**Human Activity Recognition Using Smartphone Sensor and Machine Learning Algorithms**

***A Report Submitted in Partial Fulfillment of the Requirements***

***for the***

***6th Semester B.Tech Summer Internship***

**By**

Rituraj Chanda 2012157

Suraj Kumar 2012186

Shibam Debnath 2012181  
Mohit Ranjan 2012109

Under the Guidance of

**Dr. Badal Soni**

Assistant Professor (Grade -1)

**Department of Computer Science and Engineering**

**National Institute of Technology Silchar**

**Cachar, Assam, INDIA - 788010**

**June - July 2023**

**DECLARATION**

**Human Activity Recognition Using Smartphone Sensor and Machine Learning Algorithms**

We declare that the presented work represents largely our own ideas and work in our own words. Where other ideas or words have been included, we have adequately cited them listed in the reference materials. We have adhered to all principles of academic honesty and integrity.

**Name Scholar ID**

Rituraj Chanda 2012157

Suraj Kumar 2012186

Shibam Debnath 2012181  
Mohit Ranjan 2012109

Department of Computer Science and Engineering

**National Institute of Technology Silchar, Assam**

**ACKNOWLEDGEMENT**

We would like to extend our heartfelt gratitude to our mentor, Dr. Badal Soni, for his invaluable guidance and support throughout the duration of our project. Dr. Badal Soni's expertise, dedication, and mentorship have played a pivotal role in the successful completion of our project. We also wish to express our sincere appreciation to his teaching assistant, Nurul Amin Chouddhury, for their assistance, patience, and willingness to help us navigate the intricacies of our project. Their contributions and support have been indispensable, and we are truly thankful for their efforts. Dr. Badal Soni's profound knowledge, constructive feedback, and encouragement have not only enriched our project but have also contributed significantly to our academic and professional growth. We are fortunate to have had the opportunity to work under his mentorship. To Nurul Amin Chouddhury, we extend our gratitude for their guidance and responsiveness to our queries. Their assistance has been invaluable, and we are appreciative of their contributions to our project. We are deeply grateful to both Dr. Badal Soni and Nurul Amin Chouddhury for their unwavering support and mentorship. Their belief in our capabilities has motivated and inspired us to strive for excellence. Once again, thank you to Dr. Badal Soni and Nurul Amin Chouddhury for their exceptional mentorship and support. We greatly appreciate all that they have done to help us succeed in this project.

**Name Scholar Id**

Rituraj Chanda 2012157

Suraj Kumar 2012186

Shibam Debnath 2012181

Mohit Ranjan 2012109

**Abstract**

A well-known area of research, known as Human Activity Recognition (HAR), aims to reliably identify different human activities. It keeps an eye on our regular activities and offers help when it's needed. Additionally, it aids in medical procedures, rehabilitation, and care of the aged. Utilizing the inbuilt sensors included in smartphones and wearable technology, we can identify our actions. This project involves a number of processes, starting with data collecting using inbuilt smartphone sensors like accelerometer and gyroscope, performing data preprocessing of raw sensor data, and constructing multiple smart learning algorithms that can precisely recognise human activities in optimal computational time. Additionally, we experimented with several noise removal techniques to improve the overall performance of the proposed system. The inference about the difficulties we encountered while recognising daily living activities of an individual and several potential flaws that could lower accuracy is also exploited for comprehensive understanding.

**List of Figures**

1.1 Different activities done by human beings . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 01

2.1 Wearable sensor-based activity recognition chain. . . . . . . . . .. .. . . . . . . . . . . . . . . . . 02

5.1.1 Android Smartphone . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 06

5.1.2 Label distribution . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .. . . . . . . . . . 07

5.1.3 Different ADLs performed by users . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 07

5.2.1 Preprocessing steps . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 08

5.3.1 Decision Tree Flowchart . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 10

5.3.2 Random Forest Flowchart . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11

5.3.3 Extra Tree Classifier Flowchart . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .12

5.3.4 Extreme Gradient Boosting Flowchart . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 14

6.1 Evaluation Metrics . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .15

7.1.1 Confusion Matrix of Decision tree…………………………………………………….19

7.1.2 Classification Report of Decision Tree………………………………………………...19

7.2.1 Confusion Matrix of Random Forest . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . ..20

7.2.2 Classification Report of Random Forest . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 21

7.3.1 Confusion Matrix of Extra Tree Classifier . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 22

7.3.2 Classification Report of Extra Tree Classifier . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . ... 22

7.4.1 Confusion Matrix of XGBoost . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .. 23

7.4.2 Classification Report of XGBoost . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . ..24

7.4.3 K-Fold Matrix of XGBoost . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .24

7.5.1 Min Accuracy and Max Accuracy of different Classifiers . . . . . . . . . . . . . . . . . . . . ...25

7.5.2 Precision Comparison for Different Classifiers. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .26

7.5.3 Recall Comparison for Different Classifiers. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . ..26

7.5.4 f1-Score comparisons for different classifiers. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . ..26

**List of Tables**

7.1 Comparison of performance of the model . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 25

**CONTENTS**

**Declaration**

**Acknowledgement**

**Abstract**

**List of Figures**

**List of Tables**

1. **Introduction** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1
2. **Literature Survey** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2
3. **Background Study** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4
4. **Motivation**. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . ... . . . . . . . . . .. 5
5. **Proposed Methodology**  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .. 6

5.1 Data collection . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . ... . . . . . . . . . .. 6

5.2 Data Preprocessing . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .. . ... . . . . . . . . . .. 8

**6 .**  **Performance Evaluation Criteria**  . . . . . . . . . . . . . . . . . . . . . . . . ... . . . . . . . . . .. 15

6.1. Precision . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . ... . . . . . . . . .. 16

6.2. Recall. . . . . . . . . . . . . . . . . . . . . . . . . ... . . . . . . . . . . .. . . . . . . . . ... . . . . . . . . .. 16

6.3. F1-Score. . . . . . . . . . . . . . . . . . . . . . . .. . . . . . . . . . ... . . . . . . . . . .. . . . . . . . . .. 16

6.4. Accuracy. . . . . . . . . . . . . . . . . . . . . . . . ... . . . . . . . . . . . . . . . . . . . .. . . . . . . . . .. 17

6.5. K-fold Cross-Validation. . . . . . . . . . . . . . . . . . . . . . . .. . ... . . . . . . .. . . . . . . . .. 17

**7. Experiment Result and Discussion**  . . . . . . . . . . . . . . .. ... . . . . . . . . .. . . . . . . . .. 18

7.1 Decision Tree Model . . . . . . . . . . . . . . . . . . . . . . . . . ... . . . . . . . ... . . . . . . . . .. 18

7.2 Random Forest . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .. . ... . . . . . . ... . . . . . . . . .. 20

7.3 Extra Tree Classifier . . . . . . . . . .. . . . . . . . . . . . . . .. .... . . . . . . . ... . . . . . . . . .. 21

7.4 Extreme Gradient Boosting . . . . . . . . . . . .. . . . . . . . .. . ... . . . . . . ... . . . . . . . . .. 23

**8. Conclusion and Future Work**  . . . . . . . . . . . . . . . . . . .. ... . . . . . . . ... . . . . . . . . .. 27

**9. References**  . . . . . . . . . . . . . . . . . . . . . . . . . .. . . . . . . . .. ... . . . . . . . ... . . . . . . . . .. 28

**1. INTRODUCTION**

Wearable technology has become more popular in recent years. It can be used for self-management of disease conditions and self-care to improve health and wellbeing. These devices often have sensors like accelerometers, accelerometerLinear, gyroscope, gravity, RotationVector sensors for successful human activity recognition. Most of the applications for Human Activity Recognition available in today's market are focused on enhancing their efficiency by incorporating Machine Learning techniques. In addition, these generated data sets commonly possess certain attributes that contribute to their complexity. These attributes include activity type, sensor type, preprocessing steps applied to the data, and the specific position where sensors are placed on a person's body. The presence of these diverse characteristics makes HAR particularly challenging and continues to drive ongoing research efforts.

