SUPERVISED LEARNING BASED HUMAN GAIT RECOGNITION

Project submitted

In partial fulfilment of the requirements for the degree of

MASTER OF COMPUTER APPLICATION

By

Rakib Laskar

Roll-002010503044

Reg.No-154252 of 2020-2021

Exam Roll-MCA2360020

Under the supervision of

PROF. SUSMITA GHOSH

Department of Computer Science & Engineering

Faculty of Engineering and Technology

Jadavpur University, Kolkata-700032, India

May, 2023

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

FACULTY OF ENGINEERING AND TECHNOLOGY

JADAVPUR UNIVERSITY

TO WHOM IT MAY CONCERN

I hereby recommend that the project entitled **“SUPERVISED LEARNING BASED HUMAN GAIT RECOGNITION”** prepared under my supervision by Rakib Laskar (Reg. No.154252 of2020-2021, Roll No:002010503044, Exam Roll-MCA2360020) may be accepted in partial fulfilment for the degree of Master of Computer Application in the Department of Computer Science and Engineering, Jadavpur University.

....

Professor Susmita Ghosh

Project Supervisor

Department of Computer Science and Engineering

Jadavpur University

....

Professor Nandini Mukhopadhyay

Head

Computer Science and Engineering

Jadavpur University

....

Professor Ardhendu Ghosal

Dean

Faculty of Engineering and Technology

Jadavpur University

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

FACULTY OF ENGINEERING AND TECHNOLOGY

JADAVPUR UNIVERSITY

CERTIFICATE OF APPROVAL

The foregoing project is hereby accepted as a credible study of an engineering subject carried out and presented in a manner satisfactory to warrant its acceptance as a prerequisite to the degree for which it has been submitted. It is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein, but approve the project only for the purpose for which it is submitted.

____

Supervisor

____

External Examiner

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

FACULTY OF ENGINEERING AND TECHNOLOGY

JADAVPUR UNIVERSITY

DECLARATION OF ORIGINALITY AND COMPLIANCE OF ACADEMIC ETHICS

I hereby declare that this project entitled "**SUPERVISED LEARNING BASED HUMAN GAIT RECOGNITION**" contains literature survey and original research work by the undersigned candidate, as part of his Master of Computer Application studies.

All information in this document has been obtained and presented in accordance with academic rules and ethical conduct.

I also declare that, as required by these rules and conduct, I have fully cited and

referenced all materials and results that are not original to this work.

Candidate's Name: Rakib Laskar

Roll No: 002010503044

Project Title: SUPERVISED LEARNING BASED HUMAN GAIT RECOGNITION

Data:

Signature with Date:

____

ACKNOWLEDGEMENT

I wish to record my heartfelt gratitude towards all the people who have helped me in completion of this project.

I offer sincere thanks to my guide Prof. Susmita Ghosh for her kind support, inspiration and thoughtful insights.

I offer sincere thanks to Professor Nandini Mukhopadhyay, Head of the Department Computer Science and Engineering, Jadavpur University, for his continuous support and guidance throughout the project duration.

I am extremely thankful to Department of Computer Science and Engineering, Jadavpur University for providing all the facilities and necessary help towards the development of this project.

I am grateful to my parents, family members and friends for their continuous support, inspiration, encouragement and blessings without which this project would not have been success.

Rakib Laskar

Registration No.: 154252 of 2020-2021

Roll No.: 002010503044

Department of Computer Science & Engineering

Jadavpur University

INDEX

List Of Table

List of Fig

1. Introduction

1.1 Human gait classification

Human gait activity recognition is an emerging field of motion analysis that can be applied in various application domains [1]. One of the most attractive applications includes monitoring of gait disorder patients, tracking their disease progression and the modification/evaluation of drugs. This paper proposes a wearable gait motion data gain system that allows either the classification of recorded gait data into desirable activities or the identification of common risk factors, thus enhancing the subject's quality of life. Gait motion information was acquired using accelerometers and gyroscopes mounted on the lower limbs, where the sensors were exposed to inertial forces during gait. Additionally, leg muscle activity was measured using strain gauge sensors [2]. As a matter of fact, we wanted to identify different gait activities within each gait recording by utilizing Machine Learning algorithms. In line with this, various Machine Learning methods were tested and compared to establish the best-performing algorithm for the classification of the recorded gait information. The combination of attention-based convolutional and recurrent neural networks algorithms outperformed the other tested algorithms and was individually tested further on the datasets of five subjects and delivered the following averaged results of classification: accuracy, precision, and F1-score. Moreover, the algorithms were also verified with the successful detection of freezing gait episodes in a Personality disorder patient. The results of this study indicate a feasible gait event classification method capable of complete algorithm personalization [3].

1.2 Why gait classification is important?

Walking is something that the average person probably does not give much thought. It is our most basic method of transportation, but an inability to walk or be mobile can drastically change a person's life. It can impact our independence and create significant health problems over both the short and long term.

Many people can move about with abnormal or asymmetrical gait patterns for years without any symptoms. However, when someone experiences an injury or pain, normal gait can be altered, resulting in abnormal walking that can lead to bigger health issues[4].

This is why gait analysis is important. When we study the way a person walks or runs, we can identify individuals' unique movements, determine normal gait patterns, diagnose issues causing pain, and implement and evaluate treatments to correct abnormalities.

1.3 Classification

The Classification algorithm is a Supervised Learning technique that is used to identify the category of new observations based on training data. In Classification, a program learns from the given dataset or observations and then classifies new observation into several classes or groups. Such as, Yes or No, 0 or 1, Spam or Not Spam, cat, or dog, etc. Classes can be called as targets/labels or categories [2].

Unlike regression, the output variable of Classification is a category, not a value, such as "Green or Blue", "fruit or animal", etc. Since the Classification algorithm is a Supervised learning technique, hence it takes labelled input data, which means it contains input with the corresponding output [5].

In classification algorithm, a discrete output function(y) is mapped to input variable(x).

y=f(x), where y = categorical output

The main goal of the Classification algorithm is to identify the category of a given dataset, and these algorithms are mainly used to predict the output for the categorical data.

The algorithm which implements the classification on a dataset is known as a classifier. There are two types of Classifications:

- Binary Classifier: If the classification problem has only two possible outcomes, then it is called as Binary-Classifier.
  Examples**:** YES or NO, MALE or FEMALE, SPAM or NOT SPAM, CAT, or DOG, etc.
- Binary Classifier: If the classification problem has only two possible outcomes, then it is called as Binary-Classifier.
  Examples: YES or NO, MALE or FEMALE, SPAM or NOT SPAM, CAT or DOG, etc.

Classifier approaches for human gait classification

A classification strategy was implemented for the automatic identification of pathological gaits based on IMU data. The classification framework was defined through three major steps:

(i) the classification process started from implementing class-specific HMMs, whose likelihoods of observing data given each model were evaluated;

(ii) the HMM likelihoods were included in a wider feature set, which was then classified using an SVM classifier;

(iii) a majority voting (MV) classification post-processing was cascaded to summarize the results obtained in step *ii*.

1.4 Scope of Project: -

Machine Learning techniques have been used throughout this project and so this project contains the results of Machine Learning algorithms such as K-Nearest Neighbourhood, Linear Support Vector Machine, Random Forest. Report contains

analysis of the results of the classifiers based on these algorithms.

1.5 Organization of the Report: -

The Project is organized as follows: Chapter 2 contains the history of Human Gait Classification and detailed description of existing techniques on Extractive Summarization. Chapter 3 includes the discussion of different classification techniques. Chapter 4 contains implementation details of algorithms which are used to develop the summarization process, presented stepwise. Chapter 4 include the results and discussions. Finally, the conclusion and future work are put in Chapter 5.

CHAPTER-2

LITERATURE SURVEY

2.1 Background: -

Human gait classification is the process of identifying and classifying the different walking patterns of humans based on various parameters such as step length, stride duration, and walking speed. It is an important research area in the fields of biomechanics, medical diagnosis, and rehabilitation [6].

Human gait analysis has been used in a variety of applications, including the identification of gait abnormalities caused by neurological or musculoskeletal disorders, the assessment of sports performance, and the design of prosthetic devices [7]. By analyzing the gait pattern, it is possible to identify various characteristics of an individual's walking style, which can help in the diagnosis and treatment of various medical conditions.

Gait classification can be done using various methods, including machine learning algorithms and computer vision techniques. Machine learning algorithms can be used to classify gait patterns based on various features extracted from the gait data, such as step length, stride duration, and walking speed. Computer vision techniques can be used to analyse video recordings of individuals walking and extract various features related to gait [8].

Overall, human gait classification is an important research area that has the potential to improve our understanding of the human body and help in the diagnosis and treatment of various medical conditions.

2.2 Machine learning: -

Machine learning is a type of artificial intelligence (AI) that enables computers to learn and improve their performance on a task over time without being explicitly programmed [9].

In general, machine learning is a branch of artificial intelligence that focuses on the development of algorithms and models that can learn from data, without being explicitly programmed. The goal is to create models that can automatically identify patterns and relationships in data, and then use these patterns to make predictions or decisions on new, unseen data.

Machine learning algorithms typically involve training a model on a large dataset, and then evaluating its performance on a separate set of data. The model is then refined and improved based on feedback from this evaluation process [7].

Some common types of machine learning algorithms include supervised learning, unsupervised learning, and reinforcement learning. In supervised learning, the model is trained on a labeled dataset, where each example is associated with a specific output or label. In unsupervised learning, the model is trained on an unlabelled dataset, where the goal is to find patterns and structure in the data. In reinforcement learning, the model learns by interacting with an environment and receiving rewards or penalties based on its actions.

Machine learning has a wide range of applications, from image and speech recognition to natural language processing, recommender systems, and autonomous vehicles. It is a rapidly growing field that is changing the way we approach many problems and is expected to have a significant impact on the future of technology and society [10].

2.3 Categories of machine learning: -

There are generally three categories of machine learning: supervised learning, unsupervised learning, and reinforcement learning.

2.3.1 Supervised learning:

Supervised learning is a type of machine learning in which the model is trained on a labelled dataset, where each example is associated with a specific output or label. The goal of supervised learning is to learn a function that can predict the output for new, unseen inputs.

Examples of supervised learning include image classification, speech recognition, and natural language processing[11].

Two type of supervised learning i) Classification ii) Regression

- Classification:
  The goal is Predict a discreate label or category for a given input. The input is typically a set of features, and the output is a class label from a predefined set of possible labels.
  Example: - A model maybe trained to classify emails as either spam or not spam , based on the words used in the email.

- Regression:

  The goal is to predict a continuous numerical value for a given input. The input is typically a set of features, and the output is a numeric value.

  Example: -A model may be trained to predict the price of a house based on features such as its location, size, and number of rooms.

Types: -

- Regression
- Logistic Regression
- Classification
- Naive Bayes Classifiers
- K-NN (k nearest neighbours)
- Decision Trees
- Support Vector Machine

2.3.2 Unsupervised learning:

This model is trained on an unlabelled dataset, where the goal is to find patterns and structure in the data. Unlike supervised learning, there is no correct output for each input, and the model must learn to identify patterns and relationships on its own [11].

Examples of unsupervised learning include clustering, dimensionality reduction, and anomaly detection.

- Clustering:
  Clustering is a type of unsupervised learning in which the goal is to group similar data

points together based on their features. The algorithm analyses the data and identifies patterns, grouping similar data points into clusters. Clustering can be used for various applications such as customer segmentation, image segmentation, and anomaly detection.
Example: -
An example of clustering could be grouping customers into different segments based on their purchasing behaviour.

