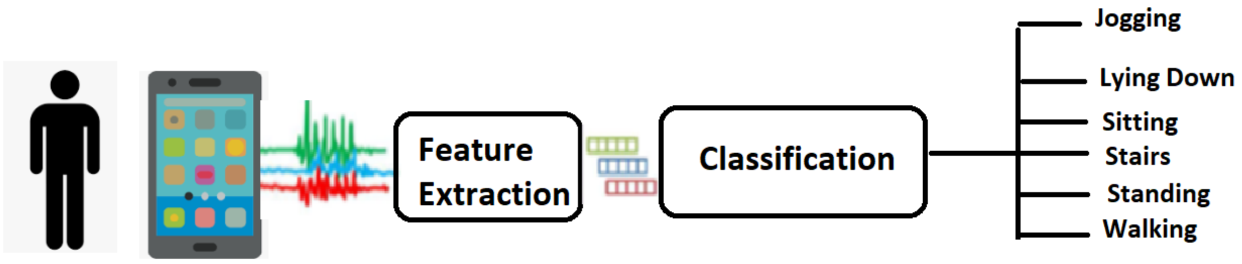
Figure 3: Variation of signal values of each activity



4 Data pre-processing and Feature description

The datasets were cleaned and updated with the respective headings. The sequence number (nseq) was added to the dataset and unwanted semicolon was removed from the z-acceleration (az) column. It was further checked for any NA values before being arranged as per increasing order of timestamp and id. It may be noted that 24,646 rows which had NA values, were omitted from the dataset. It accounted for about 0.82% of the dataset, so it is negligible. The block schematic diagram for processing of activity recognition is represented by Figure 4.

Figure 4: Block schematic for activity recognition processing



The variation of signal values in each of the x, y, and z dimensions is represented by the Figure 5. The raw data from the accelerometers was based on time-series and machine learning algorithms cannot be directly applied on such dataset [14]. It was, therefore, necessary to extract features from the raw data. The model used low pass Butterworth filter of fourth-order with a cut-off frequency at 3Hz, to extract low frequencies. In order to capture various features of the body motions for each activity and after a few trials; sliding windows of 2.0 sec and overlapping values of 0.05 sec were considered. It may be noted that the above process which include the sliding windows and overlapping values of data, reduced the effective number of sample to 1,380 from the initial input of 3000 samples.

The dataset was further processed and extracted features were added in the columns. These features and its description are described below:

4.1 Feature description

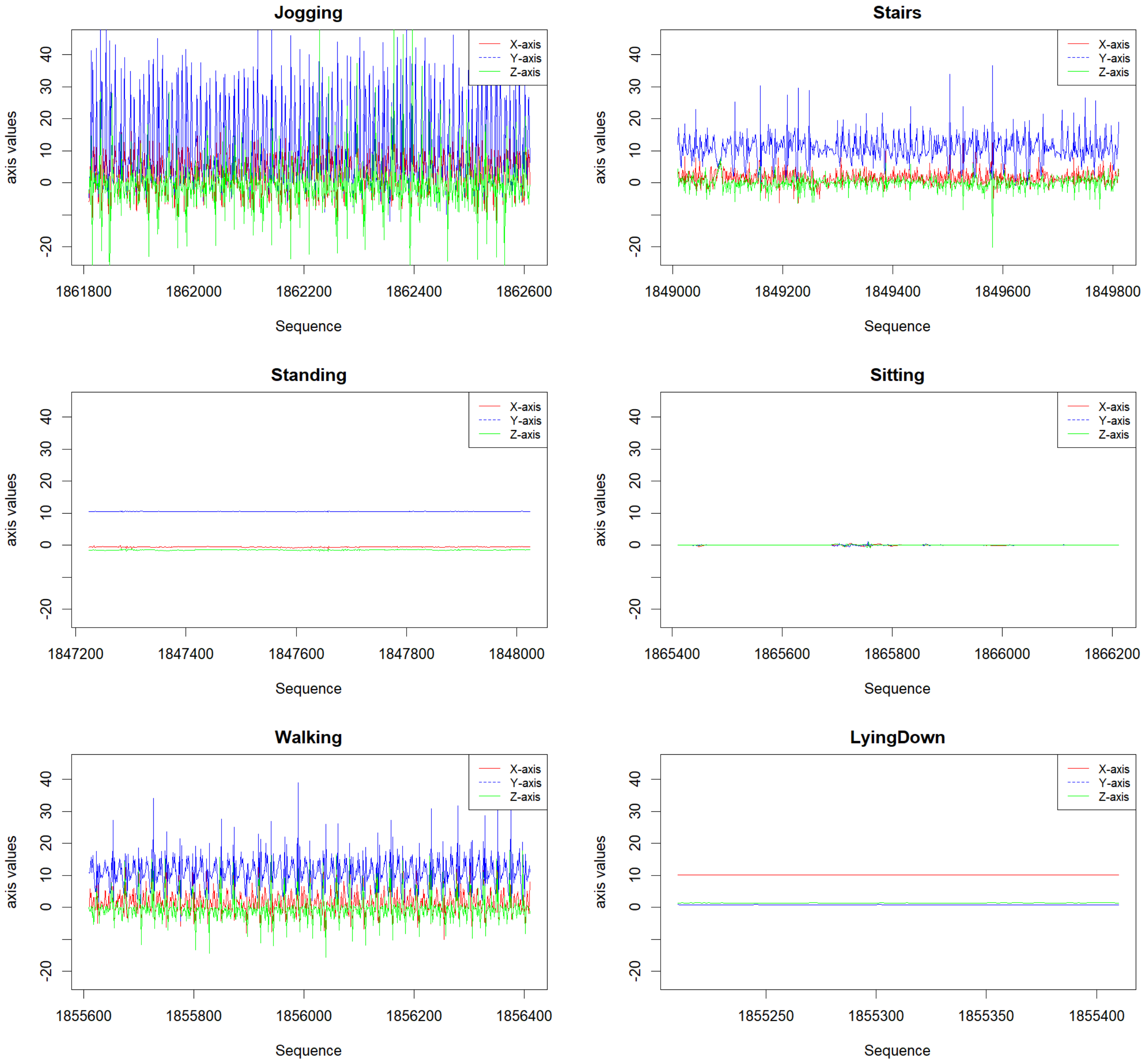
Resultant (amag): It represents the magnitude of the resultant vector of all three axes. It helps to understand the net acceleration of the moving body. It is given by below formula:

$$Resultant = \sqrt{(ax^2 + ay^2 + az^2)}$$

Low pass filter components of axes (lax, lay, laz): A low pass filter is a signal processing filter that passes signals below the cut-off frequency and attenuates signals with frequencies higher than the cut-off frequency. So, it essentially cut-off the noise component and provides a smoother form of signal. It helps to focus on the long term trend in the signal. The lax, lay and laz are the components that were obtained while passing low pass filter along acceleration in x, y and z-direction (ax, ay, az) respectively.

Low pass filter components for resultant (lamag): This represent the signal obtained while passing amag through low pass filter. It removes short term fluctuations and highlights

Figure 5: Variation of signal values for different activities (user id-194)



the long term trend of net acceleration.