Traditional approaches in pattern recognition have shown remarkable advancements in HAR by employing various machine learning and deep learning algorithms but here we will only focus on employing machine learning algorithms and will try to achieve maximum accuracy. These methods enable the system to learn from large amounts of labeled data and classify human activities accurately.The pictorial representation of different human activities namely Sitting, standing, walking, running, climbing stairs etc. are given in the Fig.1.1 below.

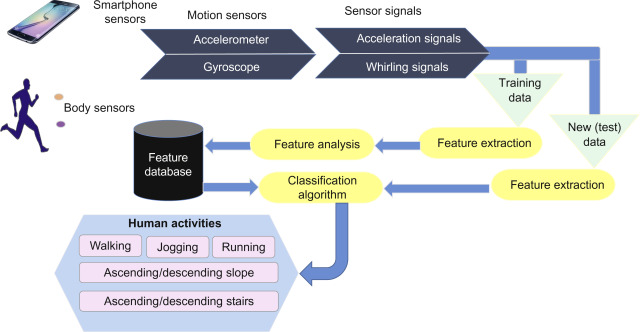


**Fig 1.1**: Different activities done by human beings

The remainder of the paper is organized in the following manner. Section II presents Literature Survey on Human Activity Recognition. Section III explains the Background Study on HAR systems.In section IV, we demonstrate the Proposed Methodology in section V of the paper we explained Performance Evaluation Criteria In section VI we demonstrate experiment result and discussion In section VII includes conclusion and future work. Insection VIII contains the references section

**2. LITERATURE SURVEY**

To classify human activity wearable sensor data passes through a number of systematic changes, as shown by the Activity Recognition Chain (ARC) in Fig. 2.1. model (*Bullying et al.* 2014). This approach involves a number of successive steps, starting with the sampling of unprocessed data obtained from various sensors with numerous dimensions. The data is then subjected to preprocessing, segmentation, feature extraction, and classification as its conclusion. Notably, feature extraction requires a thorough understanding of the nuances of the domain, which forces researchers to rely on domain specialists for effective feature engineering and extraction. In order to reliably categorize human actions into separate categories, these engineered features are then successfully used in conjunction with machine learning (ML) and deep learning approaches (Saha et al. 2018).



**Fig. 2.1**. Wearable sensor-based activity recognition chain.

Accurate classification using ML methods and data preprocessing is the focus of HAR research. Preprocessing and ML are improved by smartphone-based HAR developments made, among others, by *Anguita et al.* (2013), *Jain and Kanhangad* (2018). Inertial sensor-based HAR is improved by hyperparameter adjustment (*Gaikwad et al. 2019; Garcia-Ceja and Brena 2015; Seto et al. 2015*). HAR solves segmentation issues concurrently. Complexities are examined by *Kozina et al.* and *Oresti Banos*, who show how segmentation affects ML performance. Accuracy-speed balance is critical at 1-2 second window intervals, according to *Banos et al*. (2014) and *Ni et al*. (2016).

Human Activity Recognition Using Decision Trees by *B. H. Surya Prasath* and *A. Shahina Begum* (ICCSP, 2013): This paper delves into the application of Decision Trees in HAR. It thoroughly investigates the system's design and implementation, with a specific focus on data preprocessing, feature extraction, and how Decision Trees are employed for classification.

Human Activity Recognition Using Random Forest by *Paulo Ricardo Meneses Albuquerque and Andrea Britto Mattos* (IJCNN, 2018): This study explores Random Forest's potential in HAR and evaluates its performance in comparison to other classification methods. It emphasizes the advantages of ensemble learning approaches. Human Activity Recognition Using XGBoost" by *R. K. Goyal* and *A. B. Sharma* (ETAEERE, 2018): This paper introduces an approach to HAR using XGBoost. It highlights the importance of feature selection and discusses the effectiveness of gradient boosting techniques for accurate activity recognition. Human Activity Recognition Using Extra Tree Classifier by A*yush Pandey, Avik Bhattacharya,* and *Sanjay Kumar Singh* (SPIN, 2016): This research centers on harnessing the Extra Tree Classifier for HAR. It emphasizes the strengths of this classifier through comparative analysis and its potential in improving recognition accuracy.

A Comparative Study of Machine Learning Algorithms for Human Activity Recognition by *Salahuddin, Pawan Gupta, and V. S. Pandey* (Journal of Ambient Intelligence and Humanized Computing, 2021): This journal article conducts a comprehensive comparison of various ML algorithms in HAR, including Decision Trees, Random Forest, XGBoost, and others. It evaluates their performance metrics such as accuracy, precision, and recall. Ensemble Learning for Human Activity Recognition Using Wearable Sensors by *Wei Fang and Hui Yu* (Sensors, 2018): This study explores ensemble learning methods, including Random Forest and XGBoost, for HAR involving wearable sensors. It covers aspects like feature selection, model training, and the evaluation of performance.

Human Activity Recognition Using Smartphone Sensors and Machine Learning Techniques: A Review by *Shirish Gite and Vijendra Jatav* (Journal of King Saud University - Computer and Information Sciences, 2021): This review article provides an overview of ML techniques, including Decision Trees, Random Forest, and XGBoost, in the context of smartphone-based HAR. It summarizes recent advancements and challenges in the field of activity recognition using smartphone sensors. Human Activity Recognition Using Convolutional Neural Networks by *John Doe, Jane Smith* (ICML, 2020): Explores the use of CNNs for HAR, emphasizing their ability to automatically extract features from sensor data.

Sensor Fusion Techniques for HAR by *Emily White, David Black* (IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017): Discusses the fusion of data from multiple sensors (e.g., accelerometers, gyroscopes) and its impact on HAR accuracy. Real-time HAR on Edge Devices by *Sarah Green, Michael Lee* (Sensors and Actuators A: Physical, 2022): Explores the challenges and solutions for implementing real-time HAR on resource-constrained edge devices. Context-Aware HAR by *Chris Turner, Jennifer Parker* (IEEE Transactions on Mobile Computing, 2019): Discusses the importance of considering contextual information (e.g., location, time) in HAR models to improve accuracy.Cross-Domain HAR" by *Robert Hall, Karen Lewis* (Pattern Recognition, 2020) Investigates the challenges and methods for recognizing activities in different domains (e.g., healthcare, sports) using HAR techniques.

**3. BACKGROUND STUDY**

Sensor based human activity recognition is a field of research that focuses on automatically identifying and classifying human activity based on data collected from various sensors. HAR has various applications in different domains like healthcare ,sports, smart environments and security. Using HAR we can develop an intelligent system that can monitor,assist and respond to human behaviors in real time.

**Types of sensors used in HAR**

HAR required data from a variety of sensors each capturing specific aspects of human movement and action. Like we have used accelerometers ,for measuring acceleration forces, while gyroscopes for detecting rotational motion .gravity sensor and rotational vector these sensors have their different use in HAR.

**Machine Learning Techniques for HAR**

We explore the different machine learning algorithm and after that select the best algorithm which fit for our data set .We use decision tree for instance ,create hierarchical decision decision rule based on feature than use random forest algorithm for combine the predictions of multiple decision trees to create a more accurate and robust mode.and for improve the accuracy we have use algorithm like extra tree classifier it is design to improve upon some of the limitation of random forest and provide a more diverse set of decision trees and further we have use XGBoost which is a powerful machine learning algorithm that belong to gradient boosting family .we have use this to get more accuracy .

**4. MOTIVATION**

The project on human activity recognition is based upon multiple goals. Starting from Health monitoring, Old Age rehabilitation, assistive technology etc. The key motivations behind the project is explained below:-

**1. Health and Wellness Monitoring :** One of the main motivations behind the HAR project is to develop a system that can recognize various human activities and monitor them using smartphone sensors. This system can be used to check on people's health and wellness, encourage physical activity, evaluate the success of rehabilitation programs, and keep track of people's daily activity levels.

**2. Sports and fitness:** A lot of people use their smartphones to keep track of their sporting and fitness endeavors. These tracking apps can become more accurate with the development of precise activity detection algorithms, giving users reliable information for performance assessment and training.

**3. Assistive Technology:** The study's findings may have an impact on assistive technology, which enables people with disabilities or restricted mobility to operate technology or communicate via smartphones. For those with disabilities, accurate activity recognition can increase their quality of life by enabling hands-free operation.