- Association rule learning:
  Association rule learning is a type of unsupervised learning that is used to find patterns and relationships in data [12]. It involves discovering co-occurrences or relationships among variables in large databases. The goal is to identify the patterns that occur frequently in the data, such as items that are frequently purchased together.
  Example: -
  This technique is often used in market basket analysis, where the goal is to find associations among items frequently purchased together in a store.

Clustering Types: -

1. Hierarchical clustering
2. K-means clustering
3. Density-based clustering
4. Subspace clustering
5. Fuzzy clustering

Reinforcement Learning:

Reinforcement learning is a type of machine learning in which the model learns by interacting with an environment and receiving rewards or penalties based on its actions. The goal of reinforcement learning is to learn a policy that maximizes the expected cumulative reward over time[13].

Examples of reinforcement learning include game playing, robotics, and autonomous vehicles.

CHAPTER-3

3.Classifiers for human gait classification

3.1 K-nearest neighbors (KNN): -

KNN stands for k-Nearest Neighbours, which is a simple and popular machine learning algorithm used for both classification and regression problems.

In KNN, the basic idea is to find the k closest data points to a given query point, and then use the label or value of those points to make a prediction for the query point. The "k" value is a hyperparameter that specifies the number of nearest neighbours to consider for each prediction[14].

3.1.1The algorithm works as follows:

- **Load the data:** First, load the training dataset into memory, including the features (inputs) and labels (outputs) for each data point.

- **Choose K:** Select the value of K, which specifies the number of nearest neighbours to consider for each prediction.

- **Calculate distances:** For each query point in the test dataset, calculate the distance between the query point and each data point in the training dataset using a distance metric, such as Euclidean distance.

- **Select K-Nearest Neighbours:** Select the K training data points with the smallest distances to the query point.

- **Predict the output:** For classification problems, assign the most common class label among the K nearest neighbours to the query point. For regression problems, calculate the average value of the K nearest neighbours and assign that value to the query point.

- **Evaluate the model:** Evaluate the performance of the model on the test dataset using a performance metric, such as accuracy for classification problems or mean squared error for regression problems.

- **Tune the hyperparameters:** Experiment with different values of K and other hyperparameters, such as the distance metric, to optimize the performance of the model.

- **Use the model for prediction:** Once the model has been trained and tuned, it can be used to make predictions on new, unseen data points.

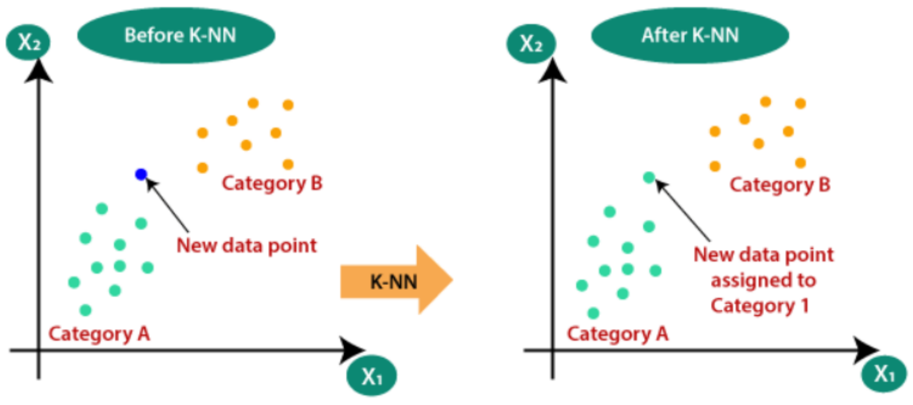


Fig 3.1

From Fig 3.1 Firstly, we will choose the number of neighbours, so we will choose the k value.

Next, we will calculate the Euclidean distance between the data points. The Euclidean distance is the distance between two points, by calculating the Euclidean distance we got the nearest neighbours. As we can see the 3 nearest neighbours are from category A, hence this new data point must belong to category A.

3.2 Support vector machine (SVM): -

SVM stands for Support Vector Machine, which is a powerful and popular machine learning algorithm used for classification and regression analysis. SVM works by finding the optimal hyperplane that separates the data points into different classes or predicts a continuous target variable [3].

In a binary classification problem, SVM finds the hyperplane that maximizes the margin between the two classes. The margin is defined as the distance between the hyperplane and the closest data points from each class, and maximizing it helps to improve the generalization of the model.

For non-linearly separable data, SVM can use a kernel function to transform the input features into a higher-dimensional space where a linear hyperplane can separate the data. Some commonly used kernel functions include linear, polynomial, and radial basis function (RBF) kernels[14].

SVM can also be used for multi-class classification by combining multiple binary classifiers, such as one-vs-one or one-vs-all.

In addition to classification, SVM can also be used for regression analysis by predicting a continuous target variable. In this case, SVM finds the hyperplane that minimizes the errors between the predicted and actual target values.

3.2.1 The algorithm works as follows:

- Load the data: First, load the training dataset into memory, including the features (inputs) and labels (outputs) for each data point.
- Pre-process the data: Pre-process the data as necessary, such as normalizing the features or handling missing values.

- Choose the kernel function: Select the kernel function to use for transforming the input features into a higher-dimensional space, if necessary.
- Choose the regularization parameter: Choose the value of the regularization parameter C, which controls the trade-off between maximizing the margin and minimizing the classification errors.
- Train the SVM model: Use the training dataset to train the SVM model by finding the optimal hyperplane that separates the data points into the two classes. This is done by solving an optimization problem that involves minimizing the objective function subject to constraints.
- Evaluate the model: Evaluate the performance of the SVM model on a separate validation dataset using a performance metric, such as accuracy, precision, recall, or F1 score. Adjust the hyperparameters as necessary to optimize performance.
- Use the model for prediction: Once the model has been trained and validated, it can be used to make predictions on new, unseen data points by applying the learned hyperplane to the input features.

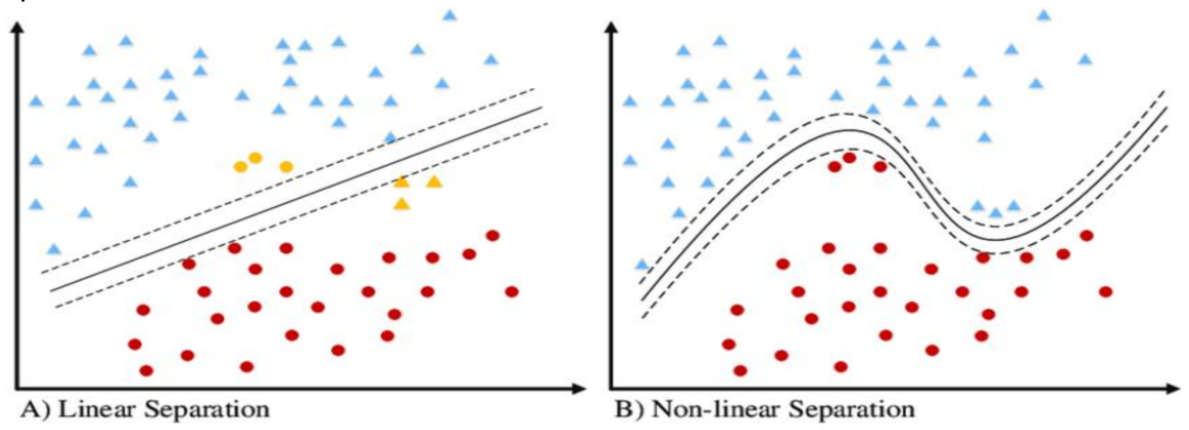


Fig-3.2

A)Linear SVMs are appropriate when the data can be separated by a linear boundary, that is, when the classes are linearly separable[15]. This means that a straight line can be drawn to separate the data into two classes. Linear SVMs are faster and simpler to train than non-linear SVMs, and they perform well when the number of features is large relative to the number of observations.

B) Non-linear SVMs are appropriate when the data cannot be separated by a linear boundary, that is, when the classes are not linearly separable. Non-linear SVMs are used to classify data with complex patterns and relationships [15]. Non-linear SVMs use kernel functions to transform the input data into a higher-dimensional feature space, where the data can be linearly separable. The choice of kernel function depends on the problem and the nature of the data.

3.3 RANDOM FOREST: -

Random Forest is an ensemble learning method that combines multiple decision trees to make a final prediction. Each decision tree is trained on a random subset of the training data, and a random subset of the input features is used for splitting at each node [15]. This process is repeated to build a forest of decision trees. During training, the algorithm creates a large number of decision trees, typically several hundred or thousand, and aggregates their predictions to form a final prediction [16].

Random Forest can be used for both classification and regression problems. For classification, the final prediction is made by taking the mode of the predictions from all the trees in the forest. For regression, the final prediction is made by taking the mean of the predictions from all the trees in the forest [13].

The Random Forest algorithm has several hyperparameters that can be tuned to optimize its performance. These include the number of trees in the forest, the maximum depth of the decision trees, and the size of the random subsets of the training data and input features.

3.3.1The algorithm works as follows:

- **Data preparation:** The first step is to prepare the data for the Random Forest algorithm. This involves cleaning the data, handling missing values, and encoding categorical variables as necessary. The data is then split into training and testing sets.
- **Building the forest:** The Random Forest algorithm builds a forest of decision trees. Each tree is trained on a random subset of the training data and a random subset of the input features. The number of trees in the forest is a hyperparameter that can be tuned to optimize performance.
- **Training the trees:** Each decision tree is trained using the training data and a random subset of the input features. The algorithm splits the data at each node using the best split that maximizes the information gain or Gini index. This process is repeated until the tree reaches its maximum depth or until all the samples at a node belong to the same class.
- **Aggregating the predictions:** Once all the decision trees are trained, the algorithm aggregates their predictions to form a final prediction. For classification problems, the final prediction is made by taking the mode of the predictions from all the trees in the forest. For regression problems, the final prediction is made by taking the mean of the predictions from all the trees in the forest.
- **Evaluating the model:** The performance of the Random Forest model is evaluated using the testing set. The accuracy, precision, recall, F1-score, or other appropriate metrics are used to evaluate the model's performance.
- **Tuning hyperparameters:** The hyperparameters of the Random Forest algorithm, such as the number of trees in the forest, the maximum depth of the decision trees, and the size of the random subsets of the training data and input features, can be tuned to optimize the model's performance.
- **Deploying the model:** Once the model has been trained and evaluated, it can be deployed to make predictions on new data.