High pass filter components of axes (hax, hay, haz): These were obtained by removing the low pass filter components from the acceleration in x, y and z-direction (ax, ay and az). It highlights the fluctuations in the signal towards the axes.

High pass filter components of resultant (hamag): It was obtained by removing lamag from the amag. It highlights the fluctuations in the signal towards the net acceleration.

Zero-cross components (zcs components): It is a point where the sign of a mathematical function changes (e.g., from positive to negative). This change could be an important event which can convey some information hidden in the signals. It is therefore, zcs components were captured for processed low pass and high pass signal components.

Peak-to-peak components (p2p components):This represents the difference of maximum and minimum for the high pass and low pass filter components.

Root mean square components (RMS components): This represents the square root of the arithmetic mean of the squares of the low pass and high pass filter component's values. Mathematically, it can be expressed as

$$RMS = \sqrt{mean(x^2)}$$

where, x is processed low pass and high pass filter components such as lax, lay, laz, lamag, hamag, hax, hay and haz.

RMS component for each of these fc was included in features. It gives a measure of the magnitude or size of the components. Unlike simple mean value, it avoids signals canceling out for positive and negative during estimation.

Kurtosis components (kur components): It is a measure of sharpness of the peak of a signal. It is a criteria for measurement of the shape of a distribution and could be computed by below formula[15]:

$$kurtosis = \frac{m_4}{m_2^2}$$

where, $m_4 = \frac{\sum(x-\bar{x})^4}{n}$, $m_2 = \frac{\sum(x-\bar{x})^2}{n}$.
x is processed low pass and high pass filter components such as lax, lay, laz, lamag, hamag, hax, hay and haz.
$\bar{x}$ is the mean of x and n is the sample size.

Skewness components (skw components): It is the distortion or asymmetry that deviates from the symmetrical bell curve, or normal distribution, in a set of data. This can be calculated by the below formula:

$$skewness = \frac{m_3}{m_2^{3/2}}$$

where, $m_3 = \frac{\sum(x-\bar{x})^3}{n}$, $m_2 = \frac{\sum(x-\bar{x})^2}{n}$,
x is processed low pass and high pass filter components such as lax, lay, laz, lamag, hamag, hax, hay and haz,
$\bar{x}$ is the mean of x and n is the sample size.
This is helpful to define the shape of a distribution along-with kurtosis components.

Crest factor component (cf components): It is the ratio of peak to RMS value of the components. It basically indicates how extreme the peaks are in a signal.

$$cf = \frac{|maximum(x)|}{\sqrt{mean(x^2)}}$$

where,
x is processed low pass and high pass filter components such as lax, lay, laz, lamag, hamag, hax, hay and haz.

Root mean square inverse difference velocity component (Vrms components): Herein, it represents to the application of RMS on the inverse of difference of the signal. The inverse of difference provides a discrete integration for a vector or a time series object [16]. For instance, if inverse of difference is applied to an integer sequence (1, 2, 3, 4, 5), it will produce a numeric output of (0, 1, 3, 6, 10, 15). The root mean square value will then be calculated for the output of inverse of difference, to give the final output.

The diffinv() function was implemented in R on the signals (i.e., processed low pass and high pass filter components) and the result was passed to the constructed RMS() function (as highlighted in RMS components above in this section), to get the intended positive feature values.

Entropy components (ent components): The shannon entropy, also known as information entropy [17], quantifies the expected value of the information contained in a vector. It is basically a measure of randomness of a system. The randomness is said to increase in a signal if the entropy increases. It can be expressed as below:

$$H(x) = -\sum_{i=1}^{n}[P(x_i) \times log_b P(x_i)]$$

$$H(x) = \sum_{i=1}^{n}[P(x_i) \times log_b(\frac{1}{P(x_i)})]$$

where,

$\sum_{i=1}^{n}$ is a summation operator for probabilities from i to n,
$P(x_i)$ is the probability of a single event.

The entropy() function of the entropy package in R was used to obtain the feature values. The signal was equally divided into 10 (ten) subsets (or bins) using discretize() function before passing to entropy(), for the calculation of ent components.

It was processed on low pass and high pass filter components such as lax, lay, laz, lamag, hamag, hax, hay and haz.

5 Algorithms and its implementation

5.1 Learning Algorithms (Supervised Machine Learning)

The supervised machine learning models were used for generating the prediction models. The labeled dataset is used to train the model, and then it predicts activities for the test data-set. The study used popular algorithms such as naïve Bayes, decision trees, support vector machines (SVM), multinomial logistic regression and random forest to check the performance of the models.

5.1.1 Naive Bayes

The naïve Bayes is essentially a probabilistic classifier based on the application of Bayes' theorem, which assumes mutual independence among the features, given the class. If this holds, the naïve Bayes can be the most accurate classifier in comparison. Bayes models often provide generalization performance that is slightly worse than that of linear classifiers [18].

The naïve Bayes classifier combines the Bayes probability model with a decision rule and the common rule is to choose the hypothesis that is most probable [19].

$$\hat{c} = ArgMax_c p(C = c) \prod_{i=1}^{n} p(F_i = f_i | C = c)$$

where p(C=c) is the probability distribution of the class variable, also known as prior probability distribution, and $p(F_i = f_i|C = c)$ is the conditional probability distribution of feature i given the class. Note that here, it has been assumed that $p(F_1 = f_1, \ldots, F_n = f_n C = c) = \prod_p (i = 1)^n p(F_1 = f_1 C = c)$ which means that when the class is known the features are mutually independent. Parameter (probability components in the model) can be estimated with maximum likelihood estimates.

$$p(C = c) = \frac{\#(C = c)}{n}$$

$$p(F_i = f_i C = c) = \frac{\#(C = c, F_i = f_i)}{\#(C = c)}$$

for i=1,. . . ,n, where (condition) is the number of data cases that satisfies the "condition". These estimates are called generative estimates.

5.1.2 Decision tree

A decision tree is like a tree structure that reflects a flowchart pattern wherein each node denotes a test on an attribute value; each branch represents an outcome of the test, and leaves represent classes. Essentially, they learn a hierarchy of if/else questions, leading to a decision [18]. The tree structure is represented in Figure 6. The classification is simple yet efficient and works with both numerical as well as categorical features.