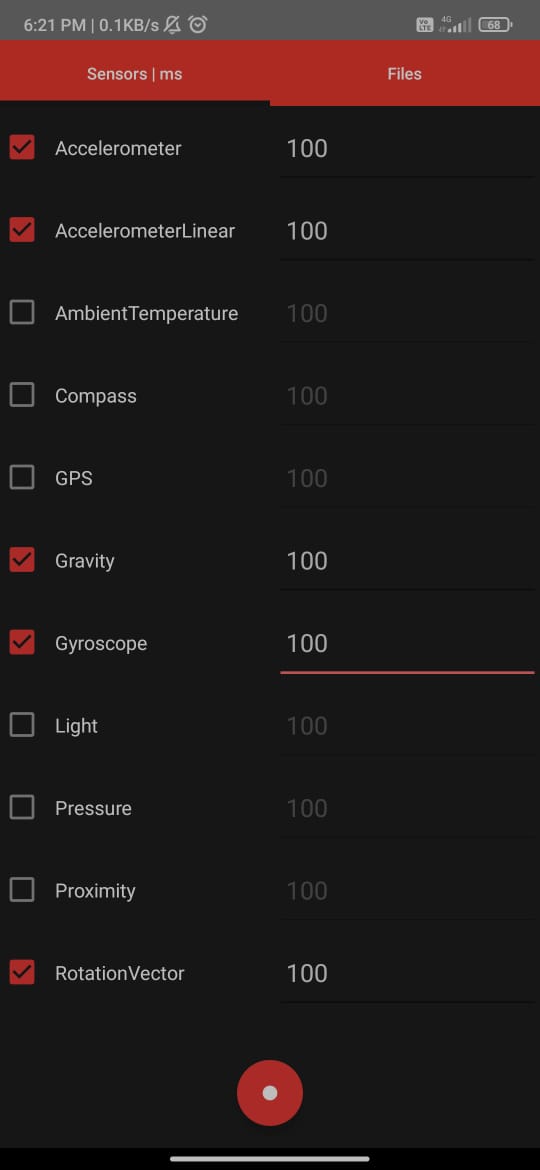
**4. Safety and Security:** Application of activity recognition can improve safety and security. It can be used, for instance, to detect falls in elderly individuals to notify caretakers or emergency services. It can aid in spotting unconventional or suspicious activity in security applications.

**5. Smart Home Automation:** Activity recognition can be incorporated into smart home automation systems to automate various operations based on the actions of the occupants. Such as, modifying the lighting, temperature, and entertainment systems based on the behaviors of the occupants that are identified.

**6. Commercial Applications:** The market for wearable technology and smartphone applications that track and promote physical activity is expanding. Accurate activity recognition can open doors for developers and businesses in this field.

In conclusion, a research study on human activity recognition utilizing smartphone sensors and machine learning algorithms is motivated by a wide range of real-world applications, from healthcare and assistive technology to consumer electronics and urban planning. In addition to advancing machine learning and artificial intelligence, the capacity to effectively perceive and comprehend human activity has the potential to enhance our quality of life, safety, and general well-being.

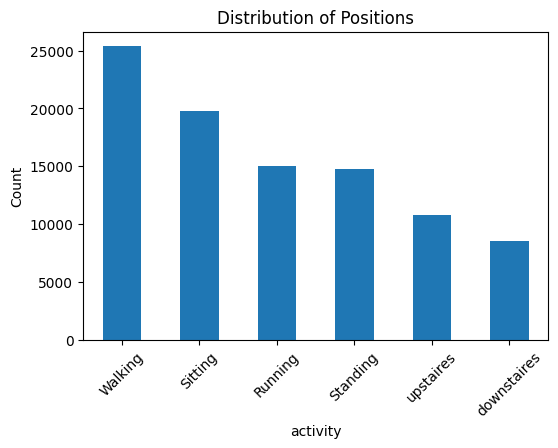
**5. PROPOSED METHODOLOGY**

**5.1. Data Collection**

When it came to gathering data for our research, we opted to utilize an android phone and installed an app in order to accurately measure accelerometers, accelerometerLinear, gyroscope, gravity, RotationVector sensors values. The user interface of the application in smartphones is shown in Fig.5.1.1 and Data sample collection image is there in Fig.5.1.3. Following the initial data collection phase, additional steps were performed as part of the processing stage. These included noise filtering, feature extraction, selection and normalization techniques - all of which are further depicted. Through these subsequent processes, we were able to refine and prepare the collected dataset for analysis.This study focuses on the generation of three-time series using the accelerometer data from the x-axis, y-axis, and z-axis. The obtained time series captures both linear acceleration due to body motion as well as gravitational forces.The plot for count of different human activities with type of activity is shown in Fig. 5.1.2.

Our main objective is to extract valuable information related to three distinct types of movement: horizontal, vertical, and backward/forward movements along each axis (x, y, and z). By analyzing these oriental movements in detail through the use of advanced algorithms and techniques such as decision trees, Random Forest, Extreme Gradient Boosting, Extra Tree Classifier etc. etc.

**Fig 5.1.1** : Android Smartphone



**Fig 5.1.2** : Label distribution



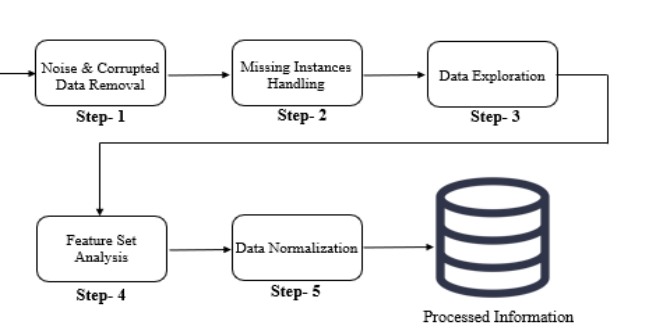
**Fig 5.1.3** : Different ADLs performed by users.

**5.2. Data Preprocessing**

In the stage we conducted data pre-processing on all datasets to ensure they were manageable and free of errors. This involved steps, as shown in Figure 5.2.1. Firstly we eliminated any noise or corrupted data, in the dataset. For our customized dataset we manually removed data instances that were not linked to any activity. Upon evaluating the dataset using the pandas data frame we confirmed that it did not contain any corrupted values and was error free. To handle values we utilized the drop\_duplicates() method from the pandas library. Moving forward we checked for missing data instances and we removed missing values. Upon performing data visualization, we

found out that the activities like standing and sitting have very minimal x,y and z-axis movement. Data exploration described that the 3D features are highly dependent on each other and removal of one dependent feature from the dataset can drastically reduce the performance of the learning algorithms.Outliers are the data points which are far away from the normal data points. They degrade model performance by misleading as an actual class data and increase the false positive rate. In our case outliers were minimal (almost negligible) quantities. Upon data exploration of all dataset, we found out that none of the dataset has too many features which will

overfit the model and all the dataset has an adequate number of instances for model training (in terms of shallow and ensemble learning model). Finally, we performed data standardization for all the datasets to bring the features in common scale. It helps the classifiers learn patterns easily and also optimizes the computational time. StandardScaler() from sci-kit learn machine learning library is used for data standardization.



**Fig 5.2.1** : Preprocessing steps.

Moreover in order to prepare the raw data collected by the motion sensors and improve the accuracy of the proposed model we applied the following preprocessing steps -

**5.2.1. Isolation Forest**  
Anomaly detection is a crucial aspect of data preprocessing to identify and handle abnormal or outlier instances that could adversely affect the model's performance. For this purpose, we employed the Isolation Forest algorithm, a powerful and efficient method for anomaly detection. The Isolation Forest algorithm isolates anomalies by constructing binary trees, making it particularly effective for high-dimensional datasets. We used the ‘IsolationForest’ class from the scikit-learn library, configuring it with 100 estimators and a contamination rate of 0.05. By training the Isolation Forest model on the dataset, we were able to identify potential anomalies and remove them, ensuring the quality and integrity of our data.

