# JTD 792

# Minor Project

30th August, 2020

**Wearable System for predicting Freezing of Gait in people affected by Parkinson's disease**

(Supervisor: Prof. S.D Joshi)


**Indian Institute of Technology, Delhi**

Submitted By:

Ayush Gupta (2019JTM2673)

2019-21

# Contents

## List of Figures

## Abstract

There were wearable systems that can detect Freezing of Gait (FoG) in people affected by Parkinson's Disease. The FoG is an episodic gait disturbance that affects locomotion in Parkinson's Disease. These systems can also generate external force to the patient so that patient can start their motion. But there is not a solution that can predict the occurrence of FoG in advance. To predict the FoG in advance, we are implementing an algorithm to predict the occurrence of FoG in advance. This wearable system uses on-body acceleration sensors to measure the patients' movements. The patient's gait was monitored by three tri-axial accelerometer sensors worn on the back, hip, and ankle of the patient. The gait featured is extracted from the accelerometer's signal, through data windowing and dimensional reduction. We are getting the accuracy of prediction with the K-NN algorithm is 84.9% with PCA and 85.2% with LDA, respectively. This system has future applications that it can implement in mobile devices and wearable fitness bands that can give rhythmic stimuli to the patient before the occurrence of FoG so that any accident to the patient can be avoided. To implement the prediction model, we are using data from the Daphnet Freezing of Gait Dataset. The Daphnet Freezing of Gait Dataset in users with Parkinson's Disease is a dataset used to recognize freeze of gait from accelerometer sensors, which are placed on the back, hip, and ankle the patient.

# 1 Introduction

## 1.1 Parkinson's Disease:

Parkinson's Disease is a neurodegenerative disorder that affects locomotion activity in the body. There are symptoms in Parkinson's Disease, such as termor, shaking, and difficulty in walking. The symptoms of this disease begin gradually, but it becomes so much dangerous for the patient after the time. It is caused by dying of nerve cells, which mainly responsible for movement in the body. Due to decaying these neurons, there is a lack of dopamine chemical that affects the whole movement in the body. This disease mainly arises with the age of the person. It is seen that these diseases occur in men more as compared to women. The main problem with this disease is that there are no medical tests to detect it in many cases, so there is difficulty in diagnosing it accurately. The falls are the most complications of this disease, which leads to increase risk of hospitalization and disability in the patient.

## 1.2 Freezing of Gait ( FoG ):

The most of the fall in PD patients is contributed by Freezing of Gait only. FoG is an episodic gait disturbance that effects shown up to 30 seconds. When the patient suffered FoG, his feet froze to the ground. FOG typically manifests as a sudden and transient inability to move. The FOG is dependent on the environment; its effect is shown much when the patient gets nervous. At the time of FOG, the brain is incapable of coordinating motion activity, which results that PD patients can not move their feet despite their intention to move. This symptom commonly occurs in gait initiation, turning, passing through narrow space, or approaching obstacles in patients' daily lives, significantly increasing the risk of falling during walking. The facts that FOG often happens suddenly, asymmetrically, and with a short duration make the clinical detection, tracking, and evaluation of the onset of FOG a challenging task.

### 1.3 Wearable Assistant for FoG Detection Rhythmic Auditory Cueing:

Different treatment methods have been proposed for FoG, but results are not up to mark and its required more effective solution to prevent its cause. The pharmacological treatment for FOG is not so effective. In this type of treatment, patients respond to levodopa, monoamine oxidase inhibitors, deep brain stimulation. In this treatment, levodopa is used to manage motor symptoms in Parkinson's disease patients. These treatments mainly depend on the clinician and the descriptions' ability of the patients. Therefore, an accurate and automatic FOG detection approach is desired to detect or even objectively predict FOG's onset in advance. **Wearable devices** are the right solution for the assessment and treatment of freezing gait in Parkinson's disease.