A decision tree uses certain metrics such as entropy and information gain to make decisions. Entropy is a measure of uncertainty in the dataset which basically measures the impurity of a node. It can be represented as below [20]:

$$E(S) = -p_{(+)} log p_{(+)} - p_{(-)} log p(-)$$

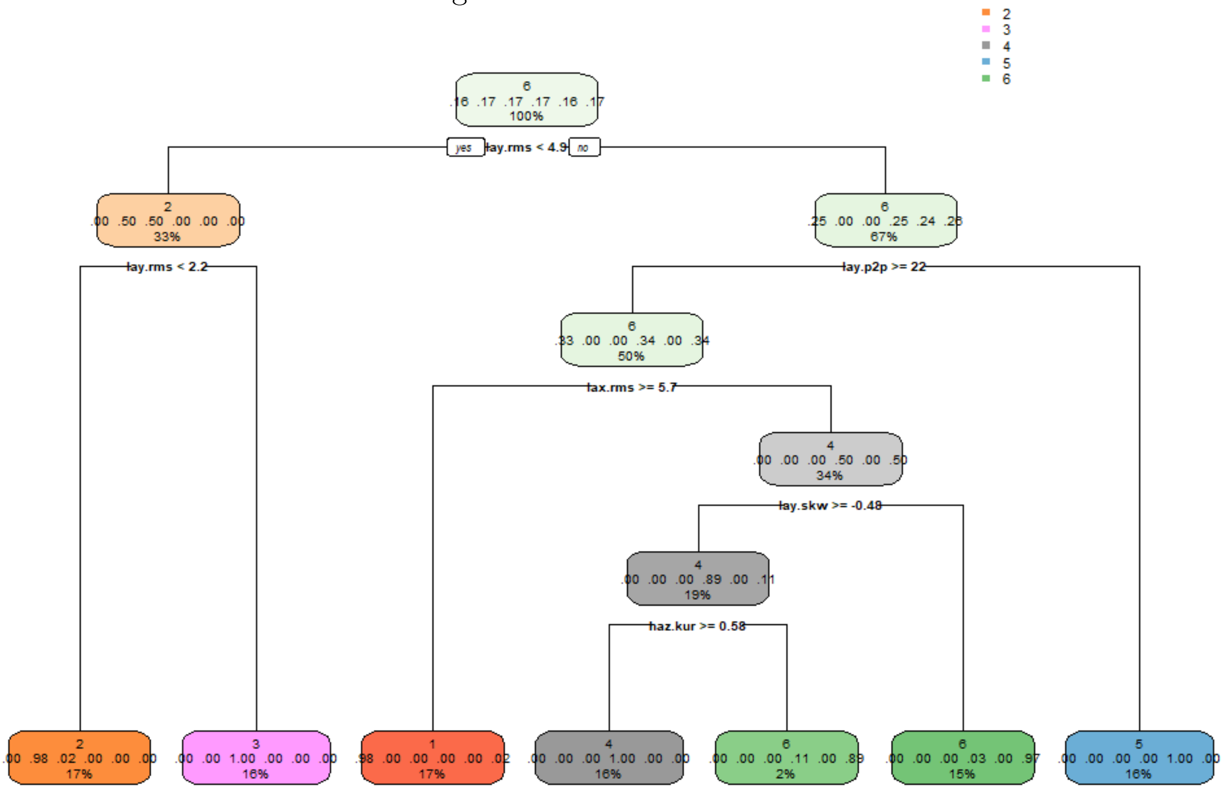
where,
$p_{(+)}$ is the probability of positive class,
$p_{(-)}$ is the probability of negative class, and
S is the subset of the training example.

The goal of the decision tree is to decrease the uncertainty in the dataset. Higher the value of entropy, higher is the uncertainty and content of information. Information gain measures how much "information" a feature gives us about the class and is, therefore, used to decide the ordering of attributes in the nodes of a decision tree. Information gain can be illustrated by the below formula:

Information Gain = (Entropy of the parent node) – (average entropy of the child nodes) [21]

The complexity increases with the increase in depth (or increasing nodes) of a decision tree. There are a few ways to control the depth of the tree using hyperparameter tuning, such as setting the maximum depth of the decision tree using the max_depth parameter.

Figure 6: Decision Tree



To avoid overfitting of the model, the pruning method is used to cut the nodes or sub-nodes that are not significant. This improves the performance of the decision tree. There are two ways for the pruning :
i. Pre-pruning : This method involves cutting or removing the nodes of a tree which has low importance. This method is applied while growing the tree.
ii. Post-pruning : As evident from its name, this method is applied once the tree is built to its depth, and then the cutting of nodes takes lace based on their significance [20].

This study has used rpart() function from the rpart package in R to obtain the model. The default settings has been retained to train and test the models.

5.1.3 Random Forest

Random forest is an ensemble method that fits several decision tree classifiers on various subsamples of the dataset and uses majority voting for classification.

It uses bagging method, which is a parallel processing technique in nature. Bagging, also known as Bootstrap Aggregation, is essentially an ensemble technique wherein it chooses random sample with replacement from the original dataset. These random samples, also known as bootstrap samples, is used to build up the models. Each model is trained independently to come up with their respective result. The results from various independent models are combined and the final output is generated based on majority voting. The process is known as aggregation. In short, the different training subset from the sample training data is created with replacement to construct models and the classification output is finalized based on majority voting [22].

Given a training feature set $X = x_1, ..., x_n$ with responses $Y = y_1, ..., y_n$, bagging repeatedly (B times) selects a random sample with replacement of the training set. Let $\hat{C}_b(x)$ be the class prediction of the b-th random-forest tree [23]. Then,

$$\hat{C}_{rf}^{B}(x) = \text{majority vote } \{\hat{C}_b(x)\}_1^B.$$

This study has used randomForest() function from the randomForest package in R to obtain the model. The model was iterated for total 10 (ten) number of trees (10, 100, 200, 300, 400, 500, 600, 700, 800, 900) to check different accuracy levels. These values were set using ntree parameter. The importance parameter was set as True to assess the importance of predictors. The number of tree was chosen based on the maximum accuracy among all and was set as ntree parameter in the final models for a different combinations of train and test datasets.

5.1.4 Support Vector Machines (SVM)

SVM classifies data by determining the optimal hyperplane that separates observations according to their class labels. This is to accommodate classes separable by linear and non-linear class boundaries. The objective of hyperplane is to maximize the separation of data points to their actual classes in an n-dimensional space. The closest points to hyperplane are called support vectors. There are three popular multiclass methods that can be applied to the multi-class problems. These methods can be summarized as below :

1. One-Vs-Rest: The one-vs-rest construct M number of SVM models where M is the number of classes. In this method, a hyperplane is constructed to separate the classes by taking all points into account and then divide them into two groups; one with a class points and other with all other class points [24].

For instance, for the t-th classifier, all the data points in class t $(x_i : t \in y_i)$ are considered as positive samples while negative samples represents the points in class t $(x_i : t \notin y_i)$.

$f_t(x)$: the decision value for the t -th classifier.
Prediction: $f(x) = argmax_t f_t(x)$
It may be noted that higher the value of f_t, the higher will be the probability that x is in the class t.

2. One-Vs-One: In one-vs-one technique, the multiclass problem is sub-divided into sub-problems; which are binary classification problems [24]. It computes $M * (M–1))/2$ number of classes. The fit is obtained among all binary sub-classifiers and finding the correct class by a majority voting mechanism along with the distance from the margin as its confidence criterion [25].

For instance, for the (s,t)-th classifier, all the points in class s $(x_i : s \in y_i)$ are considered as positive samples, whereas the points in class t $(x_i : t \in y_i)$ are considered as negative samples.
Let $f_{s,t}(x)$ be decision value of this classifier, so that:
$f_{t,s}(x) = – f_{s,t}(x)$
Prediction: $f(x) = argmax_s(\sum_t f_{s,t}(x))$
It may be noted that large value of $f_{s,t}(x)$ indicates that label s has a higher probability than the label t.

3. Directed Acyclic Graph (DAG) : The DAG is a graphical approach wherein the grouping of classes is based on some logic. It is constructed by introducing one-vs-one classifiers into the nodes of a decision Directed Acyclic Graph [24].