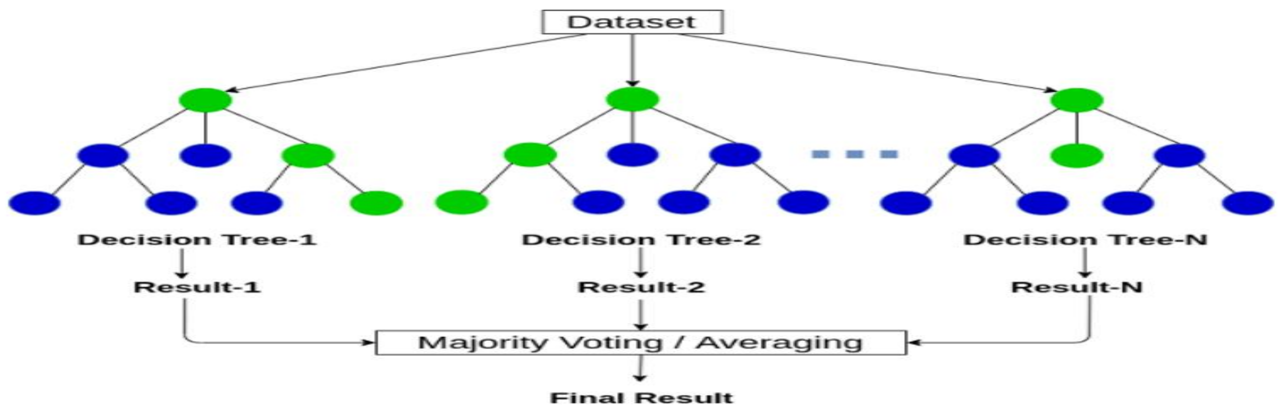
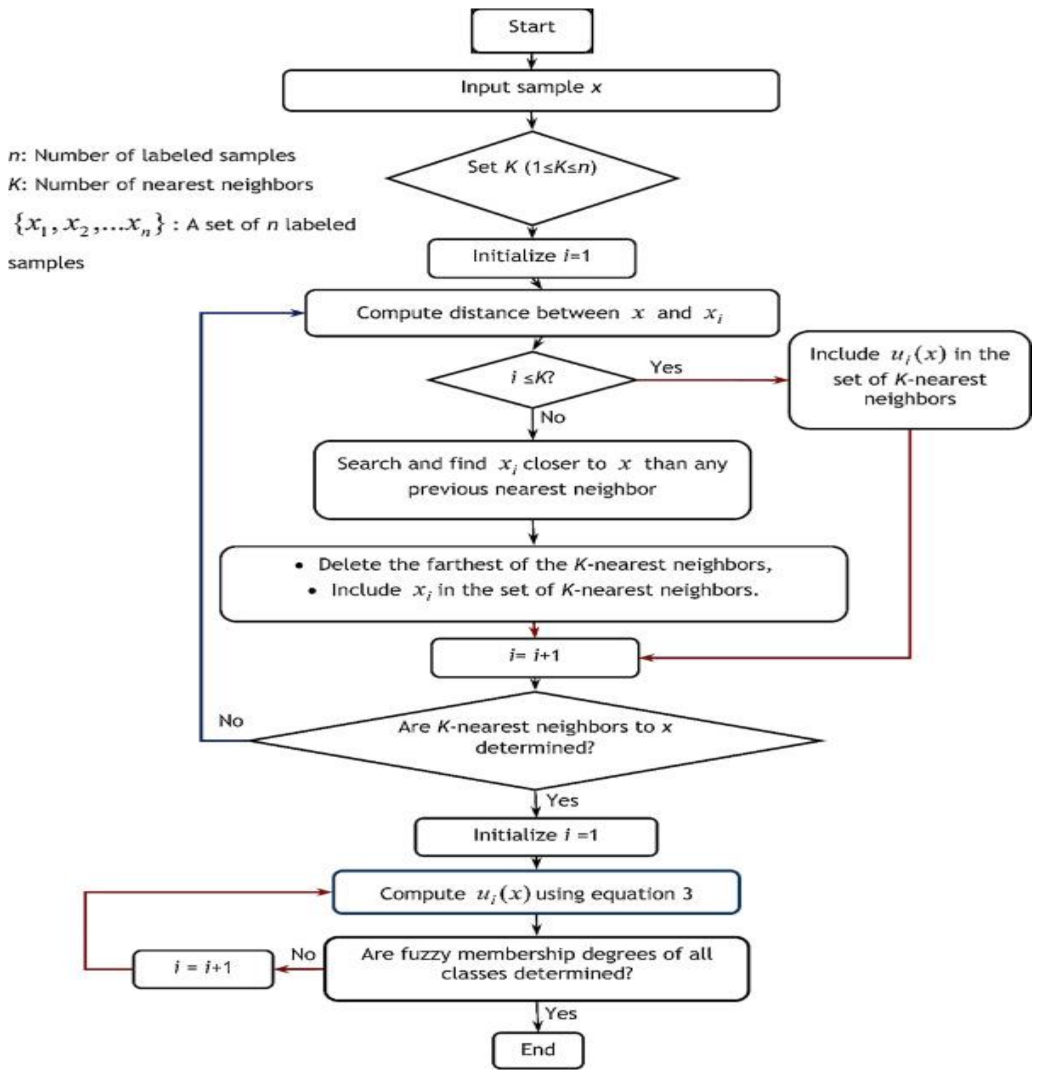


Fig-3.3

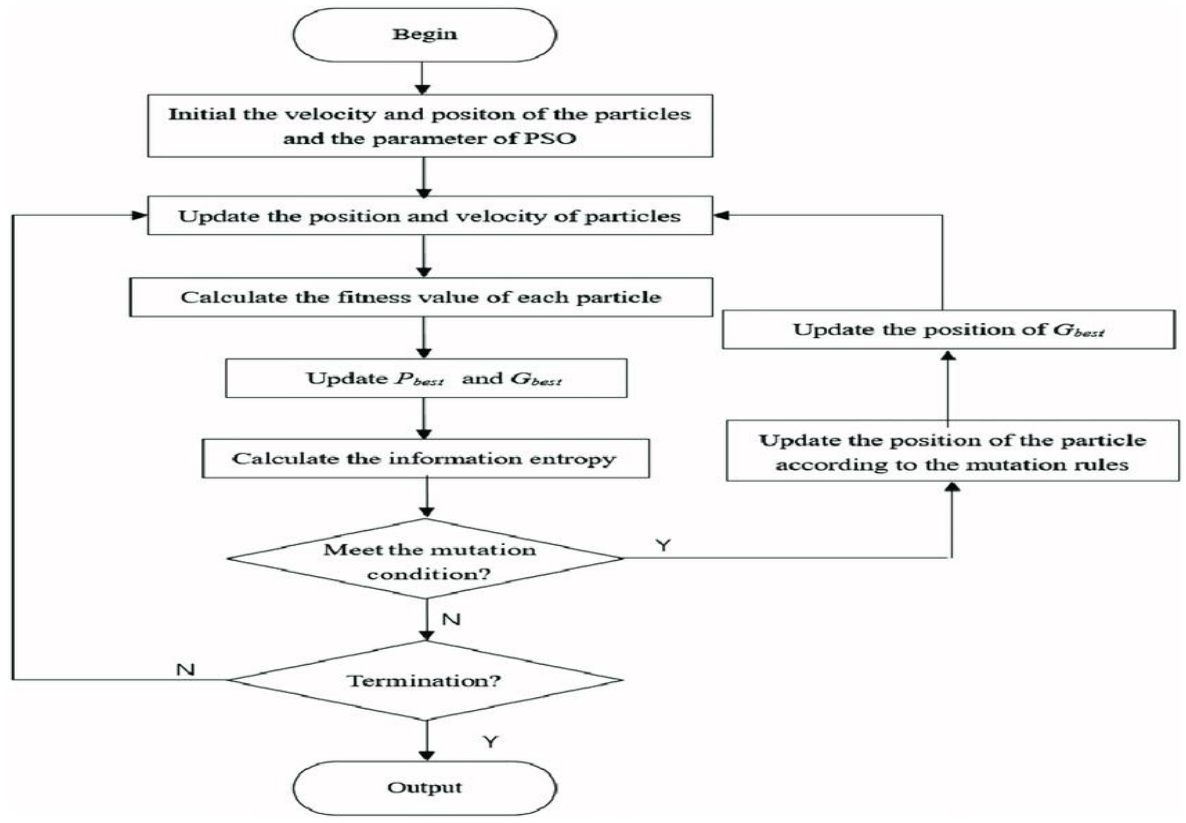
The random forest combines the output of individual decision trees to generate the final output. Bootstrapping is the process of randomly selecting items from the training dataset. This is a haphazard technique. It assembles randomized decisions based on several decisions and makes the final decision based on the majority voting.

3.4. Flowchart

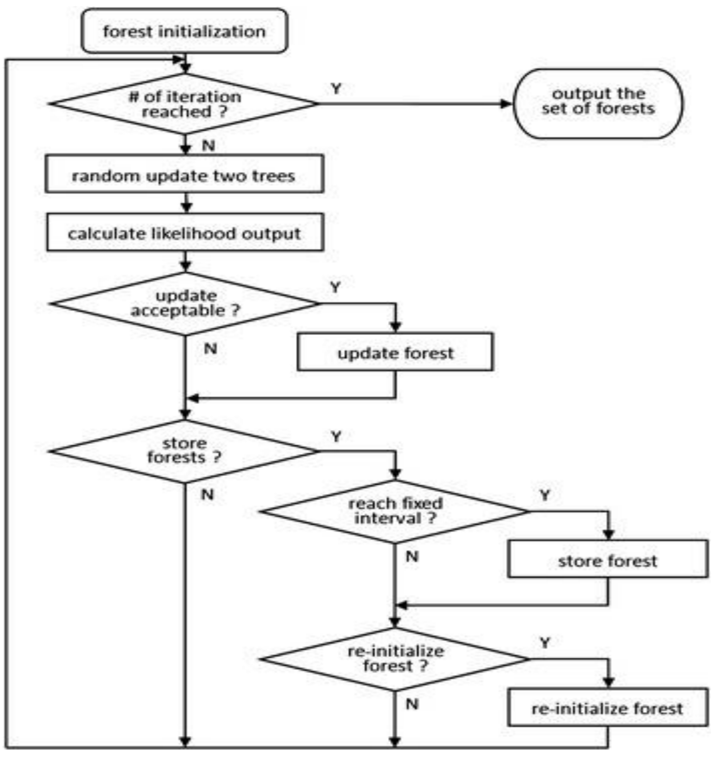
3.4.1 KNN


Start
Input sample x
n: Number of labeled samples
K: Number of nearest neighbors
$\{x_1, x_2, \ldots x_n\}$: A set of n labeled samples
Set K (1≤K≤n)
Initialize i=1
Compute distance between x and x_i
i ≤K?
Yes
Include $u_i(x)$ in the set of K-nearest neighbors
No
Search and find x_i closer to x than any previous nearest neighbor
Delete the farthest of the K-nearest neighbors,
Include x_i in the set of K-nearest neighbors.
i= i+1
No
Are K-nearest neighbors to x determined?
Yes
Initialize i =1
Compute $u_i(x)$ using equation 3
i = i+1
No
Are fuzzy membership degrees of all classes determined?
Yes
End

3.4.2 SVM.


Begin
Initial the velocity and positon of the particles and the parameter of PSO
Update the position and velocity of particles
Calculate the fitness value of each particle
Update P_{best} and G_{best}
Calculate the information entropy
Meet the mutation condition?
Y
Update the position of the particle according to the mutation rules
Update the position of G_{best}
N
Termination?
N
Y
Output

3.4.3 Random forest:-



CHAPTER-4

RESULT AND ANALYSIS

4.1 Introduction

Human gait classification is an important application area of machine learning, where algorithms are used to analyses gait data and classify different walking patterns. Machine learning algorithms can help identify the key features that differentiate different gait patterns and enable automated classification of large datasets.

Gait classification using machine learning has various applications, including the identification of gait abnormalities caused by neurological or musculoskeletal disorders, the assessment of sports performance, and the design of prosthetic devices [17]. Machine learning can also be used to analyze gait patterns in real-time, enabling early detection of gait impairments and personalized treatment plans.

Machine learning techniques can be applied to various gait parameters, including step length, stride duration, walking speed, and joint angles, to identify the patterns that distinguish one gait pattern from another. The algorithms can also learn from the data and improve their accuracy over time, as they analyse more data and incorporate feedback from human experts.