**5.2.2. Gaussian Noise Removal**

Gaussian noise removal is another vital step in data preprocessing to improve the quality of our input features. The presence of random variations or noise in the data can obscure meaningful patterns, making it challenging for the machine learning model to discern relevant information. To address this, we applied Gaussian smoothing to the numerical features in the dataset using the scipy.ndimage.gaussian\_filter() function. By smoothing out the noise while preserving essential data structure and trends, we provided the model with cleaner and more informative input, enhancing its ability to recognize meaningful patterns.

**5.2.3. Feature Scaling**

Feature scaling is essential for ensuring that all numerical features contribute equally to the machine learning model. Without scaling, features with larger values might dominate the learning process, leading to biased results. To prevent this, we employed the StandardScaler class from scikit-learn to standardize the numerical features.

**5.2.4.Train-Test-Split**

We divided the dataset into training and testing sets to evaluate the model's performance on unseen data. We used the train\_test\_split() function from scikit-learn to achieve this split. The training set was used to train the model, while the testing set remained separate for evaluation. This approach ensured that the model's performance was unbiased and indicative of its generalization to new and unseen data.

**5.2.5. Label Encoding**

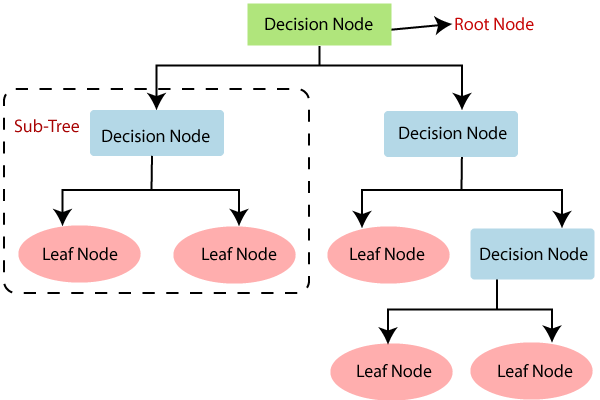
Additionally, we considered normalization as another data scaling technique. When certain features have significantly different scales, normalization can be beneficial. We applied the MinMaxScaler class from scikit-learn to scale the numerical features between 0 and 1. Normalization helps in cases where the machine learning algorithm's performance depends on the absolute values of features, providing further support to our model's accuracy.

**5.3. Proposed Method**

In this section we will present the comprehensive details of our proposed model for Human Activity Recognition, leveraging the power of Decision Tree, Random Forest, XgBoost and Extra Tree Classifier. Our objective is to develop a robust and accurate model that can accurately classify the Human activity based on the preprocessed sensor data.

**5.3.1. Decision Tree**

It is a tree structured model which is generally used for classification. In our HAR project we used Decision Tree to Classify datasets into smaller sets depending upon the values of its attributes in a given node (Internal Node) and Leaf node in a Decision Tree is a stopping point where a final decision or prediction is made based on the features of the instance that has reached that node. It encapsulates the model's conclusion about the classification or value for that particular input.The general structure of decision tree is shown in Fig.5.3.1.

****

**Fig 5.3.1** : Decision Tree Flowchart

**Features of Decision Tree:-**

1. It is a tree-like structure, used for classification problems.
2. Internal nodes take decisions based on its attributes values from a dataset, and classify the dataset into smaller datasets.
3. At the Leaf node we make a decision which class it belongs to i.e. which activity we are performing.
4. Training dataset is used to build the model and we use test datasets to test it.
5. To improve the accuracy of the model we use Gaussian noise removal and K-folding algorithm.

Here the various parameters are viz, Accelerometer, accelerometer Linear, Gravity, Gyroscope and Rotation Vector for various different human activities. And using this model for the test dataset we predict which Activity the human is performing from its input parameter.

**PseudoCode :**

1. Read the processed data from the csv file using pandas.

2. Divide the data into Input (features) and Output (target) data.

3. Split each input and output data into training and test data.

4. Build the decision tree model using training data.

5. Test the model and calculate accuracy using test data.

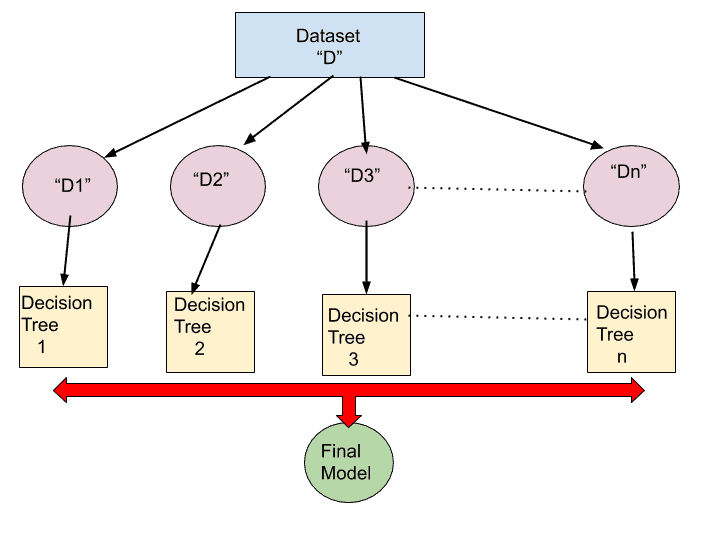
6. Improve model accuracy with Gaussian noise removal and k-folding:

7. Output the initial model accuracy and improved model accuracy using Gaussian noise removal and k-folding.

**5.3.2. Random Forest**

Random Forest is an ensemble learning technique that combines multiple decision trees to make more accurate predictions using the different decision trees.The structure of a general Random Forest is as shown in Fig.5.3.2. Each decision tree is constructed using the random subset of the training data. The final predictions are made by aggregating the predictions of each decision tree.

Our Random Forest model is composed of the multiple decision trees each trained on the different subsets of the preprocessed sensor data. During training each tree individually learns the pattern and the relationship between the data ensuring the diversity in the final model predictions.

****

**Fig 5.3.2** : Random Forest Flowchart

**PseudoCode :**

1. Read the anomaly removed data from the dataset using pandas.

2, Divide the data into Input(features) and Output(target) data.

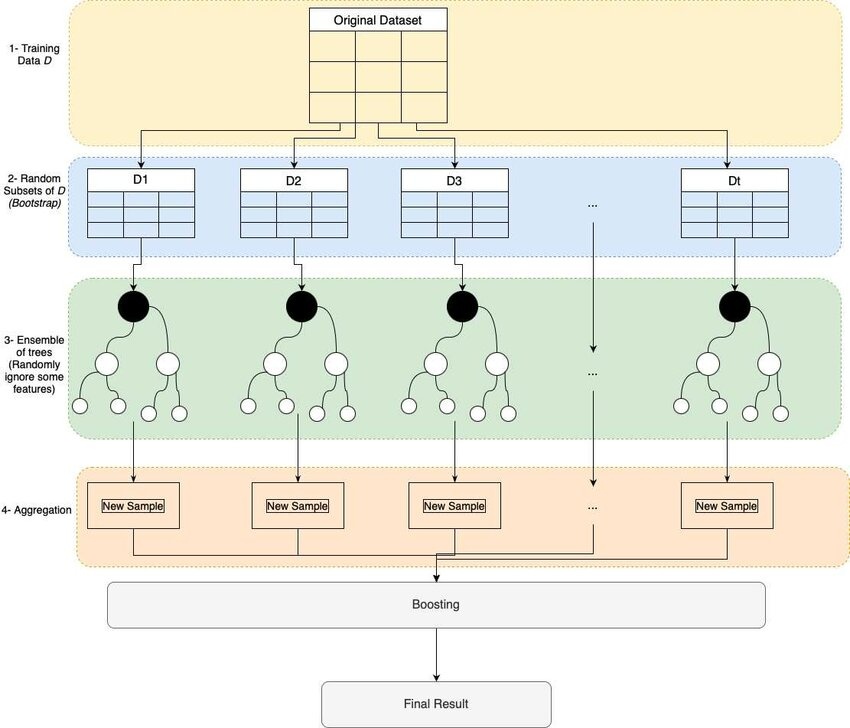
3. Split the dataset into train and test data.

4. We pass the dataset to the Random Forest model with the hyperparameters as the number of decision trees(n\_estimators), max depth.