This study has considered one-vs-one technique for multi-class SVM classification and the e1071 package in R for the SVM classification. The svm() function was used to train and test models with its default parameters, including gamma (default: 1/(data dimension)) and cost (default: 1) values [26], where gamma is the argument for use by the kernel function, and cost helps to specify the cost of a violation to the margin.

5.1.5 Multinomial Logistic Regression

Multinomial logistic regression is a classification method that generalizes logistic regression to multiclass problems, i.e., with more than two possible discrete outcomes. It is used when the dependent variable in question is nominal (equivalently categorical, meaning that it falls into any one of a set of categories that cannot be ordered in any meaningful way) and for which there are more than two categories. The model assumes that data are case-specific; that is, each independent variable has a single value for each case. The logistic model can be generalized to categorical variables Y with more than two possible levels, namely $1, \ldots, J$ categories. Given the predictors $X_1, \ldots, X_p$, coefficients $\beta_0, \beta_1, ..., \beta_p$, multinomial logistic regression models the probability of each level j of Y by [27]:

$$p_j(x) = P[Y = j|X_1 = x_1, ..., X_p = x_p]$$

$$p_j(x) = \frac{e^{(\beta_{0j}+\beta_{1j}X_1+...+\beta_{pj}X_p)}}{1 + \sum_{l=1}^{J-1} e^{(\beta_{0l}+\beta_{1l}X_1+...+\beta_{pl}X_p)}}$$

for j= 1,...,J-1 and for the last level J :

$$p_J(x) = P[Y = J|X_1 = x_1, ..., X_p = x_p]$$

$$p_J(x) = \frac{1}{1 + \sum_{l=1}^{J-1} e^{(\beta_{0l}+\beta_{1l}X_1+\ldots+\beta_{pl}X_p)}}$$

It implies that $\sum_{j=1}^{J} p_j(x) = 1$ and there are $(J-1) \times (p+1)$ coefficients. It may also be noted that the last level J is treated as reference level.

This study has used nnet library in R to access multinom() function. The default settings were used in the function to train and test the models.

5.2 Clustering (Unsupervised machine learning)

Kamber and Pei [28] has defined cluster analysis or simply clustering as the process of portioning a set of data objects (or observations) into subsets. Each subset is a cluster, such that objects in a cluster are similar to one another, yet dissimilar to objects in other clusters. It is an 'unsupervised machine learning' technique with no defined labels. It essentially looks for patterns in the data and groups them accordingly. There are different algorithms used for cluster analysis. This study has used popular cluster technique such as K-means clustering and self-organizing maps to take an informed decision of the final clusters.

5.2.1 K-Means Clustering

Most partitioning methods are distance-based. K-means is a distance or centroid-based algorithm wherein a primary partition is created, and then it uses an iterative relocation technique that attempts to improve the partitions by moving objects from one group to another until "K" distinct clusters are obtained.

Given an initial set of k means m1(1),...,mk(1), the algorithm proceeds by alternating between two steps [29]:
Assignment step: In this step, each observation is assigned to the nearest mean using least squared Euclidean distance.

$$S_i^{(t)} = \left\{ x_p : \left\| x_p - m_i^{(t)} \right\|^2 \leq \left\| x_p - m_j^{(t)} \right\|^2 \; \forall j, 1 \leq j \leq k \right\}$$

where each x_p is assigned to exactly one $S^{(t))}$, even if it could be assigned to two or more of them.

Update step: Recalculate means (centroids) for observations assigned to each cluster.

$$m_i^{(t+1)} = \frac{1}{\left| S_i^{(t)} \right|} \sum_{x_j \in S_i^{(t)}} x_j$$

The algorithm is considered to be converged when there is no more changes in the assignments. There is no guarantee that it will reach to the optimum solution.

Three different K-means methods were performed in R using the fviz_nbclut package to find the optimum numbers of clusters.

5.2.1.1 Gap statistic

This is a standard method that compares the total within intra-cluster variation for different values of K with their expected values under the null reference distribution of the data. The name comes from the fact that the estimated number of optimal clusters will be a value that maximizes the gap statistic. This means that the clustering structure is far away from the random uniform distribution of points. The algorithm produced optimal number of the clusters as 6 (six) in Figure 7.

5.2.1.2 Silhouette (average silhouette width)

It is a measure of how similar an object is to its cluster (cohesion) compared to other clusters (separation). The score ranges from -1 to 1, wherein a high value shows that the object is closely matched while a low value reflects dissimilarity. The maximum value of the score will give the best value of K. The number of clusters produced by this method was 3 (three), as highlighted in Figure 7.

5.2.1.3 WSS (within-cluster sum of squares)

It reflects the sum of distances between the points and corresponding centroids for each cluster. This is more commonly known as the elbow method as an optimum value of K is chosen at the elbow point i.e., the point wherein the distortion score decline the most. Since the algorithm does not automatically highlight the final cluster value (K), therefore, there is a high ambiguity in picking up the value of K. The value of K seems to be 5 (five) as the elbow point in the Figure 7.

The optimal number of clusters traced below for all three methods.

Figure 7: Optimal number of cluster using Gap_statistic, Silhouttee and WSS methods