With wearable technologies, it is easy to obtain insightful information on gait characteristics of the patient. These devices can easily detect the FoG. Wearable devices are more suitable for long-term monitoring of PD in living conditions because of its light-weight, and comfort. These wearable devices mainly dealt with recognizing of FOG events based on offline datasets gathered from wearable devices through different sensors such as accelerometers, gyroscopes, EEG sensors. These devices can also provide rhythmic visual, sensory, or auditory stimuli to the patient upon FoG detection to reduce its duration. When the patient is externally stimulated with Rhythmic Auditory Stimulation (RAS), the patient can quickly exit from FoG. In this wearable device, three accelerometer sensors were used to extract the gait features of the patient. All of the body's motion depends on the hip, back, and ankle of the body. So these sensors have been placed on the back, hip, and ankle of the patient. The raw signals of these accelerometers can be used for recognizing and predicting of freezing of gait.

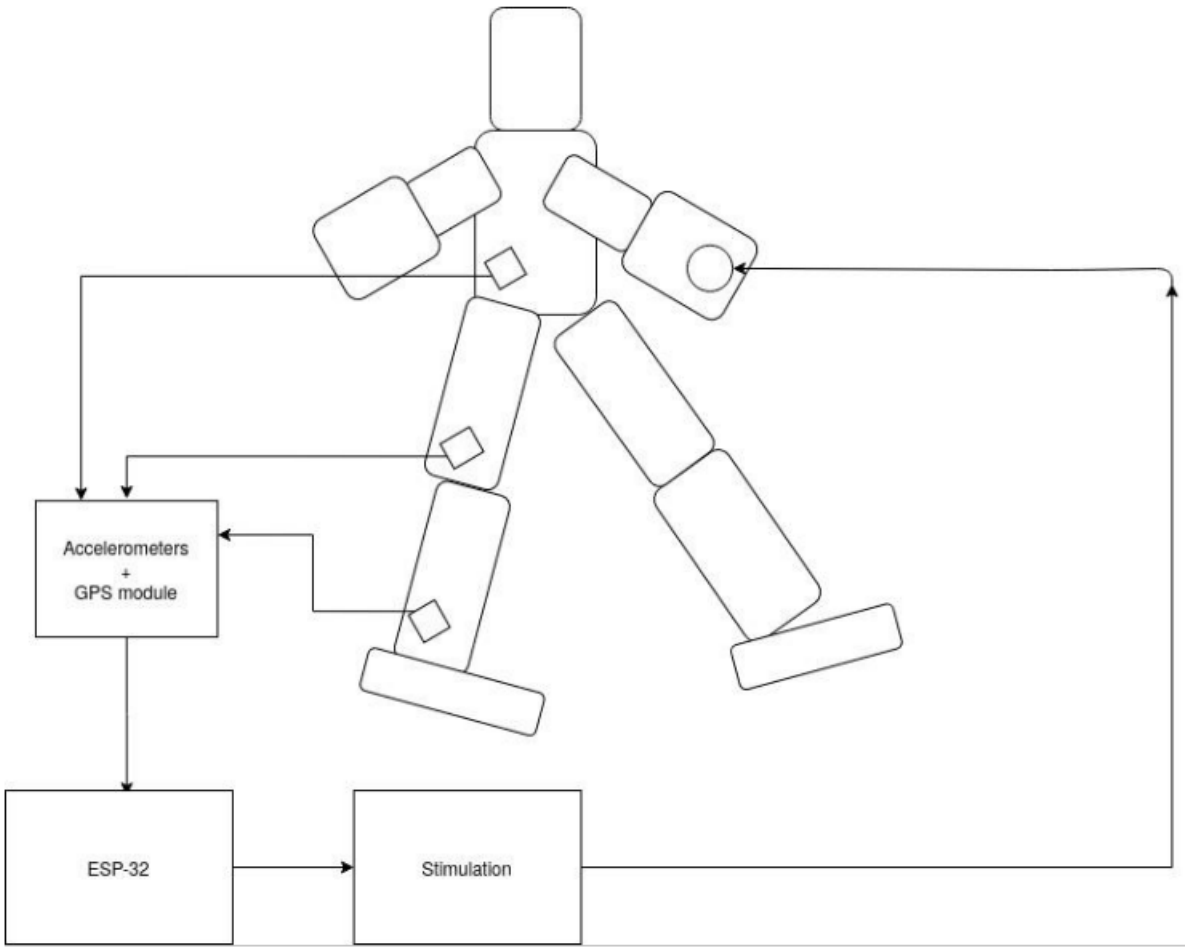


Figure 1: Wearable assistant with Accelerometer

**Rhythmic auditory stimulation (RAS)** is one of the neurological therapy methods that can free the patient from freezing. It is physiological effects that improve movement control in the patient. Through this stimulation, patients get guidance to hit the ground with their feet, and it also provides an external cue so that patient can start their movement. Rhythm is an essential element of motor movement, including motor control and output, because rhythmic auditory cueing facilitates movement by providing a mechanism for planning movements.