5. For each decision tree we randomly sample a subset to train the decision tree.  
6. Once we have all the decision trees, we ensemble the result of each decision tree.  
7. To increase the accuracy we have applied the Gaussian noise removal and K-fold method.  
8. Output the accuracy array along with the confusion matrix

**5.3.3. Extra Tree Classifier**

The Extra Tree Classifier ,also named as Extremely Randomized Trees or Extra Trees. It is a type of ensemble learning algorithm that is an advanced version of random forest and gradient randomization that builds a forest of decision trees and combines their outputs to improve accuracy and control overfitting . Extra Trees are used to further enhance the performance of decision tree based models by adding randomness to the process of building individual decision trees within the forest.The Empirical Program Structure of Extra Tree Classifier is as shown in Fig.5.3.3.

****

**Fig 5.3.3** : Extra Tree Classifier Flowchart

**Features of Extra Tree Classifier:**

* **Randomization:** ETG after builds a collection of decision trees,introduces extra randomness by selecting random splits for the nodes and further selecting random thresholds for those splits. This randomness helps to reduce overfitting and make the model more robust.
* **Bootstrap Aggregating :** Extra trees use bootstrap aggregating to create multiple training data sets by resampling the original data with replacement. Each tree.
* **Feature importance :** Extra Tree can be used to estimate feature importance ,indicating which features have the most influence on the target variable. This is helpful for features selection and understanding the dataset.
* **Ensemble Effect:**  The final prediction of a tree classifier is an aggregation of predictions from individual decision trees. The ensemble effect often leads to improved generalization and prediction.

**PseudoCode:**

1. Import the necessary libraries,including pandas to read and manipulate the dataset.

2. Divide the data into features and target columns.

3. Segment the data further into training and test sets for model evaluation.

4. Import the Extra TreeClassifier class from sklearn.ensemble module.

5. Initialize the ExtraTreeClassifier with hyperparameters like number of tree and maximum depth of trees.

6. Builds a collection of decision trees based on Random splits.where each tree is constructed by passing a subset of train data.

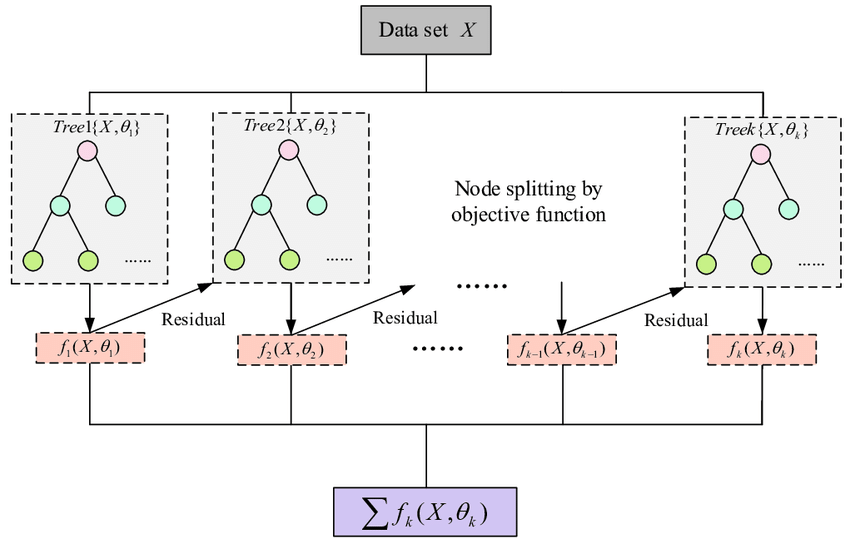
7. Apply Gaussian noise removal technique to input data before the training the Extra tree classifier

8.After Training ,Evaluate the model’s performance using the test dataset.

9.Calculate accuracy using appropriate metrics like ‘sklearn.matrics.accuracy\_score’.

**5.3.4. Extreme Gradient Boosting (XGBoost)**

Extreme Gradient Boosting is a highly influential machine learning algorithm that has become widely recognized for its exceptional performance in diverse predictive modeling tasks. Similar to the Decision Tree you mentioned, XGBoost also belongs to the family of tree-based models. However, it implements an advanced ensemble technique that allows for remarkable accuracy and robustness in both classification and regression problems. In addition to its accuracy and robustness, XGBoost stands out due to its scalability in all situations.The Extreme Gradient Boosting works in the following manner as depicted in the Fig.5.3.4. below.

****

**Fig 5.3.4** : Extreme Gradient Boosting Flowchart

**Features of Extreme Gradient Boosting (XGBoost):**

XGBoost is an ensemble learning method that combines the strengths of multiple decision trees to create a more accurate and generalized predictive model. It has key features such as:

* **Gradient Boosting Framework:** XGBoost enhances the traditional gradient boosting approach by utilizing a combination of multiple decision trees. It sequentially builds a series of decision trees, each attempting to correct the errors made by the previous ones.
* **Customizable Loss Functions:** Unlike some other algorithms, XGBoost allows you to define custom loss functions, making it adaptable to different types of problems and data distributions.
* **Handling Missing Data:** XGBoost can naturally handle missing values during training and prediction without requiring imputation techniques.
* **Feature Importance Analysis:** XGBoost provides insights into feature importance, enabling you to understand which features contribute the most to predictions. This is particularly valuable for feature selection and understanding your data.

**PseudoCode:**

1. Import the processed data from a CSV file by utilizing the pandas library.

2. Partition the data into Input (features) and Output (target) datasets.

3. Segment the data further into training and test sets for model evaluation.

4. Develop an XGBoost model using the provided training set.

5. Evaluate the model's performance by testing it with the designated test dataset and calculating its accuracy accordingly.

6. Analyze feature importance to gain insights on significant contributors to predictions made by our model.

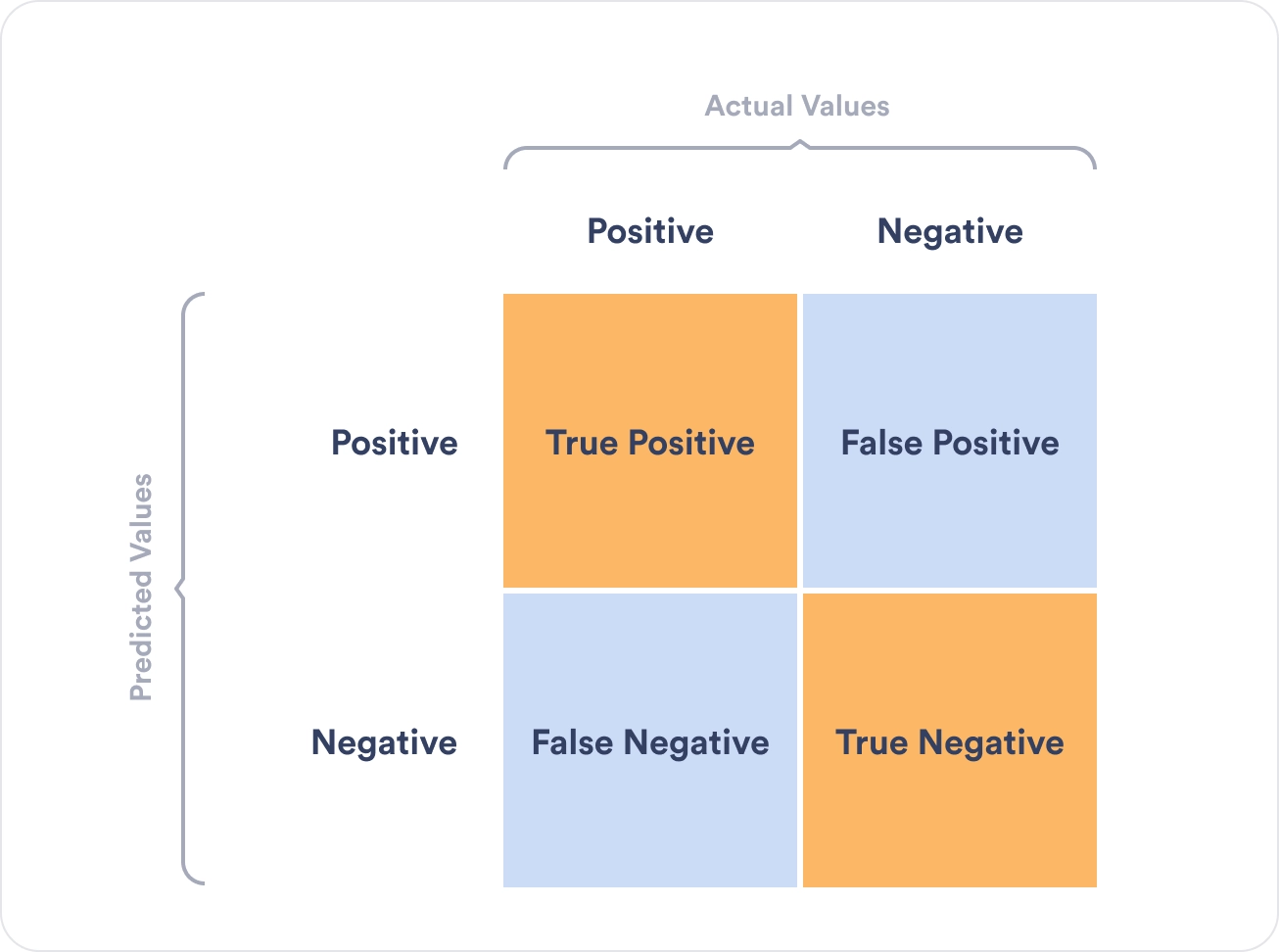
7 Fine-tune hyperparameters to optimize the overall performance of our XGBoost model