Recent advances in machine learning, such as deep learning, have enabled researchers to analyze and classify gait data with greater accuracy and efficiency than ever before[12]. These advances have the potential to revolutionize the way we diagnose and treat gait-related conditions and improve our understanding of human movement.

To compare the performance of 3 different classifiers

- Support vector machine and
- K-nearest neighbours
- Random forest.

We have experimented with human gait data set .Each experiment was executed for 10 times iterations and there accuracy, mean and stander deviation,precision,recall,F1 are noted.

ACCURACY: -

Accuracy is the percentage of correct classifications that a trained machine learning model achieves, i.e., the number of correct predictions divided by the total number of predictions across all classes. The number of correct classifications predictions divided by the total number of predictions[18]. That is: Accuracy= correct predictions + incorrect predictions
The accuracy formula for binary classification is: (TP+TN)/(TP+TN+FP+FN)
Were

- TP is the number of **true positives** (correct predictions).
- TN is the number of **true negatives** (correct predictions).
- FP is the number of **false positives** (incorrect predictions).
- FN is the number of **false negatives** (incorrect predictions).

Precision: -

In machine learning, precision is a measure of the accuracy of a classification model. It is the fraction of true positive predictions (i.e., the number of correct positive predictions) out of all positive predictions made by the model[18]. In other words, precision measures the proportion of positive predictions that were actually correct.

The formula for precision is:

precision = true positives / (true positives + false positives)

where true positives are the number of correctly classified positive examples, and false positives are the number of negative examples that were classified as positive by the model.

Recall: -

In machine learning, recall is a measure of the completeness of a classification model. It is the fraction of true positive predictions (i.e., the number of correct positive predictions) out of all actual positive examples in the dataset[18]. In other words, recall measures the proportion of actual positive examples that were correctly identified by the model.

The formula for recall is:

recall = true positives / (true positives + false negatives)

where true positives are the number of correctly classified positive examples, and false negatives are the number of positive examples that were incorrectly classified as negative by the model.

F1: -

In machine learning, F1 score is a harmonic mean of precision and recall. It is a way to balance precision and recall, and is often used as a summary statistic for classification models. The F1 score takes into account both false positives and false negatives, and provides a more balanced evaluation of a model's performance.

The formula for F1 score is:

F1 = 2 * (precision * recall) / (precision + recall)

where precision is the fraction of true positive predictions out of all positive predictions made by the model, and recall is the fraction of true positive predictions out of all actual positive examples in the dataset[19].

4.2 Description of Data Set:-

This is a Human Gati Data set. I get this Data set from University of California Irvine machine learning repository and the website is [https://archive.ics.uci.edu/ml/machine-learning-databases/00544/][7]

In my data set there are 17 columns and 2111 rows. All the features of my data set are['Gender', 'Age', 'Height', 'Weight', 'family_history_with_overweight', 'FAVC', 'FCVC', 'NCP', 'CAEC', 'SMOKE', 'CH2O', 'SCC', 'FAF', 'TUE', 'CALC', 'MTRANS', 'NObeyesdad']

- **Gender**-It is defined about the gender of a Human.
- **Age**-It is defined how older that human.
- **Height-** It is defined how much tall that man.
- **Weight**- It's defined is a man Normal weight or over weighted.
- **Family_history_with_overweight**- It is defined according to weight that human is Normal weight or over weighted.
- **TUE-** The "Time under exposure to sunlight" feature in your data set likely refers to the amount of time that an individual spends outdoors in direct sunlight.
- **FAVC-** FAVC stands for "Freezing of gait in Parkinson's disease", which is a motor symptom that affects many Parkinson's disease patients. It's defined where the person feels as though their feet are stuck to the ground and cannot move forward.
- **FCVC-** FCVC stands for Foot Contact-Velocity Curve, the FCVC feature has been used in human gait classification to differentiate between different types of gait, such as normal walking, limping, and running.
- **NCP-**It is stand for Normalized Cumulative Power. NCP is calculated by integrating the power output of each limb over a complete gait cycle and normalizing it to the person's body weight. This provides a way to compare the energy expenditure of different individuals, regardless of their size or weight.
- **CAEC (Consumption of alcohol, exercise, and caffeine)-** These feature is used to classify gait patterns based on the presence or absence of certain factors. For example, individuals who consume alcohol regularly may have a distinct gait pattern compared to those who do not. Similarly, individuals who exercise regularly may have a more efficient gait pattern than those who do not exercise.
- **SMOKE-** The "smoke" feature in my data set likely refers to whether or not an individual is a smoker.
- **CH2O-** The "Formaldehyde concentration in air" feature is used as a variable to determine if there is a correlation between exposure to formaldehyde gas and differences in gait patterns.
- **SCC-** The "sodium concentration in blood" feature may be used as a variable to help identify patterns in gait that are associated with changes in sodium levels.
- **FAF-** The "Physical activity frequency" feature in my data set refers to the frequency or regularity with which an individual engages in physical activity.
- **CALC-** "CALC" feature in my data set refer to monitoring calorie intake. Calorie intake monitoring involves tracking the amount of calories consumed through food and beverages throughout the day.
- **MTRANS-** The "MTRANS" feature in your data set likely refers to the transportation method that the individuals used to get to the testing facility or lab where the gait data was collected. The transportation method could be an indicator of the individuals' overall physical activity level, as some modes of transportation (such as walking or cycling) require more physical activity than others (such as driving or taking public transportation).
- **There are Seven Different class in my data set and all the class are**
  ['Normal_Weight', 'Overweight_Level_I', 'Overweight_Level_II',
  'Obesity_Type_I' ,'Insufficient_Weight', 'Obesity_Type_II',
  'Obesity_Type_III']

4.3 Parameter consider : -

K-Nearest Neighbourhood (KNN): -

K Value

‘k’ in KNN is a parameter that refers to the number of nearest neighbours to include in the majority of the voting process.

Classifier	Parameter	Value
KNN	K Value	25

Table-4.1

SVM: -

Kernel
It is a method used to take data as input and transform it into the required form of processing data. “Kernel” is used due to a set of mathematical functions used in Support Vector Machine providing the window to manipulate the data. So, Kernel Function generally transforms the training set of data so that a non-linear decision surface is able to transform to a linear equation in a higher number of dimension spaces. Basically,[20] It returns the inner product between two points in a standard feature dimension.

Standard Kernel Function Equation:

K (\bar{x}) = 1, if ||\bar{x}|| <= 1

K (\bar{x}) = 0, Otherwise

Major Kernel Functions:-

- Gaussian Kernel**:** It is used to perform transformation when there is no prior knowledge about data.
- Gaussian Kernel Radial Basis Function (RBF): Same as above kernel function, adding radial basis method to improve the transformation[6].
- Sigmoid Kernel**:** this function is equivalent to a two-layer, perceptron model of the neural network, which is used as an activation function for artificial *Neurons.*
- Polynomial Kernel**:** It represents the similarity of vectors in the training set of data in a feature space over polynomials of the original variables used in the kernel.
- Linear Kernel**:** used when data is linearly separable.

- **Gamma:** This parameter decides how far the influence of a single training example reaches during transformation, which in turn affects how tightly the decision boundaries end up surrounding points in the input space. If there is a small value of gamma, points farther apart are considered similar. So more points are grouped together and have smoother decision boundaries (maybe less accurate)[9]. Larger values of gamma cause points to be closer together (may cause overfitting).
- **The ‘C’ parameter:** This parameter controls the amount of regularization applied to the data. Large values of C mean low regularization which in turn causes the training data to fit very well (may cause overfitting). Lower values of C mean higher regularization which causes the model to be more tolerant of errors (may lead to lower accuracy).

Classifier	Parameter	Value
SVM	Kernel	Linear,RBF
SVM	C	0.1,1,10,100,1000
SVM	Gamma	1,0.1,0.01,0.001,0.0001

Table-4.2

Random Forest

n_estimators: We know that a random forest is nothing but a group of many decision trees, the n_estimator parameter controls the number of trees inside the classifier. We may think that using many trees to fit a model will help us to get a more generalized result, but this is not always the case[21]. However, it will not cause any overfitting but can certainly increase the time complexity of the model.

Max_features: Random Forest takes random subsets of features and tries to find the best split. max_features helps to find the number of features to take into account in order to make the best split. It can take four values “auto”, “sqrt “, “log2” and None.

- In case of auto: considers max_features = sqrt(n_features)
- In case of sqrt: considers max_features = sqrt(n_features), it is same as auto
- In case of log2: considers max_features = log2(n_features)
- In case of None: considers max_features = n_features

max_depth: It indicate the maximum height up to which the trees inside the forest can grow. It is one of the most important hyperparameters when it comes to increasing the accuracy of the model, as we increase the depth of the tree the model accuracy increases up to a certain limit but then it will start to decrease gradually because of overfitting in the model. It is important to set its value appropriately to avoid overfitting [21]. The default value is set to None, none specifies that the nodes inside the tree will continue to grow until all leaves become.

max_leaf_nodes: It sets a limit on the splitting of the node and thus helps to reduce the depth of the tree, and effectively helps in reducing overfitting. If the value is set to None, the tree continues to grow infinitely.

Classifier	Parameter	Value
Random Forest	n_estimators	20,50,100,150
Random Forest	max_features	‘sqrt’, ‘log2’, ‘none’
Random Forest	max_depth	3,6,9
Random Forest	max_leaf_nodes	3,6,9

Table-4.3

Table 4.1 to 4.3 all are define the useable parameter which i used for three classifiers.

4.4 Performance Metrices: -

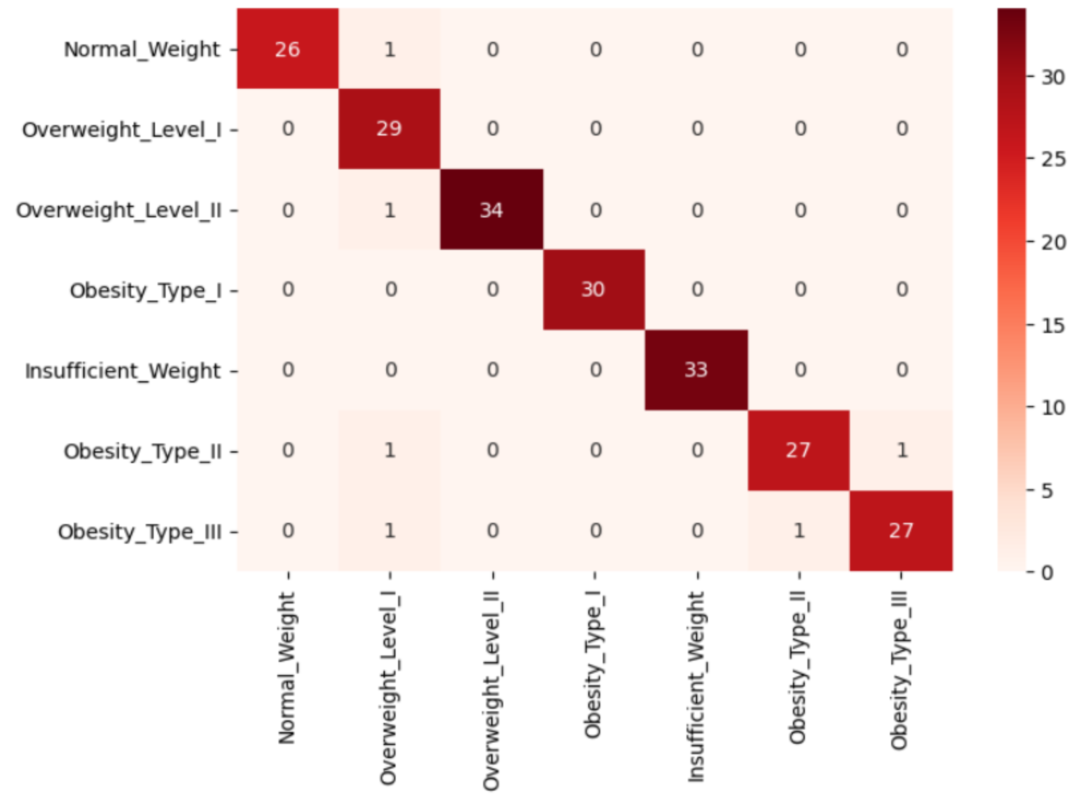


Fig-4.4

Fig 4.4 is the best confusion matrices from all confusion matrices we can observed there are 7 classes .All the class that are in X axis they are actual classes and in Y axis they are in predicated class.