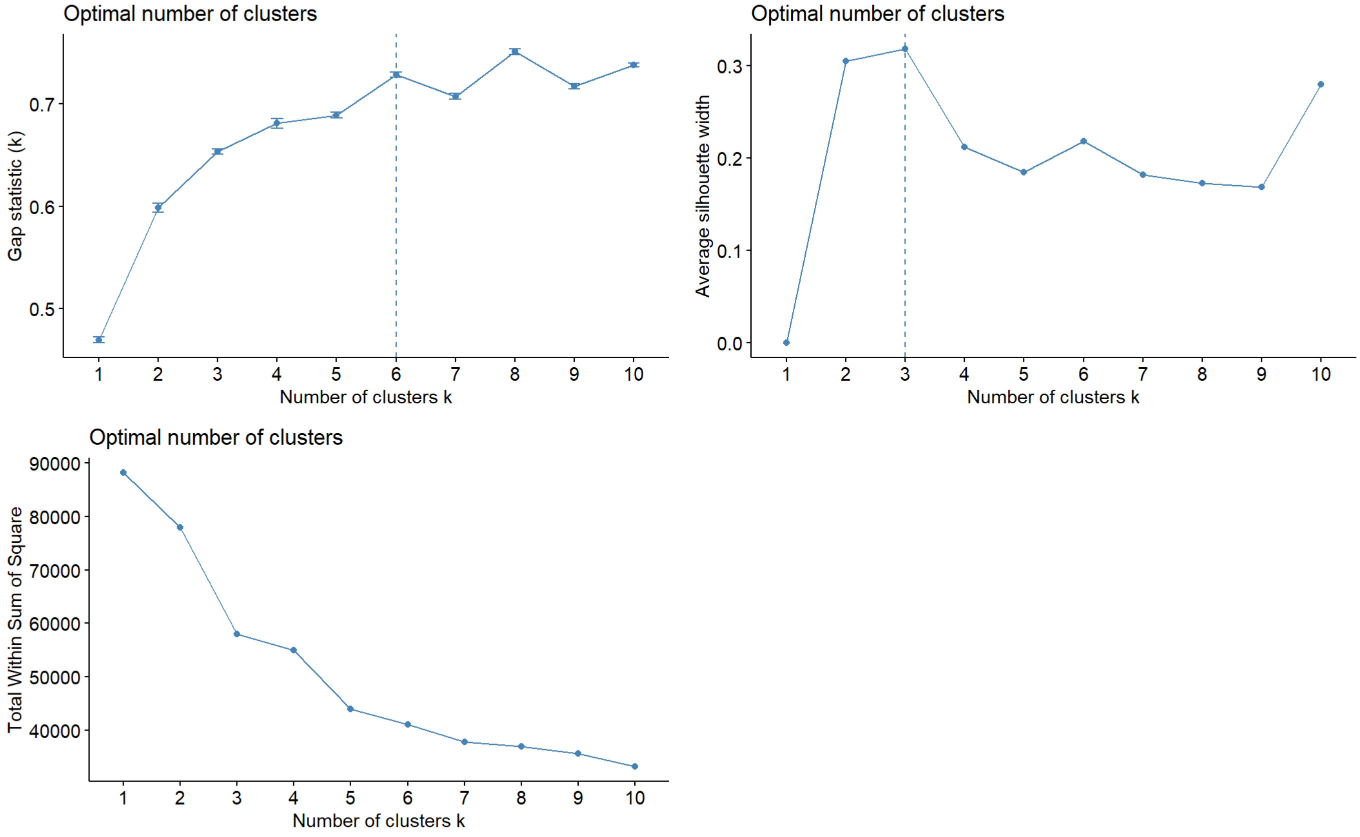
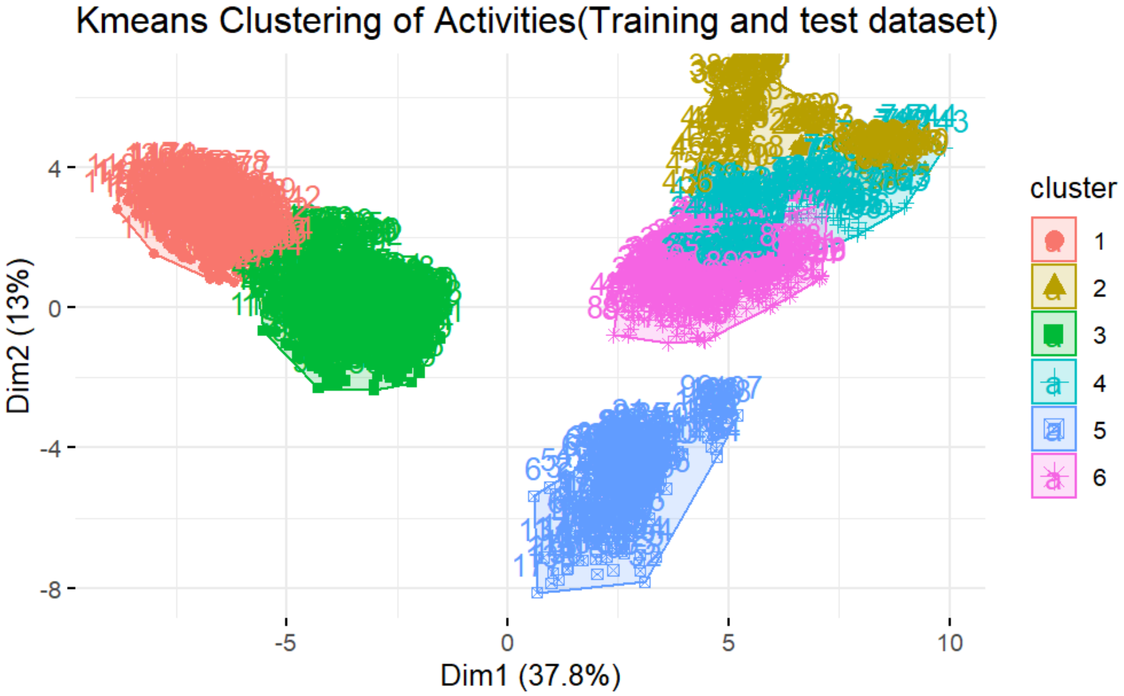


Table 4: Comparison of optimal K-means methods

Method	No. of Clusters
Gap_stat	6
Silhouette	3
WSS	5

The number of clusters for the classification was chosen as 6 (six) as highlighted by Gap_stat. The choice is also because of the fact that it objectively aligns with the known clusters (no. of activities) in the data.

Figure 8: K-means clustering, K=6



5.2.2 Self-organizing Maps

This is an artificial neural network based clustering technique that produces a low-dimensional (typically two dimensional) representation of a high-dimensional data set while preserving topographical structure of the data. Herein, the network is trained using competitive learning [30] wherein nodes compete for the right to respond to a subset of the input data. Hence, it is different than error-correction learning used in other neural networks.

It is also known as Self-organizing maps (SOMs) as it is an unsupervised clustering technique that cause different parts of the network to respond similarly to certain input patterns. It is considered similar to the cerebral cortex in the human brain which handle different inputs such as visual and auditory [31].

The weights of the neurons of this neural network is initialed to small random values. The other way for initialization is to sample evenly from the two largest principal component eigenvectors.

The learning is much faster in the latter because the initial weights are chosen from the space of the first principal components, already give good approximation of SOM weights. It has become popular due to the exact reproducibility of the results [32].

The euclidean distance to all weight vectors is computed as soon as training data is fed to the network. The neuron whose weight vector is most similar to the input is called the best matching unit (BMU). The weights of the BMU and neurons close to it in the SOM grid are adjusted towards the input vector. The magnitude of the change decreases with time and with the grid-distance from the BMU. The update formula for a neuron v with weight vector $W_v(s)$ is

$$W_v(s+1) = W_v(s) + \theta(u, v, s) \cdot \alpha(s) \cdot (D(t) - W_v(s))$$

where,
s is the step index
t is an index into the training sample
u is the index of the BMU for the input vector D(t)
$\alpha(s)$ is a monotonically decreasing learning coefficient
$\theta(u, v, s)$ is the neighborhood function which gives the distance between the neuron u and the neuron v in step s [33].

5.3 Performance metrics

It is important to understand the result and there should be some criteria that can compare and highlight the performance of models under evaluation. The results of various supervised and unsupervised models were evaluated on the metrics as explained ahead.

5.3.1 Accuracy

Accuracy is one of the most important metric for evaluating classification models. It reflects the fraction of predictions that the model got right, which comes from the below formula:

$$Accuracy = \frac{\text{number of correct predictions}}{\text{total number of predictions}}$$

For binary classification, accuracy can also be calculated in terms of positives and negatives as follows:

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

Where, TP = True Positives, TN = True Negatives, FP = False Positives, and FN = False Negatives.
Higher the accuracy, better is the performance of the classifier.

5.3.2 Kappa

Cohen's kappa (kappa) statistic is a measure for the performance of a model that handles both multi-class and imbalanced class problems very well [34]. It is a metric that is used not only to compare a classifier, but also to compare classifiers among themselves. Higher the kappa value, better is the performance of the classifier.