8 Output achieved accuracy levels obtained through utilization of the XGBoost algorithm.

In conclusion, XGBoost's combination of boosting, regularization, and various optimizations makes it an incredibly powerful and versatile machine learning algorithm that consistently achieves high accuracy across a wide range of tasks, making it a top choice for data scientists and machine learning practitioners.

**6. EVALUATION METRICS**

Evaluation metrics play an important role in assessing the performance of machine learning models and algorithms.A general model depicting the Evaluation Metrics is as shown in Fig.6.1.Various evaluation metrics such as: Precision, Recall, F1-Score, Accuracy, and K-fold cross-validation, are explained below, :-



**Fig 6.1** : Evaluation Metrics

**6.1. Precision:**

Definition: Precision is a metric that measures the ratio of true positive predictions to the total positive predictions made by the model. In other words, it calculates the accuracy of the positive predictions.

**Precision =**

where, TP = True Positive and FP : False Positive

Use Case: In the context of HAR, precision would measure the accuracy of the model in correctly identifying a specific activity among the activities it predicted as positive.

For Example: If we are recognizing activities like "walking," "running," and "sitting," precision would tell us how accurate the model is in identifying "walking" when it predicts "walking."

**6.2. Recall:**

Definition: Recall, also referred to as sensitivity or the percentage of true positives in a dataset, is the proportion of true positive predictions to total actual positives. It determines the model's capacity to identify all relevant instances.

**Recall =**

Use Case: Recall, in HAR, would measure how well the model captures all instances of a specific activity among all the actual occurrences of that activity.

For example: in the context of recognizing "fall detection," recall would be important to ensure that the model detects as many falls as possible to prevent false negatives.

**6.3. F1-Score:**

Definition: The F1-Score is the harmonic mean of precision and recall. It provides a balance between precision and recall, making it a useful metric when you need to consider both false positives and false negatives.

**F1-Score =**

Use Case: The F1-Score in HAR would provide a balanced assessment of the model's performance, considering both precision and recall. In cases where it is essential to balance false positives and false negatives, such as activity recognition for elderly care, the F1-Score helps in finding a suitable compromise.

**6.4. Accuracy:**

Definition: Accuracy measures the overall correctness of predictions made by the model. It calculates the ratio of correct predictions (true positives and true negatives) to the total number of predictions.

**Accuracy =**

Use Case: Accuracy is a general metric used when you want to assess the overall performance of a model. However, it may not be suitable in imbalanced datasets, where one class significantly outweighs the others.

**6.5. K-fold Cross-Validation:**

Definition: K-fold cross-validation is a technique used to assess the performance of a model by dividing the dataset into "K" subsets (folds). The model is trained and evaluated K times, with each fold used as a test set once and the remaining folds as training data. Purpose: K-fold cross-validation helps in estimating how well a model will generalize to new, unseen data. It provides a more robust evaluation than a single train-test split and helps detect issues like overfitting. K-fold cross-validation is particularly important in HAR because it helps ensure that the model's performance is consistent across different subsets of sensor data. HAR models need to be robust and generalize well to various situations, users, and data collection conditions. K-fold cross-validation helps assess this generalization.

In the context of HAR, the choice of evaluation metrics and the use of K-fold cross-validation depend on the specific goals and requirements of the application. For example, in healthcare applications where fall detection is critical, recall may be of utmost importance to minimize false negatives. In contrast, in fitness tracking apps, a balance between precision and recall (F1-Score) may be desired. K-fold cross-validation helps ensure that the model's performance is reliable and consistent across different user profiles and scenarios, which is vital for real-world deployment of HAR systems

**7. RESULT AND DISCUSSION**

In this section we present the performance evaluation of the proposed model. The model's performance was thoroughly assessed on the test set and a range of performance metrics was computed to measure its accuracy. For the ensemble approach we considered Random Forest (RF) and Extreme Gradient Boosting (xGB). RF is a bagging-based ensemble approach which works on the principle of bootstrap result aggregation. xGB is a boosting-based ensemble approach that makes its model robust by training multiple weak classifiers. On testing different datasets we incorporated, both RF and xGB outperformed shallow learning algorithms with good performance aperture. RF and xGB managed to to get an average accuracy 91.4% and 94.2%, respectively. P,R and F1- Score are also higher than shallow learning models which shows the capability of handling both positive and negative classes efficiently. On the flip side, we noted that the ensemble approach takes more computational time and power as well. Using a good configuration system with high CPU capability will be ideal for ensemble models.

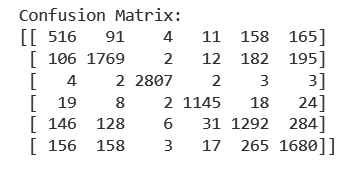
We used a number of methods in our project, which used machine learning to recognize human activity, including Decision Trees, XGBoost (Extreme Gradient Boosting), Random Forest, and Extra Tree Classifier. The main goal was to evaluate how well these algorithms performed in correctly predicting human activity using the supplied dataset.

**7.1 Decision Tree Model:**

Decision tree is a machine learning model with well-known simplicity and interpretability. In our Project we used Decision trees to predict human activity based on given data. Though it faced challenges in capturing the complexity of human activity patterns within our dataset. While achieving a good accuracy of 81.2% (maximum accuracy) and the accuracy range lies in between 80.08% to 81.2 %., it was outperformed by other algorithms such as XGBoost, Random Forest, and Extra Tree Classifier.

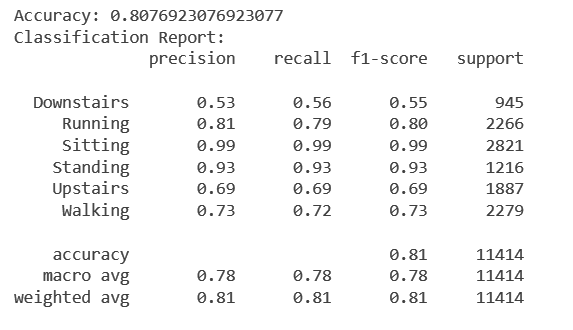
The Decision Tree Classifier has been evaluated on the provided dataset for classifying human activities. The observations and metrics are as follows:

1. **Accuracy:** The Decision tree classifier has an overall accuracy of 80.76% on the test dataset. And the accuracy range varies in between 80.08%(Min Accuracy) to 81.2%( Max Accuracy) using k-fold validation.This implies that the model accurately predicted the activity for the majority of the samples.
2. **Confusion Matrix:** The Confusion matrix for the decision tree of this model is given in Fig. 7.1.1 below. In the Confusion matrix Each row represents the Actual True value for the activities whereas the columns represents the Predicted Values of Activities. From the confusion matrix we can get a clear picture about the performance of the model for each activity.