- **One normal_weight person is predicting as over_weight_level_I**
- **One Over_weight_level_II is predicting as over_weight_level_I**
- **One obesity_type_II is predicting as Over_weight_level_II as well as obesity_type_II.**
- **obesity_type_III is predicting as Over_weight_level_I as well as obesity_type_II**

Result:

Training-testing split	Accuracy	Random state	Mean	Standard deviation	Precision	Recall	F1 score	Training Time
90-10	**0.9726**	**65**	**0.94**	**0.5**	**0.97**	**0.97**	**0.97**	**0.96**

Table-4.5

4.5 RESULT AND ANALYSIS

CLASSIFICATION USED: -Here we used three classifications:

1. k-nearest neighbour
2. Support Vector Machine
3. Random Forest

4.5.1 K- NEAREST NEIGHBOUR

Training-testing split	Accuracy	Random state	Mean	Standard deviation	Precision	Recall	F1 score
60-40	**0.72**	**35**	**0.71**	**0.005**	**0.73**	**0.72**	**0.69**
80-20	**0.76**	**90**	**0.73**	**0.011**	**0.77**	**0.76**	**0.74**
90-10	**0.78**	**90**	**0.74**	**0.018**	**0.79**	**0.77**	**0.75**

Table-4.6

All datasets in table-4.6 are split into 60:40 where 60% of data were used for training purposes while the remaining 40% were used for testing purposes and same as for 80:20 and 90:10.

As per results we observe that among the three type of splitting, the ratio 90:10 gives the best results for mean-accuracy , std, precision, recall , F1-score after 10 times iteration .

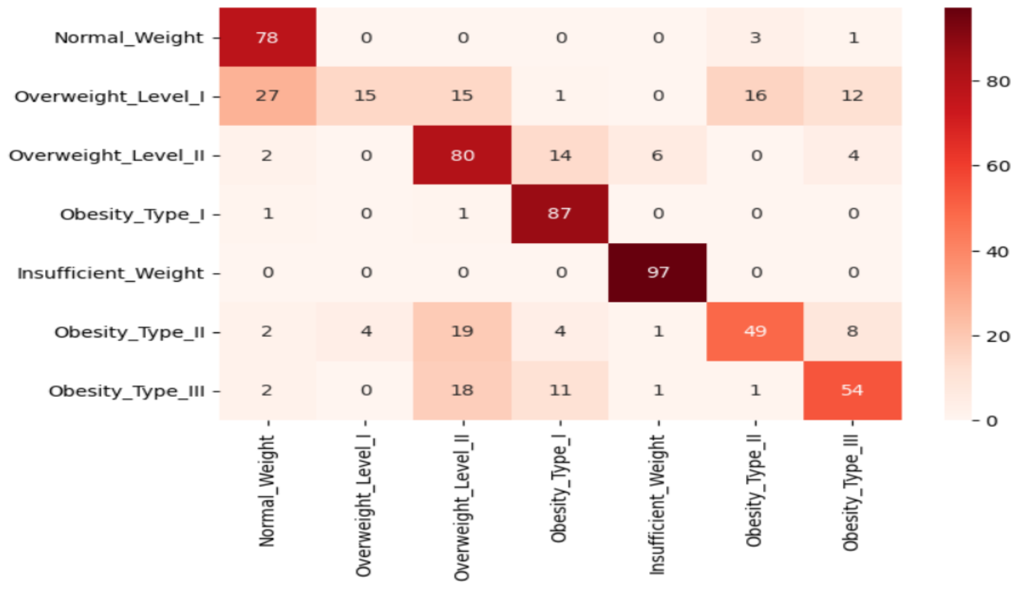


Fig-4.7

Confusion matrix for best estimator in knn of 60-40 train test (Fig-4.7)

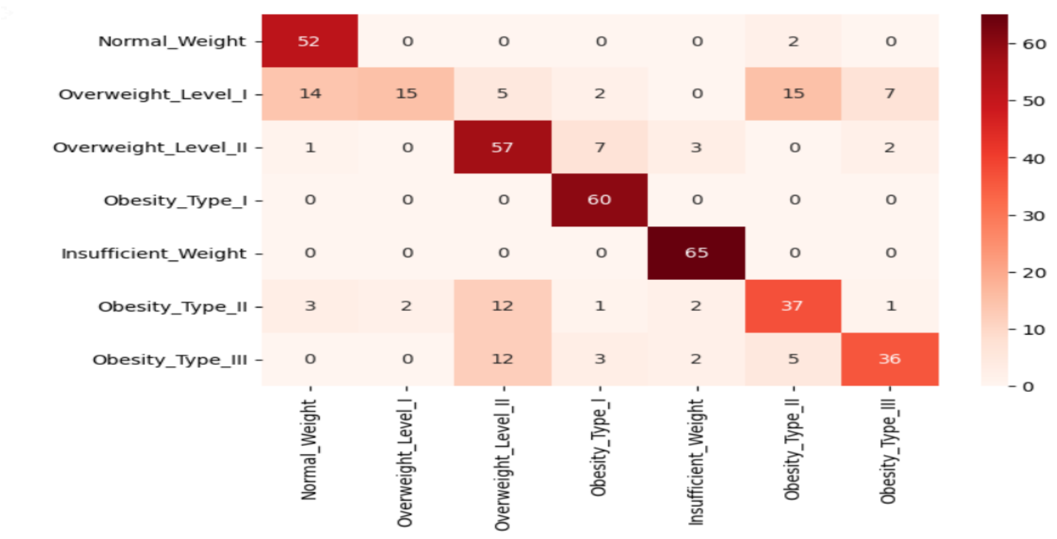


Fig-4.8

Confusion matrix for best estimator in knn of 80-20 train test (Fig-4.8)

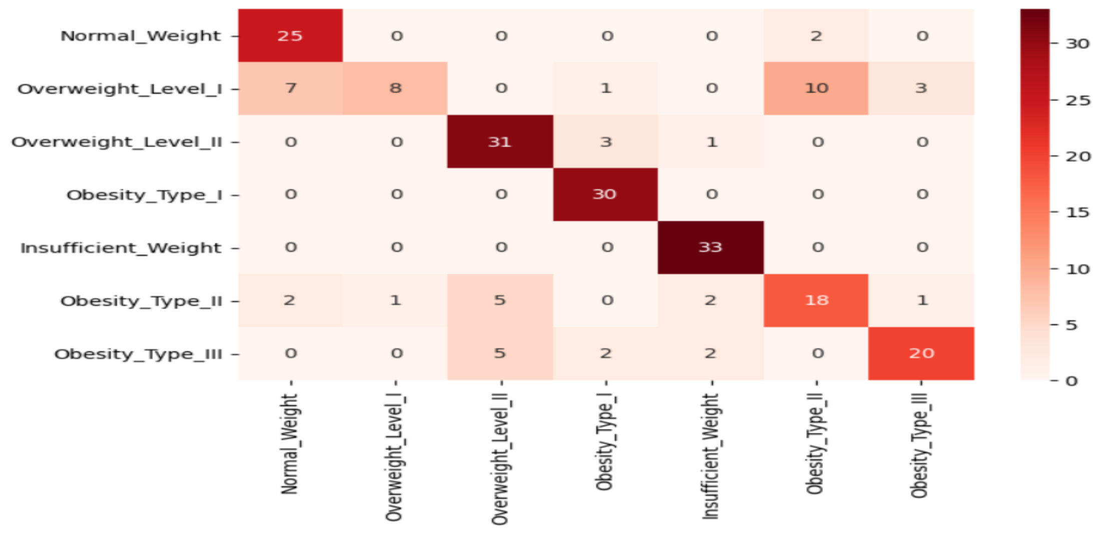


Fig-4.9

Confusion matrix for best estimator in knn of 90-10 train test (Fig-4.9)

4.5.2 SUPPORT VECTOR MACHINE

Training-testing split	Accuracy	Random state	Mean	Standard deviation	Precision	Recall	F1 score	Training time
60-40	**0.950**	**20**	**0.93**	**0.008**	**0.95**	**0.95**	**0.95**	**0.07**
80-20	**0.959**	**90**	**0.95**	**0.007**	**0.969**	**0.968**	**0.96**	**0.17**
90-10	**0.96**	**70**	**0.96**	**0.009**	**0.96**	**0.96**	**0.95**	**0.10**

Table-4.10

From Table-4.10 all datasets are split into 60:40 where 60% of data were used for training purposes while the remaining 40% were used for testing purposes and same as for 80:20 and 90:10.

As per results we observe that among the three type of splitting, the ratio 90:10 gives the best results for mean-accuracy , std, precision, recall , F1-score after 10 times iteration .

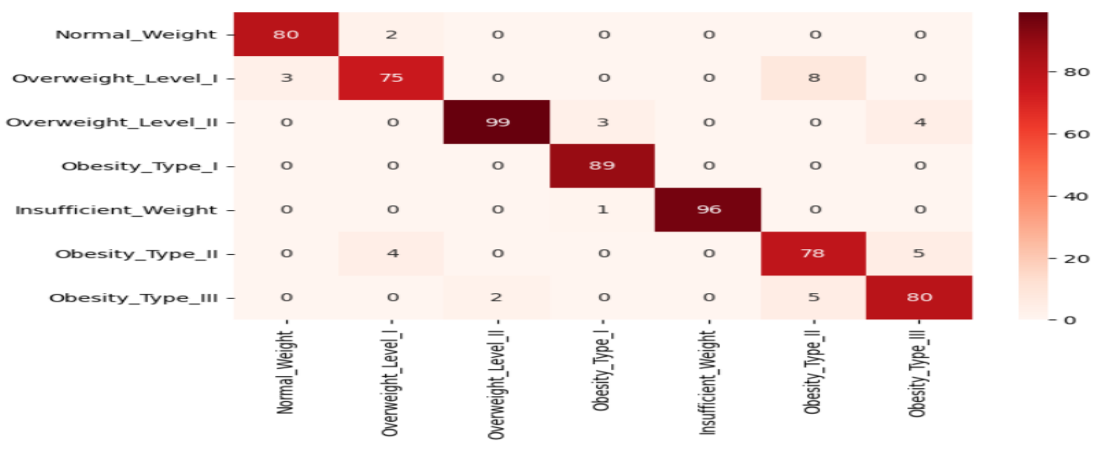


Fig-4.11

Confusion Matrix for Best Estimator in Support Vector Machine of 60-40 train test (Fig-4.11)