## 2 Accelerometer Sensor

The accelerometer is a microelectronic mechanical device that measures the force of acceleration due to gravity in the 'g' unit. Acceleration is the change in velocity with respect to time, and velocity is related to the position of the moving object, so with the help of the object's acceleration position can easily be determined. Acceleration sensors can be used in applications requiring tilt sensing. Accelerometers are Micro Electronic Mechanical System (MEMS) that can sense either static or dynamic acceleration forces. The gravity acceleration is an example of static force; on the other hand, dynamic forces include vibrations and movements. Static acceleration due to gravity can easily find out the object's tilt position with respect to ground. With the help of the amount of dynamic acceleration, we can analyze the way of moving devices. The accelerometer's working principle is a simple mass-spring system in which the object's acceleration depends on the spring's displacement.

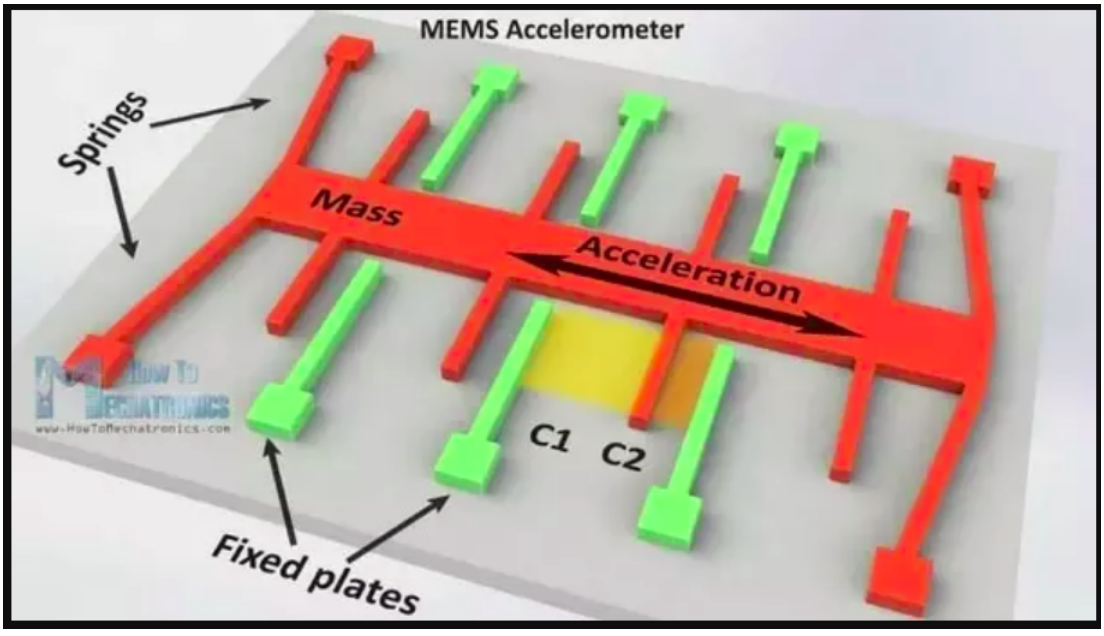


Figure 2: Working of Accelerometer

So, through this, we can determine position as a function of acceleration. The piezoelectric effect is applied to this spring-mass system, which converts mechanical motion into

the electrical signal. In this effect, one capacitor plate is attached to the object in the mass-spring system, and the other capacitor plate is fixed. When the mass moved, then the capacitance between plates would also change. Through this, we get a change in capacitance as a function of acceleration. Hence in this way, the mechanical signal can easily be converted into electrical signals. The accelerometer can measure position in all 3-dimension with respect to ground.