Cohen's kappa is calculated with the following formula [35]:

$$k = \frac{p_0 - p_e}{1 - p_e}$$

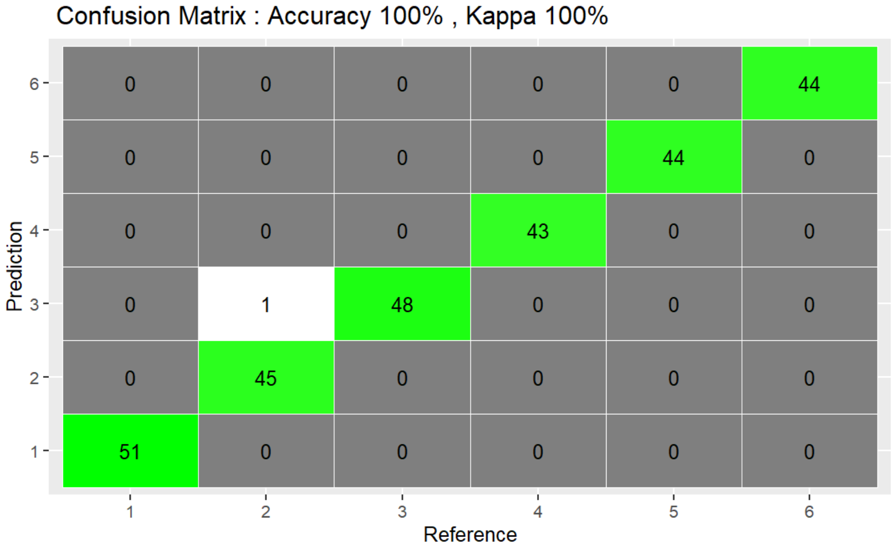
where, p_0 is the overall accuracy of the model, and p_e is the measure of the agreement between the model predictions and the actual class values as if happening by chance.

5.3.3 Confusion Matrix

Confusion matrix is a very popular measure used while solving classification problems. It can be applied to binary classification as well as for multiclass classification problems. It is good visualizing technique wherein we see the class-wise performance and mis-classification of dataset.

For instance, the Figure 9 represents performance metrics (Accuracy, Kappa and confusion Matrix) for the SVM classifier. It highlights that the model performed very accurate and predicted activities (class) with almost 100% accuracy. There was only one misclassification reported wherein the model predicted one activity as activity no.3 (three) while it was activity no.2 (two). Kappa at 100% also echoed that the model was excellent.

Figure 9: Performance metrics of SVM classifier (80% train, 20% test dataset)



Prediction \ Reference	1	2	3	4	5	6
6	0	0	0	0	0	44
5	0	0	0	0	44	0
4	0	0	0	43	0	0
3	0	1	48	0	0	0
2	0	45	0	0	0	0
1	51	0	0	0	0	0

The confusion matrix in Figure 9. indicates the distribution of prediction for 20% of test data (i.e., 276 data) from the overall dataset (1,380) of extracted features.

5.3.4 Precision

It refers to the proportion of the correct predictions among all the prediction of a certain class. It basically highlights the ratio of true positives among all the predicted positive outcomes. It can be expressed as [36]:

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

where, TP is true positive, and FP refers to false positive.

5.3.5 Recall

It is a proportion for a certain class that quantifies the number of correct positive predictions made out of all predictions for that class made by the classifier. It can be expressed as [36]:

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

where, TP is true positive, FN is false negative and FP refers to false positive.

5.3.6 F1 Score

F1 score is an important parameter that highlights the overall quality of the classifier's prediction. Higher the F1 score, better is the performance of a model. The maximum score that can be attained by the F1 is 1(one).

It can be expressed as [36]:

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

This study deals with multi-label classification and, therefore, there was a need of a single performance indicator that can help in analyzing the models with clarity. It was therefore, single performance indicator was achieved by averaging the precision and recall scores of individual classes [36]. The metrics such as precision, recall, and F1 score in Table 5. refers to the averaging of the precision, recall, and F1 scores of individual classes respectively.

6 Results and Discussion

The machine learning models were implemented on the sample of labeled dataset with its extracted features. All the supervised models achieved above 95% accuracy levels for different combinations of train and test dataset. On few occasions, it had reached to almost 100% accuracy with negligible misclassification error. The Table 5. depicts the performance of the supervised models and clustering methods on the various combination of training and testing dataset. It highlights that the random forest achieved over 96% score for the entire range of performance metrics for all train-test dataset combinations. It was evident that the random forest with train-test dataset combination of 80%-20% performed the best with an almost ideal 100% score achieved in each performance metric section. The decision tree could not match the rest of the supervised classifier. The model with dataset combination of 90%-10% achieved the highest score for the decision tree among all performance metrics. Naive Bayes performed very well, and its model with the dataset combination of 70%-30% seems to perform the best amongst them. It achieved the highest value in all performance metrics. On the other hand, SVM classifiers performed very well with each of them achieving over 99% score for all the metrics. SVM with the dataset combination of 80%-20%, seems to perform the best among three. The multinomial logistic regression achieved over 99.5% score for the entire metrics and for all combinations of the dataset. The model with 90%-10% dataset combination performed the best among them and seems to be at par with the random forest model of 80%-20% combination of the dataset.

Table 5: Performance comparison of supervised models and clustering methods for various test-train combinations of the dataset.

Model	Dataset (train- test)	Accuracy (%)	Kappa(%)	Precision (%)	Recall (%)	F1 (%)
Random Forest	90% -10%	98.55	98.25	98.66	98.40	98.50
	80% -20%	100.00	100.00	100.00	100.00	100.00
	70% -30%	96.76	99.71	99.75	99.77	99.76
Decision Tree	90% -10%	97.10	96.51	96.81	97.06	96.90
	80% -20%	95.65	94.78	95.50	95.64	95.54
	70% -30%	96.14	95.37	96.13	96.31	96.17
Naive Bayes	90% -10%	98.55	98.25	98.61	98.61	98.55
	80% -20%	98.91	98.69	98.98	98.96	98.94
	70% -30%	99.04	98.84	99.01	99.10	99.02
SVM	90% -10%	99.28	99.13	99.31	99.28	99.28
	80% -20%	99.64	99.56	99.66	99.64	99.65
	70% -30%	99.52	99.42	99.51	99.53	99.51
Multinomial Logistic Regression	90% -10%	100.00	100.00	100.00	100.00	100.00
	80% -20%	99.64	99.56	99.66	99.64	99.65
	70% -30%	99.76	99.71	99.75	99.77	99.76
K-means	complete	36.23	23.48	36.23	39.56	37.29
SOM	complete	64.78	52.00	60.15	60.92	57.68

On the other hand, the unsupervised clustering techniques like K-means and SOM could not match the accuracy level of supervised models for the same set of data and features. SOM performed with around 65% accuracy level, kappa value of 52%, precision of 60.15%, recall of 60.92% and f1 score of 57.68% while classifying the labeled data correctly. In contrast, K-means achieved only around 36% of accuracy, 23.48% of Kappa, 36.23% of precision, 39.56% of recall and 37.29% of F1 score. Since, SOM performed better than K-means on every performance metric value, it is safe to consider than SOM performed better than K-means for the given dataset and conditions.