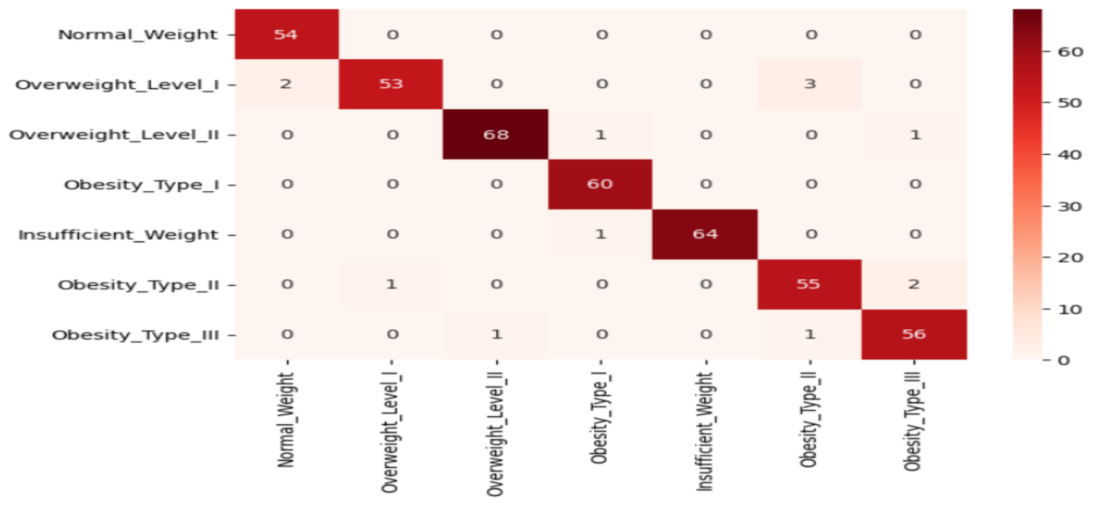


Fig-4.12

Confusion Matrix for Best Estimator in Support Vector Machine of 80-20 train test (Fig-4.12)

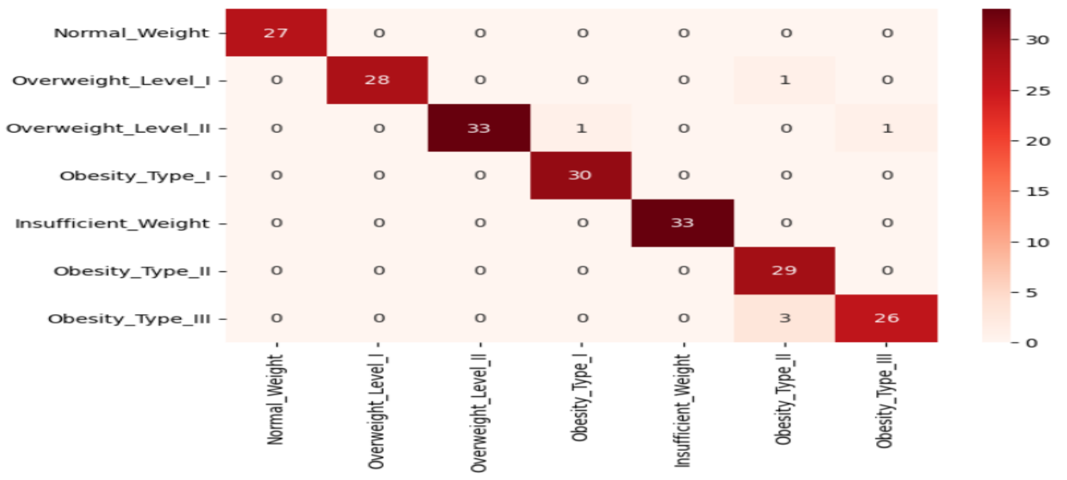


Fig-4.13

Confusion Matrix for Best Estimator in Support Vector Machine of 90-10 train test (Fig-4.13)

4.5.3 RANDOM FOREST

Training-testing split	Accuracy	Random state	Mean	Standard deviation	Precision	Recall	F1 score	Training time
60-40	**0.95**	**50**	**0.93**	**0.007**	**0.94**	**0.94**	**0.94**	**1.32**
80-20	**0.97**	**90**	**0.94**	**0.01**	**0.97**	**0.97**	**0.97**	**0.36**
90-10	**0.9726**	**65**	**0.94**	**0.5**	**0.97**	**0.97**	**0.97**	**0.96**

Table-4.14

In the Table-4.14, all datasets are split into 60:40 where 60% of data were used for training purposes while the remaining 40% were used for testing purposes and same as for 80:20 and 90:10.

As per results we observe that among the three type of splitting, the ratio 90:10 gives the best results for mean-accuracy , std, precision, recall , F1-score after 10 times iteration .

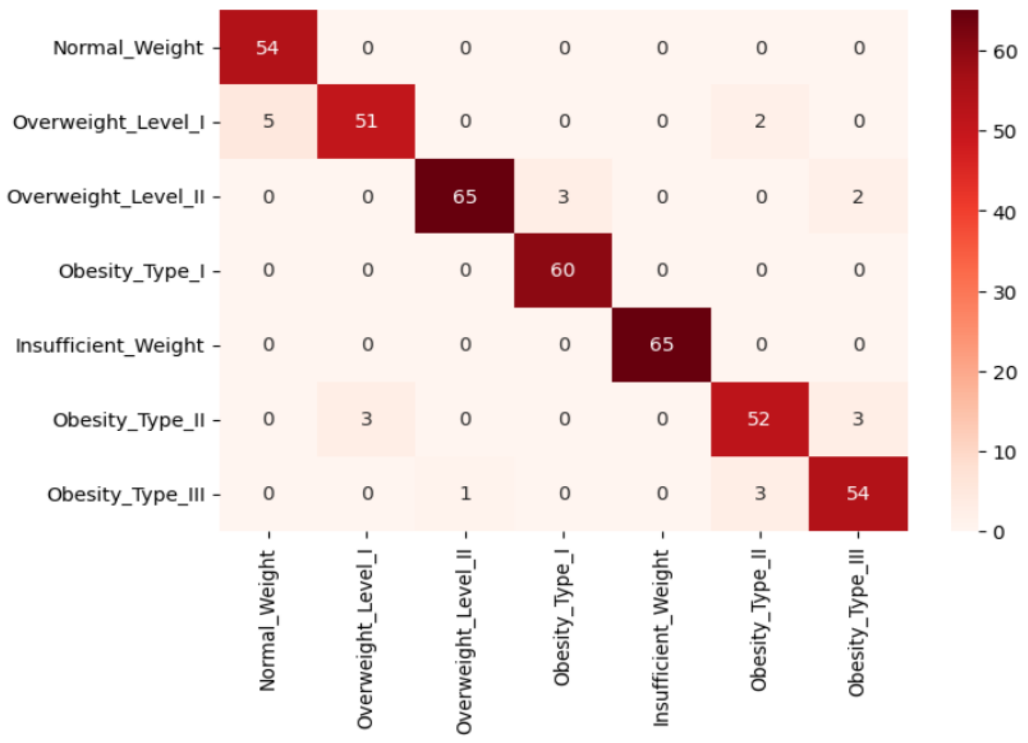


Fig-4.15

Confusion Matrix for Best Estimator in Random Forest of 60-40 train test (Fig-4.15)

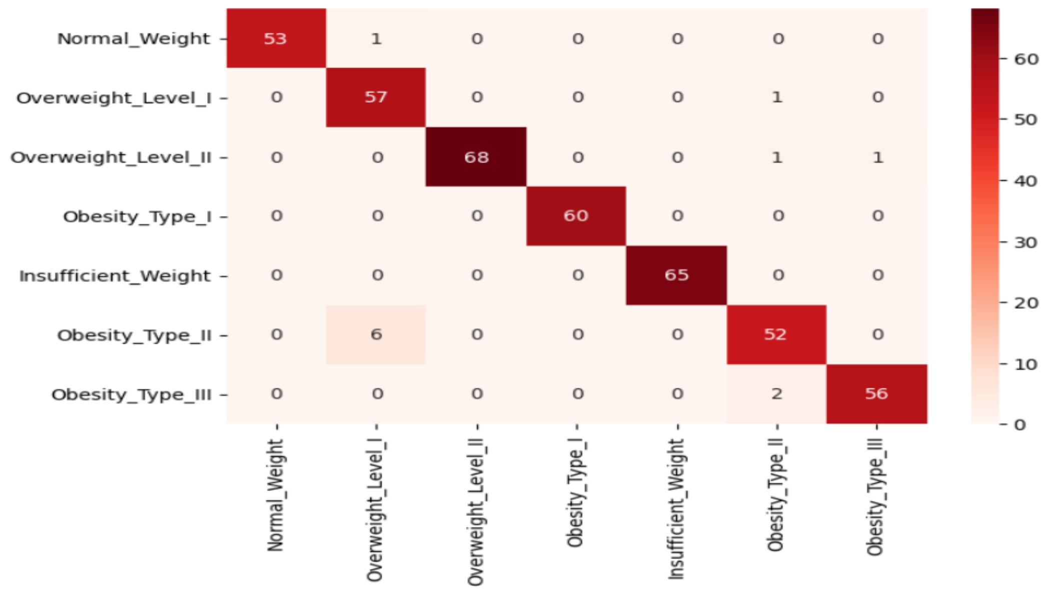


Fig-4.16

Confusion Matrix for Best Estimator in Random Forest of 80-20 train test (Fig-4.16)

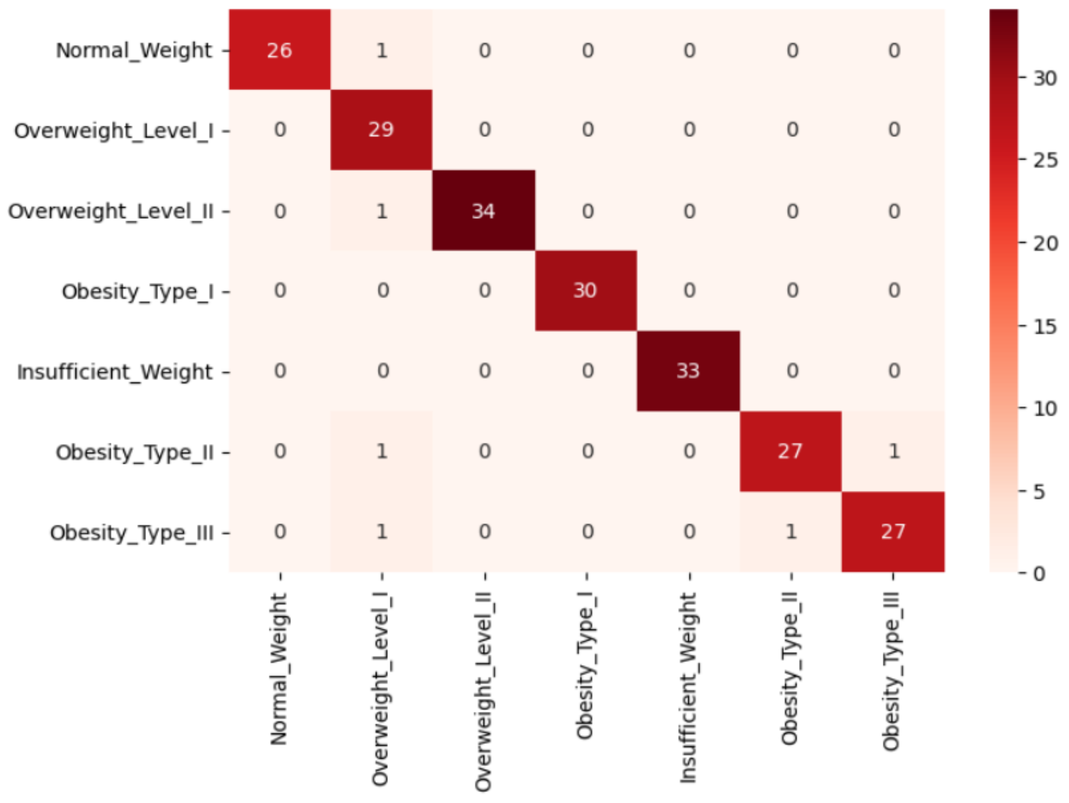


Fig-4.17

Confusion Matrix for Best Estimator in Random Forest of 90-10 train test (Fig-4.17)