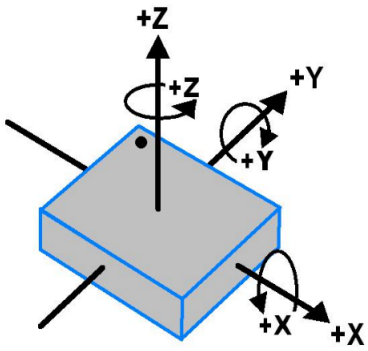


Figure 3: Tri-axial Accelerometer

## 3 Detection of Freezing

There are different statistical methods to detect FoG, which uses statistical features of raw signals from wearable devices' sensors. Freezing can be easily detected through the power spectra of the raw signal of the accelerometer. To detect FoG, we can determine a simple threshold that can discriminate between freezing and the normal locomotion. **Freezing Index** defined as the ratio of power spectral densities in freeze band (3-8 Hz) and locomotion band (0.5-3 Hz) to characterize FOG using frequency domain information. It would be threshold value, which can discriminate between freezing and non- freezing. A wearable system can be designed based on FI's value so that it can automatically detect FoG and apply a rhythmic auditory intervention to help patients resume walking while FoG is happening. The freezing index can improve the classification accuracy. Based on the freezing index, different machine learning methods can be applied to predict freezing in advance.

# 4 Prediction of FoG from dataset

There were many techniques provided to detect the freezing of gait. But there is less research done to predict the occurrence freezing of gait in advance. If the prediction of FoG has been made in advance, then it can avoid any accident to the patient. There should be a well-defined machine learning model to predict any class of events. Every model must be learned efficiently in all kinds of real applications, such as classification and clustering. Many algorithms are used for classification, such as k- Nearest Neighbours (k-NN), Support Vector Machine (SVM), Naive Bayes, neural network model. The k-NN algorithm is so simple, has high performance, and easy to implement for large data. Hence, we will use the k-NN model to predict the freezing of gait based on the DAPHNET dataset. We will use it to classify the data into FoG and no FoG category.

## 4.1 k-Nearest Neighbor Algorithm:

k-NN is one of the simplest of all machine learning algorithms. k-NN is used in pattern recognition for classification and regression. k-NN can be used for both classification and regression. The training datasets are used to train the model; then, this model can predict the test data class. k-NN has been used in statistical estimation and pattern recognition. There are different measures for distance calculation like Euclidean, Manhattan distance, and Chebyshev. Among all these, Euclidean is the most popular choice to measure the distance between the two points. In k-NN, the k is the number of neighbours which is assigned to unknown data by measuring its Euclidean distance. Euclidean distance (d) between two points x and y of M dimensions is given by:

$$d(x, y) = \sqrt{\sum_{i=1}^{M} (x_i - y_i)^2}$$

### 4.2 k-NN Classification:

k-NN classification is to classify an unknown object from known objects. Let us consider a simple example, and there are plus and minus signs and the query point with a red circle, as shown in figure 4. We need to find the class of the red circle, whether it belongs to plus or minus, by calculating the distance from the query point(red dot) to each point(plus and minus). Depending on the k value, the class of the object changes accordingly.