The best performing models for all the 5 (five) classifiers were chosen for its further comparison with SOM, which performed better among clustering techniques. The prediction of selected 5(five) classifiers on the sampled unlabeled dataset was compared with the outcome of the SOM on the same dataset. The percentage of labels commonly identified for the activities by the supervised models and SOM are illustrated in Table 6. as classification similarity.

Table 6: Similarity comparison of SOM and Supervised Models on unlabeled dataset

Classification Similarity (%)

Classifier	**Unlabelled data-set**
SOM and Decision Tree	7.82%
SOM and Naïve Bayes	22.17%
SOM and Random Forest	20.86%
SOM and SVM	23.47%
SOM and Logistic (Multinomial)	22.60%

The Table 6. illustrates that SOM and SVM achieved the highest similarity at 23.47 % which was closely followed by the logistic (22.60%), naive Bayes (22.17%) and random forest (20.86%) respectively. Decision tree classifier agreed the least with SOM (7.82%) on the prediction of the unlabeled dataset.

7 Conclusion and Future work

Mobile phones are compact yet powerful devices that possess the ability to perform numerous tasks. In this thesis, the different models were studied and compared. There are a number of factors that affects the performance of the models, including the quality of data, feature selection, train and test data selection, model parameter selection, sampling technique and choice of set seed. The supervised models seems to work far better and more accurately on the data set when compared with unsupervised ones.

This was expected as supervised models had learnt from the sample data to achieve high accuracy. Moreover, the special attention was given to avoid class imbalance while learning, which also helped the supervised models to achieve very high performance metrics.

The results show that supervised learning models had a low agreement with the unsupervised models. Therefore, further work is required in the future to check for improvement of the efficiency of these clustering techniques. In addition, there could be a possibility that different features selection for supervised and unsupervised learning can lead to a different result.

References

[1] Strategy Analytics, *"Strategy Analytics: Half the World Owns a Smartphone"*, Strategy analytics, 2021.Retrieved 2022-05-16.

[2] SNV Organization *"Healthy people are foundation healthy economies and societies"*, 2018

[3] Los Angles Times, *" What's the most popular workout in the world?"*,Los Angles Times,U.S, August 25,2015.

[4] Jennifer R. Kwapisz, Gary M. Weiss & Samuel A. Moore *"Activity Recognition using Cell Phone Accelerometers"*. ACM SIGKDD Explorations Newsletter Volume 12 Issue 2 December, 2010, doi.org/10.1145/1964897.1964918

[5] Wire sensor data mining, *"Wireless Sensor Data Mining"*, cis.fordham.edu, 2012, Retrieved 2022-05-16.

[6] Maurer, U., Smailagic, A., Siewiorek, D., & Deisher, M., *"Activity recognition and monitoring using multiple sensors on different body positions"*, In IEEE Proceedings on the International Workshop on Wearable and Implantable Sensor Networks, 3(5), 2006.

[7] Akram Bayat, Marc Pomplun, Duc A. Tran, *"A Study on Human Activity Recognition Using Accelerometer Data from Smartphones"*, The 11th International Conference on Mobile Systems and Pervasive Computing (MobiSPC-2014), Elsevier B.V, doi:10.1016/j.procs.2014.07.009, 2014.

[8] Gary M. Weiss, Kenichi Yoneda & Thaier Hayajneh, *"Smartphone and Smartwatch-Based Biometrics Using Activities of Daily Living' Performance using Efficient Machine Learning Technique"*,IEEE Access (Volume: 7),ISSN: 2169-3536, doi: 10.1109/ACCESS.2019.2940729, 2019.

[9] G. M. Weiss, J. W. Lockhart, *"The Benefits of Personalized Smartphone-Based Activity Recognition Models"*, April, 2014 doi: 10.1137/1.9781611973440.71.

[10] Girija Chetty, Mohammad Yamin *"Intelligent human activity recognition SCHEME for ehealth applications"*, Malaysian Journal of Computer Science. Vol. 28(1), 2015

[11] Ricardo Chavarriaga, Hesam Sagha, Alberto Calatroni, Sundaratejaswi Digumarti, Gerhard Tröster, José del R. Millán, Daniel Roggen. *"The Opportunity challenge: A benchmark database for on-body sensor based activity recognition"*, Pattern Recognition Letters, 2013

[12] Jeffrey W. Lockhart, Gary M. Weiss, Jack C. Xue, Shaun T. Gallagher, Andrew B. Grosner, and Tony T. Pulickal *"Design Considerations for the WISDM Smart Phone-Based Sensor Mining Architecture"*, Proceedings of the Fifth International Workshop on Knowledge Discovery from Sensor Data (at KDD-11), San Diego, CA, 2011.

[13] Waikato Environment for Knowledge Analysis, *"Machine learning algorithms for data mining tasks"*, cs.waikato.ac.nz, Retrieved 2022-05-16.

[14] Weiss, G. M., and Hirsh, *"H. Learning to predict rare events in event sequences"*, In Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining, AAAI Press, Menlo Park, CA, 359-363, 1998.

[15] Stan Brown, *“Measures of Shape: Skewness and Kurtosis”*,article in brownmath, 2022, Retrieved 2022-05-16.

[16] A.Traletti, *“diffinv”*, w3cub, Retrieved 2022-06-01.

[17] Omnicalculator, *“shannon-entropy”*, omnicalculator, Retrieved 2022-06-01.

[18] Andreas. MUuller, Andreas C, Sarah Guido, *“Introduction to Machine Learning with Python: A Guide for Data Scientists”*, ISBN:9352134575, 9789352134571, O’Reilly Media, Incorporated, 2018.

[19] M. Narasimha Murty, V. Susheela Devi, *“Pattern Recognition: An Introduction”*, ISBN:9788173717253, Orient BlackSwan, 2011.

[20] Anshul Saini, *“Decision Tree Algorithm – A Complete Guide”*, analyticsvidhya, 2021, Retrieved 2022-05-29.

[21] Gopalan, Bhuvaneswari. *“What is Entropy and Information Gain? How are they used to construct decision trees?”*, Numpy Ninja, 2020, Retrieved 2022-05-16.

[22] Sruthi E R, *“Understanding Random Forest ”*, analyticsvidhya, 2021, Retrieved 2022-05-28.

[23] Yi Archer Yang ,*“randomforest”*, McGill University, math.mcgill, Resources, page: 587, 2016, Retrieved 2022-06-2.

[24] Chirag Goyal, *“Multiclass Classification Using SVM”*, analyticsvidhya, 2021, Retrieved 2022-05-28.

[25] David Meyer, *“Support Vector Machines”*, cran.r-project.org, 2021, Retrieved 2022-05-28.

[26] e1071 (version 1.2-1), *“svm: Support Vector Machines”*, rdocumentation.org, Retrieved 2022-05-28.

[27] Eduardo García-Portugués *“Predictive Modeling”*, section A.3 Multinomial logistic regression, bookdown.org. ISBN 978-84-09-29679-8, 2022.

[28] Han, J., Kamber, M., & Pei, J. *“Data mining: Concepts and techniques, third edition (3rd ed.)”*. Morgan Kaufmann Publishers, 2012.