4.6 Accuracy graph:

KNN (60-40)

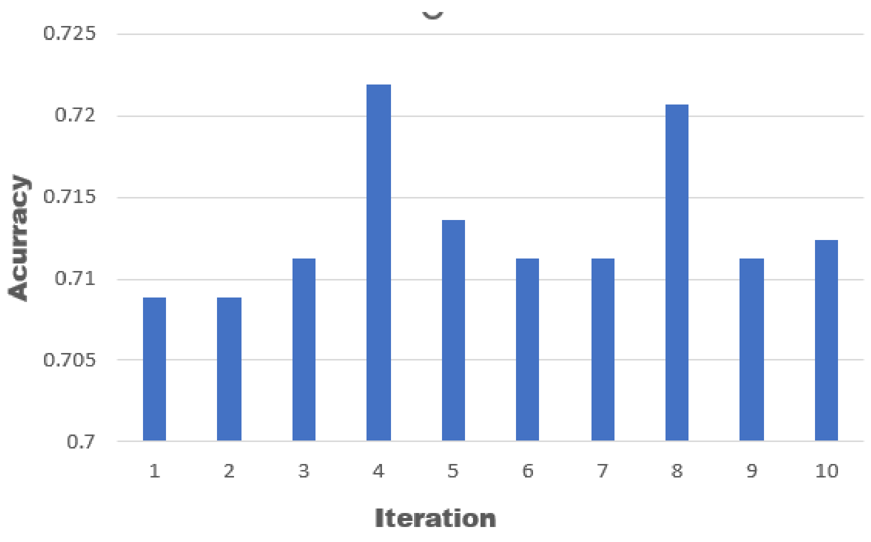


Fig-4.18

KNN (80-20)

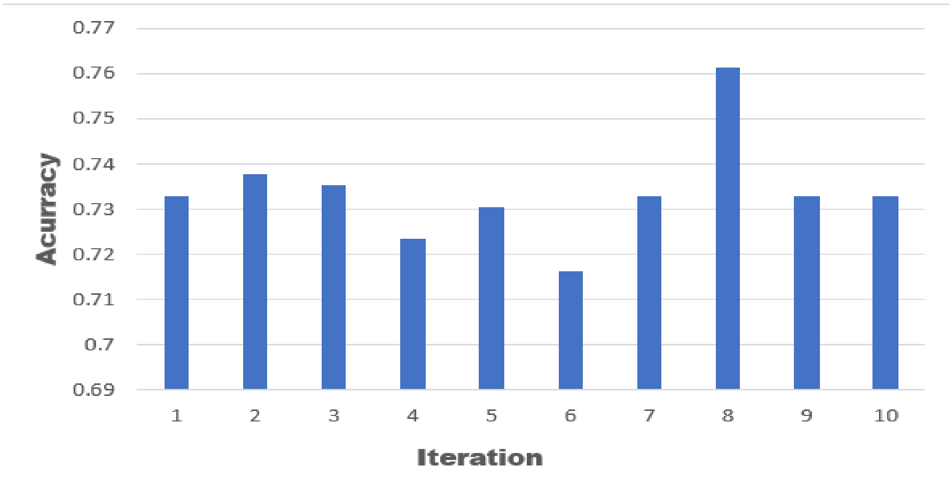


Fig-4.19

KNN (90-10)

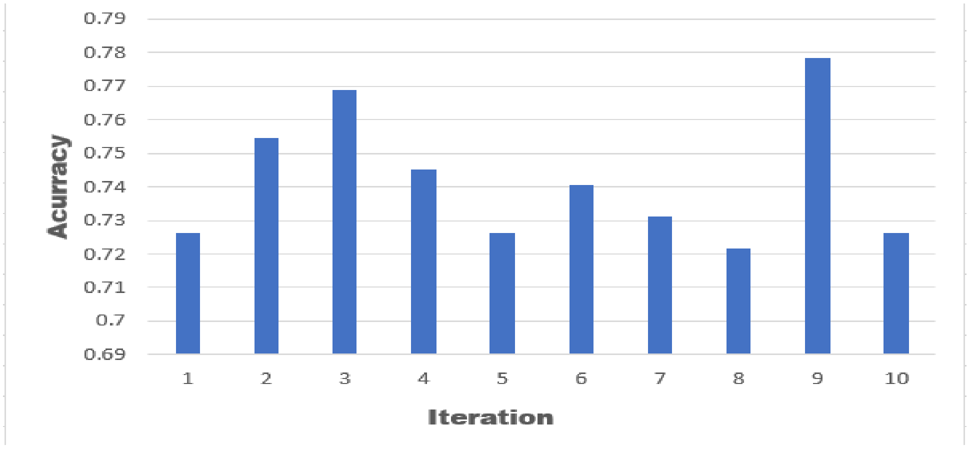


Fig-4.20

SVM (60-40)

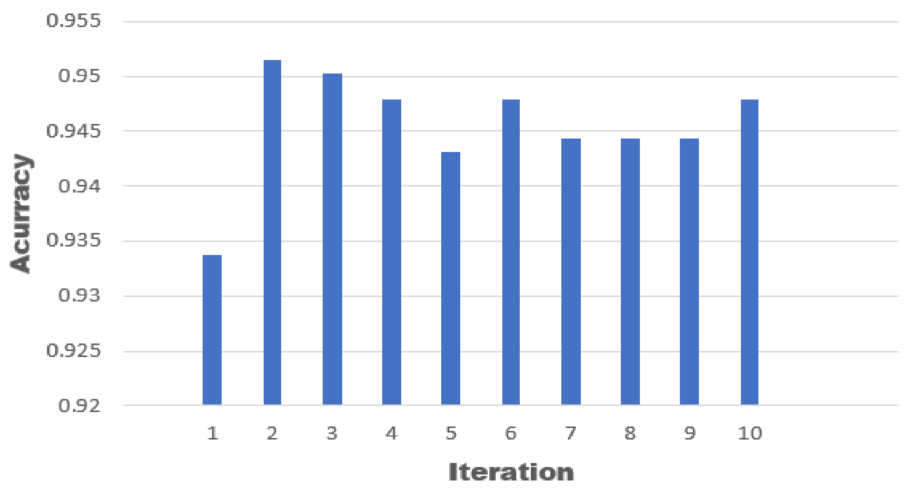
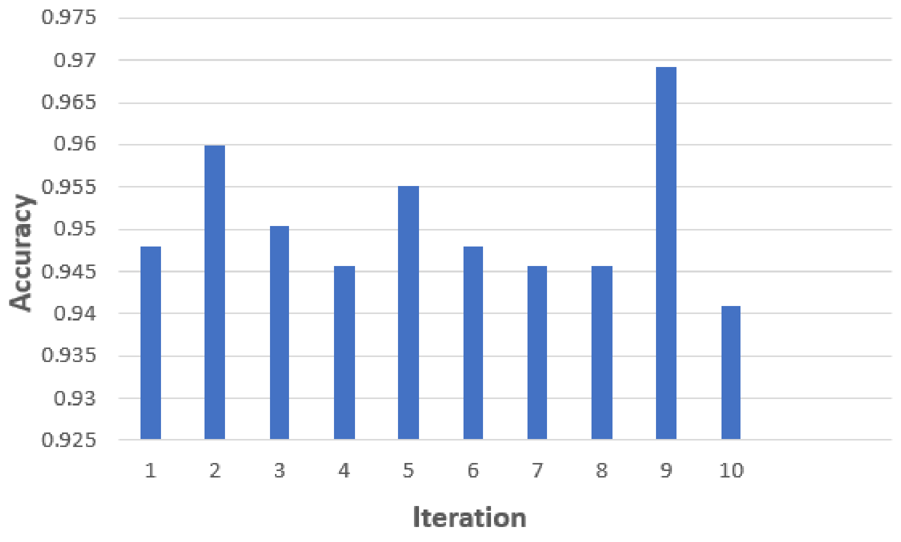


Fig-4.21

SVM (80-20)



Fi g-4.22

SVM (90-10)

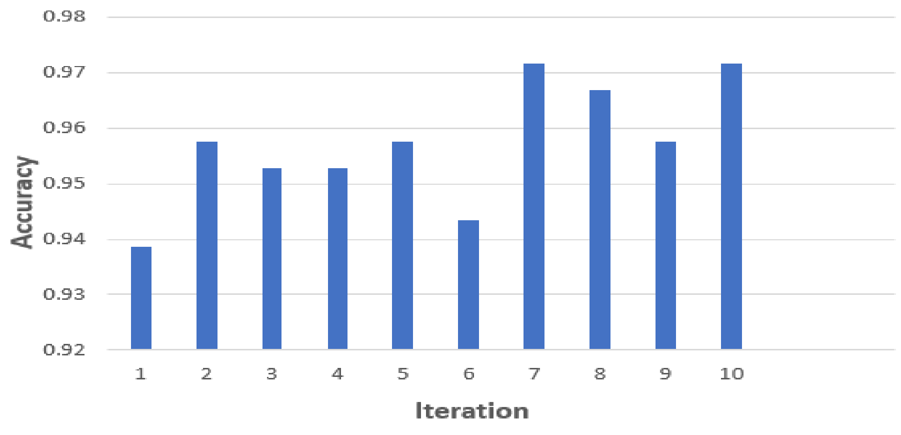


Fig-4.23

Random forest (60-40)

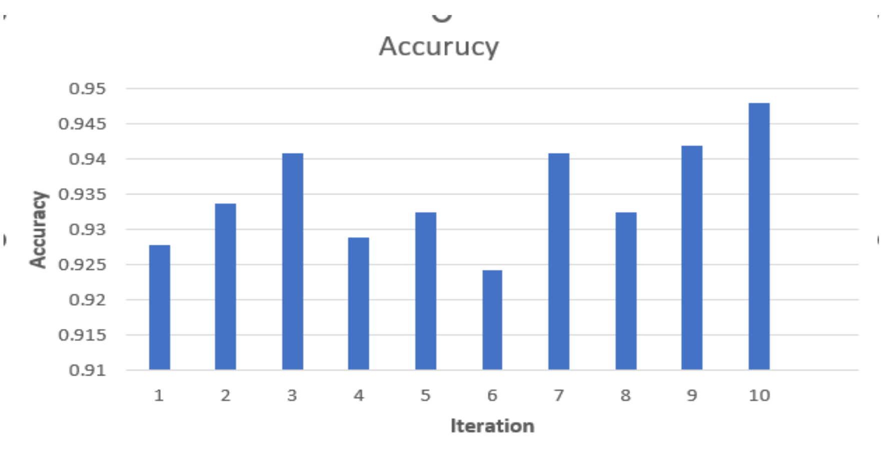


Fig-4.24

Random forest (80-20)