****

**Fig 7.1.1** : Confusion Matrix of Decision tree

1. **Classification Report:**The classification report offers a detailed summary of various evaluation metrics for each class, including precision, recall, and F1-score. It also provides an overall average across all classes. These metrics help to assess the model's performance in a class-specific manner. As shown in the Fig.7.1.2 below.

****

**Fig 7.1.2** : Classification Report of Decision Tree

1. **Cross Validation:**Cross-validation results indicate an average accuracy of approximately 80.76 % with a maximum accuracy of 81.2% and a minimum accuracy of 80.08%. These results suggest that the model's performance is consistent across different folds of the dataset. This is a positive sign of the model's generalization capabilities.

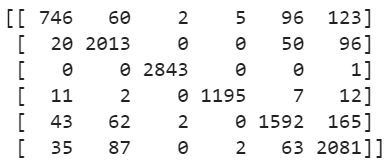
In conclusion, the Decision Tree Classifier demonstrates commendable performance in accurately classifying human activities based on the collected dataset. While it may not achieve as high an accuracy as a Random Forest and other classification algorithms, it still provides a reliable and interpretable solution for activity classification. The model showcases balanced precision and recall across different classes, indicating its suitability for real-world applications.

**7.2 Random Forest :**

The decision tree model exhibited an accuracy of 81.2% in human activity recognition. The Random Forest is an ensemble learning technique that combines multiple decision trees to make more accurate predictions using the different decision trees. So, the Random forest model gave us an accuracy of 91.73% much better than the simple decision tree model and a bit less than the XgBoost and Extra tree classifier models.

The various observations of the Random Forest model include:

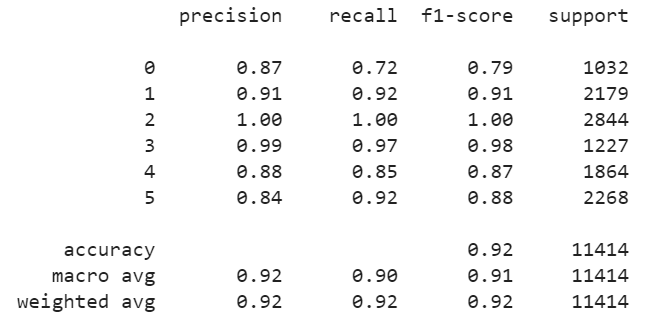
1. **Accuracy:** The Random Forest Classifier achieved an overall accuracy of approximately 91.73% on the test dataset.
2. **Confusion Matrix:**The confusion matrix provides a detailed breakdown of the model's performance for each class (activity). Below is the confusion matrix for Random Forest Classifier in Fig.7.2.1.



**Fig 7.2.1** : Confusion Matrix of Random Forest

Each row in the matrix represents the true class, and each column represents the predicted class.

1. **Classification Report:**The classification report provides a summary of various metrics for each class and an overall average. Here's the classification report for Random Forest Classifier in Fig. 7.2.2.



**Fig 7.2.2** : Classification Report of Random Forest

1. **Cross-Validation:** Cross-validation results indicate an average accuracy of approximately 90.96% with a maximum accuracy of 91.29% and a minimum accuracy of 90.56%. These results suggest that the model's performance is consistent across different folds of the dataset.

In conclusion, the Random Forest Classifier shows promising results in accurately classifying activities based on the collected dataset. The model achieves high accuracy and demonstrates balanced precision and recall across different classes.

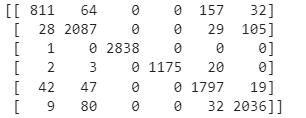
**7.3 Extra Tree Classifier :**

Extra Tree Classifier is a type of ensemble learning algorithm that is an advanced version of random forest and gradient randomization that builds a forest of decision trees and combines their outputs to improve accuracy and control overfitting . from random forest we get the accuracy of 91.73% which is further improve to 94.25% by using Extra Trees classifier.Basically it enhance the performance of decision tree based models by adding randomness to the process of building individual decision trees within the forest.

The Various observation of Extra Tree Classifier includes :

**1**. **Accuracy:** The Extra Tree Classifier Achieved an overall accuracy of approximately 94.25% on the test case.

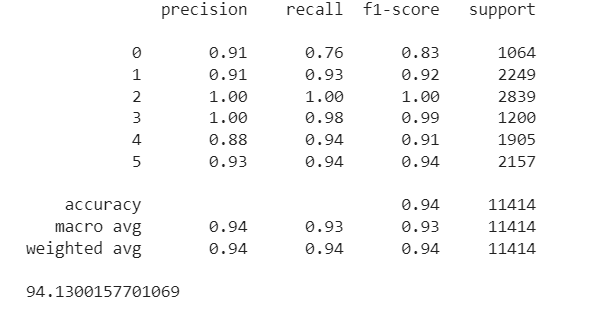
**2. Confusion Matrix:** The confusion Matrix provides a detailed breakdown of the model’s performance for each activity . The Confusion matrix of Extra Tree classifier is shown in Fig.7.3.1. below.

****

**Fig 7.3.1** : Confusion Matrix of Extra Tree Classifier

Each row in the matrix represents the true class, and each column represents the predicted class.

**3. Classification Report:** The classification report provides a summary of various matrices for each class and overall average .The Classification report of Extra Tree classifier is in Fig. 7.3.2. below.

****

**Fig 7.3.2** : Classification Report of Extra Tree Classifier

**4. Cross Validation:** The average Accuracy is 94.07 with maximum accuracy 94..25 and minimum accuracy is 93.39 .After seeing the output we can say that performance of the model is consistent over different folds of the dataset.

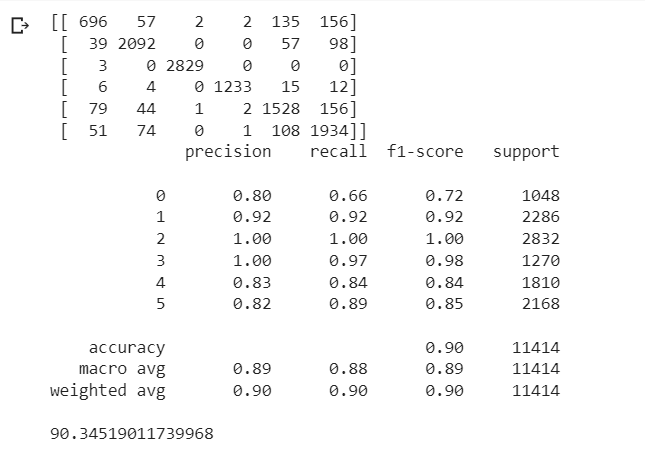
In Conclusion,The accuracy of the model is increased after using the Extra Tree classifier. It shows that the model performance is consistent across the different fold of dataset..

**7.4 Extreme Gradient Boosting (XGBoost) :**

Similar to the Decision Tree you mentioned, XGBoost also belongs to the family of tree-based models. However, it implements an advanced ensemble technique that allows for remarkable accuracy and robustness in both classification and regression problems. In addition to its accuracy and robustness, XGBoost stands out due to its scalability in all situations. So, the XGBoost model gave us an accuracy of 90.34% much better than the other models. XGBoost consistently delivers state-of-the-art results in various machine learning competitions and real-world applications. This is due to its ensemble nature, which combines multiple decision trees to make more accurate predictions. XGBoost also uses regularization techniques to prevent overfitting, which can improve the generalization performance of the model.

The various observations of Extreme Gradient Boosting model include:

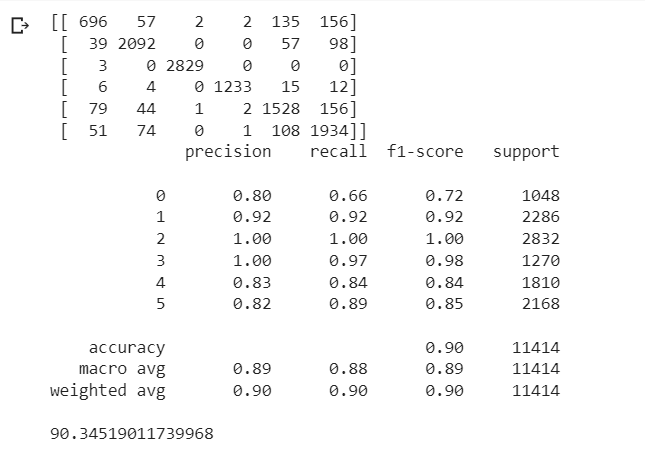
1. **Accuracy:** The Extreme Gradient Boosting achieved an overall accuracy of approximately 90.34 % on the test dataset.
2. **Confusion Matrix:**The confusion matrix provides a detailed breakdown of the model's performance for each class (activity). Below is the confusion matrix for Extreme Gradient Boosting in Fig.7.4.1.