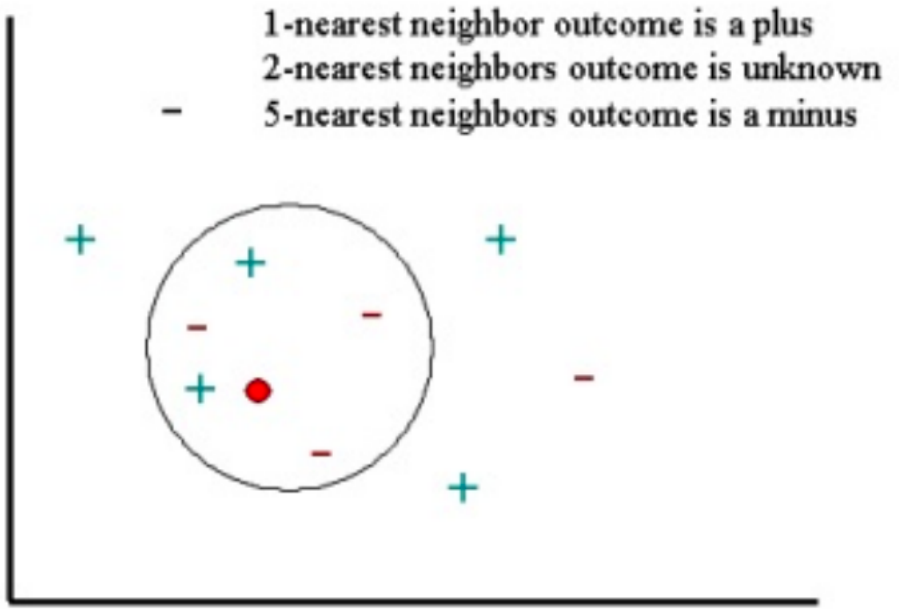


Figure 4: k-NN Classification

### 4.3 k-NN Regression:

k-NN regression is the simple implementation to calculate the average value of the k-Nearest Neighbors. Both k-NN regression and k-NN classification uses the same formula. Regression is used when we have to classify in the continuous data.

### 4.4 Steps in the k-NN algorithm:

The k-NN algorithm can be implemented in the following steps:

1. Firstly, load the data for training and testing of the model
2. Choose the value of k; the value of k should be moderate if its value is so low that there would be so much noise. On the other hand, if its value is so high, then there would be a processing issue
3. For better results, $k = \sqrt{n}$ and it should be an odd value
4. Divide dataset into test and training data
5. Iterate from 1 to the total number of training data points
    - Apply Euclidean distance to calculate the distance between the test point and training point
    - Sort the calculated distances in ascending order based on distance values
    - Get top k rows from the sorted array
    - Get the most frequent class of these rows
    - Return the predicted class.

## 5 Methodology

To predict the FoG in advance, we would apply the method to reach to our result. There are the following steps that would be followed to predict the Freezing of Gait in wearable devices. The whole workflow can be understood from the figure 5, which we will use in wearable device.

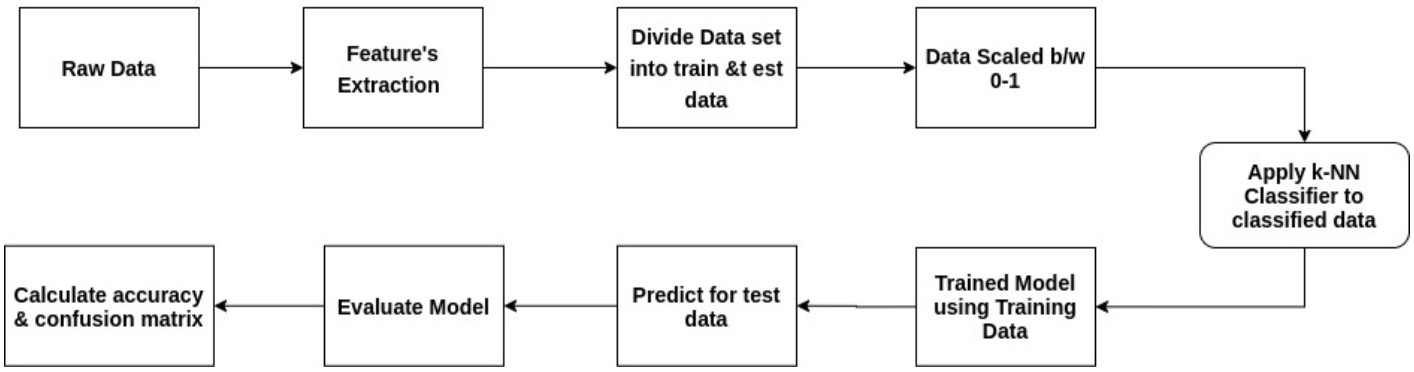


Figure 5: Workflow to predict FOG

### 5.1 Gait Monitoring

The patient's gait is monitored by three tri-axial accelerometer sensors worn on the back, hip, and ankle of the patient. The features for classification are extracted from the raw data of the accelerometer through the dimensionality reduction technique. We are using data from the Daphnet Freezing of Gait Dataset in users with Parkinson's disease. This dataset consists of data from different ten users in which users 4 and 10 did not show any freeze. This data is in the form of the matrix with one line per sample and one column per channel. The channels are as follows:

1. Time of sample in the millisecond
2. Ankle acceleration - horizontal forward acceleration [mg]
3. Ankle acceleration - vertical [mg]
4. Ankle acceleration - horizontal lateral [mg]
5. Hip acceleration - horizontal forward acceleration [mg]
6. Hip acceleration - vertical [mg]
7. Hip acceleration - horizontal lateral [mg]
8. Back acceleration - horizontal forward acceleration [mg]
9. Back acceleration - vertical [mg]
10. Back acceleration - horizontal lateral [mg]

11. Annotations

- Annotations 0: means it is not part of experiment
- Annotations 1: No freeze
- Annotations 2: Freezing

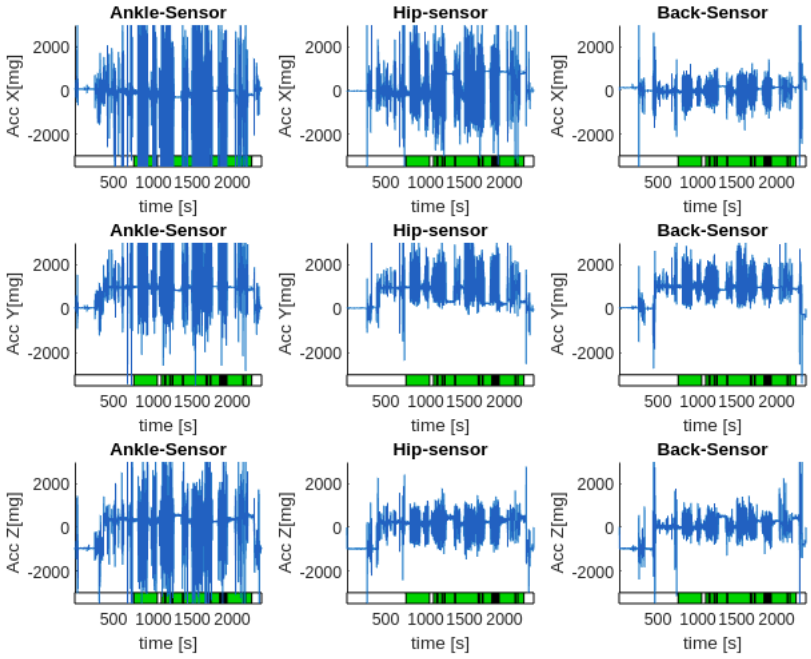


Figure 6: Display of the sensors readings

## 5.2 Feature Extraction

In most classification, firstly, raw data should be projected into different feature spaces so that classification can be made easier. This pre-processing stage is known as feature extraction. In this stage, the dataset's dimension is reduced so that model learns quickly, and hence by this processing speed increases. There are many dimension reduction techniques, such as PCA, LDA, KLDA.

1. **Principal Component Analysis (PCA):** It is the most common method for reducing the dimensionality of the large set of possibly correlated features. To design any model, there must have more data to predict accurately. But when data is so much large, it can be possible that redundancy in features of the dataset will increase the computation time. So to avoid this, PCA would be a better solution. This technique figures out co-relations and patterns among the datasets after finding out co-relations, it transformed the original feature space into a new space with most data variation. There are steps which are used to compute PCA:
    - Standardization of the data
    - Computing the covariance matrix
    - Calculating the eigenvectors and eigenvalues
    - Computing the Principal Components
    - Reducing the dimensions of the data set
2. **Linear Discriminant Analysis (LDA):** This technique also reduces the dimensionality of the dataset. It is based on Fisher's linear discriminant, a method to find out the linear combination of features that separate samples of different classes. LDA is more effective than PCA when it comes to a multi-class problem, but it requires to know in advance the label of each training data. LDA creates a new axis on which it projects the information from the different features to minimize the variance and maximize the distance between means of the two classes.