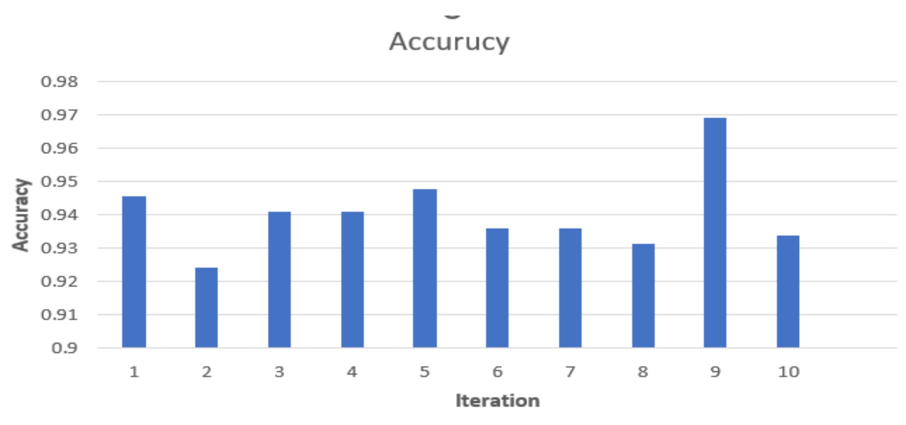


Fig-4.25

Random forest (90-10)

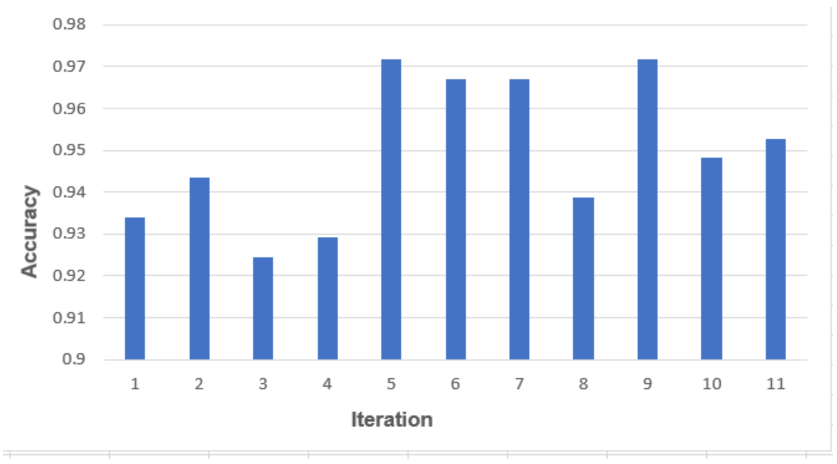


Fig-4.26

All the graphs (Fig 4.18 to Fig-4.26) are denoting the accuracy carve of all classifier.

Where X axis is denoting the Number of Iteration and Y axis is denoting the actual accuracy

After each iteration. Also We can easily identify which classifier gives the highest accuracy among all those classifier .

For my result random forest (90-10) gives the best accuracy result (fig-4.26) i.e.,0.9726

4.7 Analysis

In terms of Accuracy,

Random forest is giving the best result with an accuracy of 97.26% and this implies that

- The activities can be distinguished best with a straight line with respect to the features
- Since the values of the features are numerical data so different activities have different numeric values for all the features which eventually Random Forest to find the best result.

In terms of Standard Deviation,

Random Forest also gives the greatest value of Standard Deviation of 0.5 and this implies that

- Random Forest is the most vulnerable classifier to predict human gait classifier

  as it tries different combinations of maximum depth and number of trees.

- Some operations in Random Forest such as calculating Impurity or entropy

  value for each node takes much time to complete the training.

K-nearest neighbors is providing least value of Standard Deviation as the

accuracy of this classifier is almost same in every iteration.

Chapter-5

5.Conclusion and Future Direction

5.1 Conclusion :-

Human gait classification is a fascinating and rapidly advancing field that utilizes various techniques, including machine learning algorithms, computer vision, and wearable sensors, to analyses and classify different walking patterns. Gait classification has significant implications in fields such as biomechanics, medical diagnosis, rehabilitation, and sports science[19].

By accurately analysing gait parameters and patterns, gait classification enables the identification of gait abnormalities, monitoring of disease progression, personalized rehabilitation planning, and assessment of sports performance. It provides valuable insights into the mechanics and control of human walking, allowing for early detection and intervention in gait-related disorders.

Machine learning algorithms play a crucial role in gait classification by extracting meaningful features, learning patterns, and improving classification accuracy over time [22]. Deep learning approaches, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), have shown promising results in gait classification tasks.

The integration of wearable sensors and advancements in data collection and analysis techniques have enabled real-time gait analysis and personalized monitoring. This offers opportunities for early detection of gait abnormalities, facilitating timely interventions and personalized treatment plans.

Human gait classification continues to evolve, with ongoing research focused on refining classification algorithms, improving feature extraction methods, and integrating multi-modal data sources[15]. The combination of machine learning techniques and advancements in sensor technology holds great promise for further advancements in gait classification accuracy, reliability, and clinical applicability.

Overall, human gait classification has the potential to significantly impact medical diagnosis, rehabilitation practices, and sports performance analysis. It contributes to our understanding of human movement and provides valuable insights for improving the quality of life for individuals with gait-related conditions.

5.2 Future direction: -

The field of human gait classification is poised for exciting future directions that hold immense potential for advancements in diagnosis, treatment, and overall understanding of human movement. Here are some key future directions [23]:

1)Multi-modal Data Integration: Incorporating multiple data modalities, such as video, wearable sensors, force plates, and electromyography (EMG), can provide a more comprehensive and detailed analysis of gait patterns. Integrating these modalities and leveraging their synergies can enhance the accuracy and reliability of gait classification models.

2)Real-time Monitoring and Feedback: Advancements in wearable sensor technology and signal processing techniques will enable real-time gait monitoring and feedback systems. These systems can provide immediate feedback to individuals during gait training or rehabilitation, facilitating corrections and optimizing their gait patterns in real-time [15].

3)Longitudinal Analysis: Longitudinal studies involving the continuous monitoring of individuals' gait patterns over extended periods can provide valuable insights into gait changes over time. Longitudinal analysis can help identify subtle variations and detect early signs of gait abnormalities, enabling timely intervention and personalized treatment [19].

4)Deep Learning Approaches: Continued advancements in deep learning techniques, such as recurrent neural networks (RNNs), generative adversarial networks (GANs), and attention mechanisms, can further improve the accuracy and interpretability of gait classification models. These approaches have the potential to capture complex temporal dependencies and extract high-level representations from gait data.

5)Explainable AI: As gait classification models become more sophisticated, there is a growing need for interpretability and explain ability. Developing methods that can provide meaningful explanations for the decisions made by gait classification models will enhance trust, acceptance, and clinical adoption of these systems [21].

6)Personalized Treatment and Rehabilitation: Gait classification can enable personalized treatment and rehabilitation programs tailored to individual needs. By identifying specific gait parameters or patterns associated with different conditions, personalized interventions can be designed to optimize outcomes and improve the effectiveness of rehabilitation protocols.

7)Transfer Learning and Generalization: Transfer learning techniques can leverage pre-trained models on large datasets to enhance the performance of gait classification models, especially when data availability is limited[24]. Additionally, developing models that can generalize across different populations, demographics, and environments will further enhance the practical utility of gait classification in diverse settings.

8)Clinical Translation: Bridging the gap between research and clinical practice is crucial for the widespread adoption of gait classification methods. Conducting large-scale clinical studies and validating the effectiveness of gait classification models in real-world healthcare settings will be essential for their integration into clinical workflows and decision-making processes[23].

Chapter-6

6.References

1.O'Reilly, M. A., Whelan, D., & Ward, T. E. (2016). Gait classification using time-series shapelets. IEEE transactions on biomedical engineering, 64(5), 1062-1072.

2. El-Hajj, R., Itani, S., Harb, A., Karam, G., & Tamim, H. (2018). Gait classification in individuals with Parkinson's disease using artificial intelligence techniques: A systematic review. Parkinson's Disease, 2018, 6786453.

3. Ganea, R. L., Paraschiv-Ionescu, A., Büla, C., & Aminian, K. (2018). Automatic recognition of gait patterns using kinematic parameters and machine learning algorithms. Journal of neuroengineering and rehabilitation, 15(1), 1-10.

4. Le, Q., & Nguyen, H. T. (2017). Human gait analysis and recognition for control of a robotic exoskeleton. IEEE Transactions on neural systems and rehabilitation engineering, 25(11), 2280-2290.

5. Auvinet, E., Multon, F., St-Arnaud, A., & Rousseau, J. (2002). Posture and gait analysis: A classification tool applied to parkinson's disease. IEEE Transactions on Neural Systems and rehabilitation engineering, 10(4), 430-441.

7. URL: [https://archive.ics.uci.edu/ml/machine-learning-databases/00544/]

8. Hastie, T., Tibshirani, R., & Friedman, J. (2009). The elements of statistical learning: Data mining, inference, and prediction. Springer science & business media.

9. Hausdorff, J. M. (2009). Gait dynamics, fractals and falls: Finding meaning in the stride-to-stride fluctuations of human walking. Human movement science, 26(4), 555-589.

10. Crea, S., Donelan, J. M., & Lichtwark, G. (2017). Gait classification in neurological disease: A review. Journal of neuroengineering and rehabilitation, 14(1), 1-15.

11. Khandelwal, S., Saxena, S., & Tiwari, R. (2021). Gait classification using machine learning techniques: A systematic review. Biomedical signal processing and control, 68, 102648.

12. Lee, S. W., & Park, K. S. (2018). Gait classification of parkinson's disease using machine learning techniques. Sensors, 18(7), 2108.

13. Muro-de-la-Herran, A., García-Zapirain, B., & Méndez-Zorrilla, A. (2014). Gait analysis methods: An overview of wearable and non-wearable systems, highlighting clinical applications. Sensors, 14(2), 3362-3394.

14. Bishop, C. M. (2006). Pattern recognition and machine learning. Springer.

15. Hastie, T., Tibshirani, R., & Friedman, J. (2009). The elements of statistical learning: Data mining, Inference, and prediction. Springer science & business media.

16. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT press.

17. Mitchell, T. M. (1997). Machine learning. McGraw-Hill.

18. Shalev-Shwartz, S., & Ben-David, S. (2014). Understanding machine learning: From theory to algorithms. Cambridge university press.

19. Hua, Y., Jiang, M., Wang, J., & Wang, Z. (2020). Human gait recognition using convolutional neural networks and long short-term memory networks. Neural computing and applications, 32(17), 12851-12862.

20. Ju, Z., Liu, G., & Wang, L. (2020). Gait classification using ensemble learning based on feature selection. Applied sciences, 10(4), 1314.

21. Li, J., Wang, Y., Liu, H., Sun, W., Zhang, L., & Zhang, S. (2020). Gait recognition using 3D deep convolutional neural networks. Journal of Ambient intelligence and humanized computing, 11(10), 4365-4373.

22. O'Reilly, M. A., Whelan, D., & Ward, T. E. (2016). Gait classification using time-series shapelets. IEEE Transactions on biomedical engineering, 64(5), 1062-1072.

23. Wang, J., Wang, J., Ji, Q., & Cheng, X. (2019). Gait classification using a convolutional neural network with hierarchical feature learning and metric learning. IEEE transactions on neural networks and learning systems, 31(5), 1516-1528

24. URL: [www.javatpoint.com/algorithm-for-machine-learning]