[29] MacKay, David *“An Example Inference Task: Clustering”*, Chapter 20, Information Theory, Inference and Learning Algorithms. Cambridge University Press. pp. 284–292. ISBN 978-0-521-64298-9. MR 2012999, 2003.

[30] David Rumelhart, Zipser David, James L. McClelland, et al. *” Parallel Distributed Processing”*, Vol. 1. MIT Press. pp. 151–193, 1986.

[31] Haykin, Simon. *“9. Self-organizing maps”*, Neural networks - A comprehensive foundation (2nd ed.), Prentice-Hall. ISBN 978-0-13-908385-3, 1999.

[32] A. Ciampi, Y. Lechevallier, *“Clustering large, multi-level data sets: An approach based on Kohonen self organizing maps”*, in D.A. Zighed, J. Komorowski, J. Zytkow (Eds.), PKDD 2000, Springer LNCS (LNAI), vol. 1910, pp. 353-358, 2000.

[33] Kohonen, Teuvo Honkela, Timo *“Kohonen network”*, Scholarpedia. 2 (1): 1568. Bibcode:2007SchpJ...2.1568K. doi:10.4249/scholarpedia.1568, 2011, Retrieved 2022-05-16.

[34] Landis, J.R. Koch, G.G. *"The measurement of observer agreement for categorical data"*. Biometrics 33 (1): 159–174, 1977

[35] Maarit Widmann, *"Cohen's Kappa: What It Is, When to Use It, and How to Avoid Its Pitfalls"*, thenewstack.io, 2020, Retrieved 2022-05-16.

[36] Issa Memari *"Precision, Recall, Accuracy, and F1 Score for Multi-Label Classification"*,article in Synthesio Engineering, 2021, Retrieved 2022-05-16.

[37] Jeffrey W. Lockhart,Gary M. Weiss, Jack C. Xue, Shaun T. Gallagher, Andrew B. Grosner, Tony T. Pulickal, *"Design Considerations for the WISDM Smart Phone-based Sensor Mining Architecture"*,Department of Computer and Information Science, Fordham University, 2011.

Appendices

A The dataset

A.1 Source

The data has been released by the Wireless Sensor Data Mining (WISDM) Lab.
(http://www.cis.fordham.edu/wisdm/)
The labelled and un-labelled data sets can be downloaded directly from the below website.
https://www.cis.fordham.edu/wisdm/dataset.php

A.2 Ethical consideration

A.2.1 Privacy and distribution

The data in this set were collected with Actitracker system, which is available online at http://actitracker.com and in the Google Play store. The data is not private and is available in public domain.

A.2.2 Cost

It is available for free.

A.2.3 Collection system and detailed information

The system is described in the paper [37].

A.2.4 Special request

When using this dataset, it was requested to cite the paper [37].

B Tables

B.1 Comparative table of previous works (as mentioned in literature review)

Table 7: Comparative table of previous works (as mentioned in literature review)

Paper	Authors	Sensors	Activities	Features used	ML Algorithms
Activity Recognition using Cell Phone Accelerometers [4]	Jennifer R. Kwapisz, Gary M. Weiss, Samuel A. Moore	Accelerometer	A4,A24,A5,A6, A2, A1	MV, SD, AAD, ARA, TBP, BD.	Decision Trees Logistic Regression Multilayer Neural Networks
Activity Recognition and Monitoring Using Multiple Sensors on Different Body Positions [6]	Uwe Maurer, Asim Smailagic, Daniel P. Siewiorek, Michael Deisher	Accelerometer (eWatch)	A1, A2, A4, A5, A6, A25	MV, RMS, SD, VAR, AAR, BD, P,IR,ZCC, MCC,ARA	Decision Tree, KNN, NB, Bayes Net
Smartphone and Smartwatch-Based Biometrics Using Activities of Daily Living [8]	Gary M. Weiss , Kenichi Yoneda, Thaier Hayajneh	Accelerometer, Gyroscope, Magnetometer	A1,A2,A4,A5, A6,A25,A9,A28, A29,A30,A31, A32,A33,A34, A35	ARA, SD, AAD, BD,TBP	Random Forest, Decision Tree, KNN
A Study on Human Activity Recognition Using Accelerometer Data from Smartphones [7]	Akram Bayat, Marc Pomplun, Duc A. Tran	Accelerometer	A5, A6, A36, A37, A38, A25	MZ, MN, MX, MV,RMS ,SD, VARZCC, MCC	Multilayer Perceptron, SVM, Random Forest, LMT , Simple Logistic Logit Boost
The Benefits of Personalized Smartphone-Based Activity Recognition Models [9]	Gary M. Weiss, Jeffrey W. Lockhart	Accelerometer	A4, A24, A39, A1, A2,A3	ARA, SD, AAD, BD,TBP	Random Forest, Decision Tree, Instance based Learning, Neural network Naïve Bayes, Rule Induction Logistic Regression
Intelligent human activity recognition scheme for eHealth applications [10]	Girija Chetty and Mohammad Yamin	Accelerometer, Gyroscope	A1, A2, A3, A4, A5,A6	MV, SD, SMA, SE, C	Random Forest, Decision Tree, K-means, Random Committee, Decision Tree, Lazy IBk, Naive Bayes

Activities: A1-Standing, A2-Sitting, A3-Lying Down, A4-Walking, A5-Climbing Up, A6-Stepping Down, A7-Sit Ups, A8-Vaccuming, A9- Brushing, A10-Driving, A11-Inactive, A12-Cleaning, A13-Cooking, A14-Medication, A15 Sweeping, A16-Washing Hands, A17-Watering Plants, A18-Tai Chi Movements, A19-Hammering, A20-Screwing, A21-Spanner Using, A22-Power Drill, A23-Martial Arts

Movements, A24- Jogging, A25-Running, A26-Cycling, A27-Biking, A28- Kicking a soccer Ball,A29- Basketball dribbling,A30- Catch a Tennisball(underhand) , A31-Typing, A32-Writing, A33- Clapping, A34- Folding clothes, A35- Eating , A36- Slow Walking, A37 - Fast Walking, A38- Dancing(Aerobics), A39 -Stairs (up or down).

Features: MV-Mean value, SD-Standard Deviation, VAR-Variance, MD-Median, MX-Maximum, MN-Minimum, SMA-Signal Magnitude Area, RMS-Root Mean Square, IR-Interquartile Range, SE-Signal Entropy, AC-Autorregresion Coefficients, CC-Correlation Coefficient, LFC-Largest Frequency Component, WAFS-Frequency Signal Weighted Average, FS-Frequency Signal, S-Skewness, K-Kurtosis, EF-Energy of a Frequency, IAV-Interval Angle between two Vectors, AAD-Average Absolute Difference, ARA-Average Resultant Acceleration, TBP-Time Between Peaks, BD-Binned Distribution, E-Energy, C-Correlation, MoA-Magnitude of Acceleration, CASC-Cross-axis Signals Correlation, FFT-Fast Fourier, MZ- Mean across Z axis, Transform, SpE-Spectral Energy, EFFT-Frequency domain Entropy log of FFT,ZCC-Zero-Cross Correlation,MCC-Zero-Cross Correlation, V-Variance,P-75% Percentile, FRP-Frequency Range Power, Q-Quartiles, AV-Absolute Value, FE-Frequency Entropy.