### 5.3 Classification

Standard classification amounts to deciding which class among a class pool does explain a new observation the best. Classification is depended strongly on the feature extraction. It is usually based on the training phase, where a classification algorithm builds a classifier by analyzing a set of D-dimensional data known as the training set. Here, the dataset is in the form of a matrix that contains different 9- features and one target state. In target, it is

indicating by 0,1 and 2, which means the experiment is not started, indicating no freeze and freeze, respectively. Training data do the learning of classifiers, so we will divide our dataset into a training dataset and testing dataset. We have different 9 features which should be scaled between 0 to 1 so that classifier can work accurately. We are classifying our data into three classes no FoG, FoG, and pre-FoG with the help of a k-NN classifier.

### 5.4 Prediction

We are predicting testing data from our k-NN model; if the predicted value is 0, it means freezing will not come in the patient, and on the other hand, if the predicted value is one, that means the patient will get FoG. This classifier is also evaluating different dimension reduction techniques and comparing its accuracy and computing time.

## 6 Result and Future Work

We are getting accuracy to predict FoG with the k-NN classifier is about 90% and computing time around 6 sec with different dimensionality reduction techniques. It is found that the k-NN algorithm is less complicated, and it can easily be implemented with extensive data. We had applied our algorithm on the offline dataset, but there is a requirement that it should also be used with online (real-time) data so that this application can be placed in the smartphone of the user.

# 7 Appendix

## 7.1 k-NN Classifier

```
#.........Prediction of FoG  by using K-NN.....#
#........author@ Ayush Gupta ................#
#.........Minor Project ..........................#

# importing all library
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import time
t0= time.clock()



from sklearn import datasets
#...knn classifier ......#
from sklearn.neighbors import KNeighborsClassifier

#Import train_test_split function
from sklearn.model_selection import train_test_split    #dividing
from sklearn.preprocessing import StandardScaler

#Import scikit-learn metrics module for accuracy calculation
from sklearn import metrics
from sklearn.metrics import confusion_matrix
from sklearn.metrics import f1_score
from sklearn.metrics import accuracy_score
```

```
# importing PCA for dimensional reduction..
from sklearn.decomposition import PCA
# lda library
from sklearn.discriminant_analysis import LinearDiscriminantAnaly
# loading csv file into data.
data = pd.read_csv("Book1.csv")
 #  extracting features from data.
features= data.iloc[:, 1:10]
 #  extracting target  from data.
target=data.iloc[:, 10]

# splitting data into 70% training and 30% test
X_train, X_test, y_train, y_test = train_test_split(features, tar

#..........scaling data...
sc_feature=StandardScaler()
# train data after standardisation
X_train= sc_feature.fit_transform(X_train)
X_test=sc_feature.transform(X_test)

t1=0
t2=0
# print("train target:",y_train)
# print("test target:",y_test)

#............prediction using k-NN without PCA..............

def kNN_withoutPCA(traindata, testdata):
    print("............k-NN without PCA..........")
```

```
print(len(traindata))
import math
k=math.sqrt(len(y_test))
print(k)

#...Creating  KNN Classifier
#check for accuracy when no. of neighbour=5 or 7
knn = KNeighborsClassifier(n_neighbors=345, metric='euclidean

#..Training the model using the training sets
knn.fit(traindata , y_train)

#..Predict the response for test dataset
print("X_test:", X_test)
y_pred = knn.predict(testdata)
print("Prediction with k-NN without pca:",y_pred)



#..Evaluating  Model...
cm=confusion_matrix(y_test , y_pred)
print("Confusion of k-NN without PCA:\n",cm)



# Model Accuracy, how often is the classifier correct?
#### for measuring accuracy ###
print("Accuracy:",metrics.accuracy_score(y_test , y_pred))
# print(f1_score(y_test , y_pred))
```

```
    t1 = time.clock() - t0          # time analysis
    print("Time elapsed: ", t1) # CPU seconds elapsed (floating p
    return y_pred, t1

#.................k-NN with PCA.....................

def knn_PCA(traindata, testdata):
    print(".............KNN with PCA.................")

    # fit and transform data  dimension reduction
    pca = PCA(n_components=2)

    # dimensional reduction with pca
    X_train_pca = pca.fit_transform(traindata)
    X_test_pca = pca.transform(testdata)
    print(len(X_train_pca))
    import math
    k=math.sqrt(len(y_test))
    print(k)

    #...Creating  KNN Classifier
    knn = KNeighborsClassifier(n_neighbors=345,

    # Model Accuracy, how often is the classifier correct?
     #### for measuring accuracy ###
    print("Accuracy:",metrics.accuracy_score(y_test, y_pred_pca))
```

```
    # print(f1_score(y_test, y_pred))




    t2 = time.clock()-t0     # time analysis
    print("Time elapsed: ", t2) # CPU seconds elapsed (floating p
    return y_pred_pca, t2


#.........................kNN with LDA.....................

def knn_LDA(traindata, testdata):
    print(".............KNN with LDA.................")

    # fit and transform data  dimension reduction
    lda = LDA(n_components=2)
    #train data after lda reduction
    X_train_lda = lda.fit_transform(traindata, y_train)
    X_test_lda = lda.transform(testdata)




    print(len(X_train_lda))
    import math
    k=math.sqrt(len(y_test))
    print(k)
```

```
#...Creating  KNN Classifier


#check for accuracy when no. of neighbour=5 or 7
knn = KNeighborsClassifier(n_neighbors=345, metric='euclidean

#..Training the model using the training sets
knn.fit(X_train_lda, y_train)

#..Predict the response for test dataset
print("X_test_lda:", X_test_lda)
y_pred_lda = knn.predict(X_test_lda)
print("Prediction with k-NN with lda:",y_pred_lda)



#..Evaluating  Model...
cm=confusion_matrix(y_test, y_pred_lda)
print("Confusion of k-NN with LDA:\n",cm)



# Model Accuracy, how often is the classifier correct?
#### for measuring accuracy ###
print("Accuracy:",metrics.accuracy_score(y_test, y_pred_lda))
# print(f1_score(y_test, y_pred))



t3= time.clock() - t0   # time analysis
print("Time elapsed: ", t3) # CPU seconds elapsed (floating p
return y_pred_lda,t3
```

```
time1=0
time_pca=0
time_lda=0



y_knn, time1 =kNN_withoutPCA(X_train, X_test)

y_knn_pca, time_pca=knn_PCA(X_train, X_test)

y_knn_lda, time_lda=knn_LDA(X_train, X_test)


print("t0:", t0)
print("time without dimension reduction:", time1)
print("time after pca:", time_pca-time1)
print("time after lda:", time_lda-time_pca)
print("t2:", t2)
```

## References


[1] Accelerometer. https://www.youtube.com/watch?v=eqZgxR6eRjo.

[2] M. Bachlin, M. Plotnik, D. Roggen, I. Maidan, J. M. Hausdorff, N. Giladi, and G. Troster. Wearable assistant for parkinson's disease patients with the freezing of gait symptom. *IEEE Transactions on Information Technology in Biomedicine*, 14(2):436–446, 2010.

[3] F. Demrozi, R. Bacchin, S. Tamburin, M. Cristani, and G. Pravadelli. Towards a wearable system for predicting the freezing of gait in people affected by parkinson's disease. *IEEE Journal of Biomedical and Health Informatics*, pages 1–1, 2019.

[4] edureka. https://www.edureka.co/blog/principal-component-analysis/.

[5] Y. Guo, L. Wang, Y. Li, L. Guo, and F. Meng. The detection of freezing of gait in parkinson's disease using asymmetric basis function tv-arma time–frequency spectral estimation method. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 27(10):2077–2086, 2019.

[6] k-NN algorithm. https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/.

[7] Towards Data Science. https://towardsdatascience.com/knn-k-nearest-neighbors-1-a4707b24bd1d.