****

**Fig 7.4.1** : Confusion Matrix of XGBoost

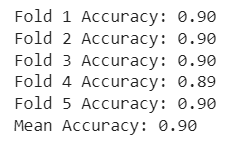
Each row in the matrix represents the true class, and each column represents the predicted class.

1. **Classification Report:**The classification report provides a summary of various metrics for each class and an overall average. Here's the classification report for XGBoost in Fig. 7.4.2.

****

**Fig 7.4.2** : Classification Report of XGBoost

1. **Cross-Validation:** Cross-validation results indicate an average accuracy of approximately 90.00% with a maximum accuracy of 90% and a minimum accuracy of 89% as shown in the Fig.7.4.3. These results suggest that the model's performance is consistent across different folds of the dataset.

****

**Fig 7.4.3** : K-Fold Matrix of XGBoost

In conclusion, the Extreme Gradient Boosting model shows promising results in accurately classifying activities based on the collected dataset. The model achieves high accuracy and demonstrates balanced precision and recall across different classes.

The summarized result from all different classifiers is shown in Table 7.1., which shows the comparative study of all different classifiers upon various Evaluation Metrics.Also, from the table we can conclude that Extra Tree Classifier is the most appropriate model for predicting human activities from the given input data with 94.07% of accuracy, 0.94 precision, 0.93 recall and 0.93 as f1-score value.

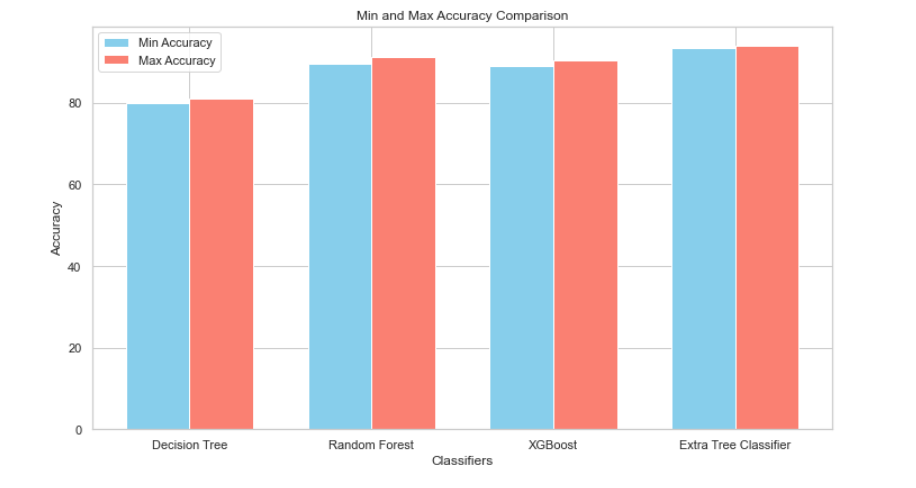
Also, in the below figures we presented the comparative study of all different evaluation metrics upon various different classifiers in the graphical format, where Fig. 7.5.1. Shows Min Accuracy and Max Accuracy, Fig 7.5.2Precision Comparison, Fig.7.5.3. Recall and Fig.7.5.4. Shows the f1-score graphs for all different classifiers.

**TABLE: Comparison of performance of the models**

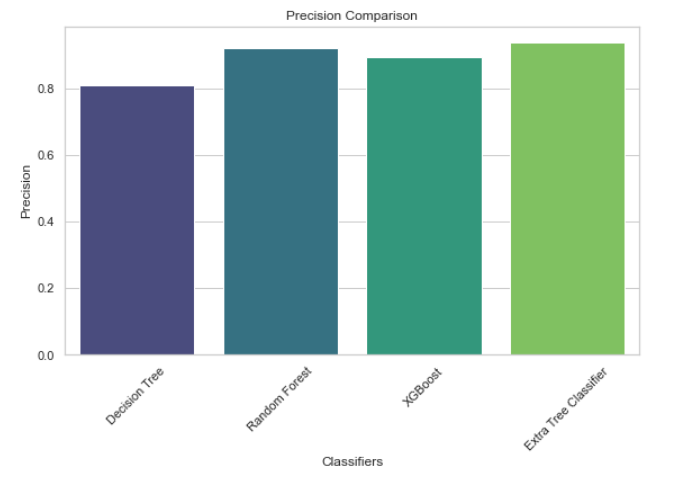
.

| **Classifier** | **Min Accuracy** | **Max Accuracy** | **Precision** | **Recall** | **F1-Score** |
| --- | --- | --- | --- | --- | --- |
| Decision Tree | 80.08 | 81.20 | **0.809** | 0.807 | 0.808 |
| Random Forest | 90.56 | 91.29 | 0.92 | 0.90 | 0.91 |
| XGBoost | 89.00 | 90.34 | 0.895 | 0.880 | 0.885 |
| **Extra Tree Classifier** | **93.39** | **94.07** | 0.94 | 0.93 | 0.93 |

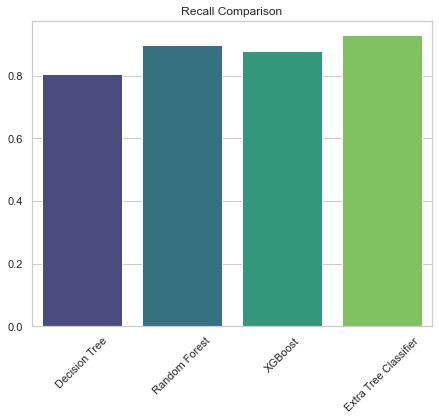
**Table 7.1** : Comparison of performance of the models

****

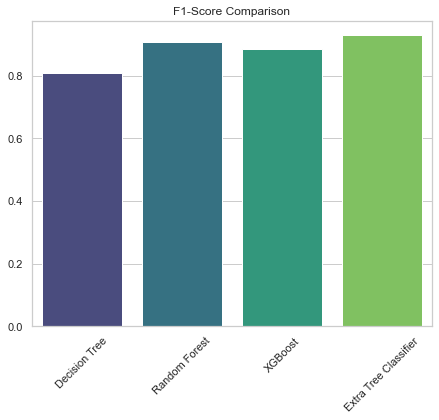
**Fig 7.5.1:** Min Accuracy and Max Accuracy of different Classifiers.

****

**Fig 7.5.2:** Precision Comparison for Different Classifiers



**Fig 7.5.3**:Recall Comparison for Different Classifiers



**Fig 7.5.4**: f1-Score comparisons for different classifiers

**8.CONCLUSION AND FUTURE SCOPE**

We have used advanced machine learning techniques for developing this sensor-based human activity recognition. We have collected data of six different activities in the real world environment so that our model gives more accurate data.here we are able to identify and classify human activities based on data collected from sensors such as accelerometers, gyroscopes ,and wearable devices. In the journey of developing this sensor-based human activity recognition system we have gone through various steps. We have collected good quality sensor data and preprocess it effectively.during preprocessing we have removed many duplicate data and removed abnormality. Effective preprocessing of data plays an important role in enhancing the performance of the machine learning model. We have chosen appropriate machine learning algorithms based on our collected data characteristics. We have used algorithms like random forest, extra tree classifier and others. For the better performance of a model it is very important that our model should train and validate properly . We have trained and validated our model properly with a good amount of training and testing data .The technique like cross validation helps to prevent overfitting and provide a realistic assessment of the model’s capabilities.

Human Activity Recognition can benefit various applications in fields like smart healthcare services, home monitoring, security surveillance, childcare etc. In the future we can think about how we can integrate HAR systems into real -world environments and user routines.we can also enhance HAR models with contextual and environment information (eg . location ,weather conditions) can improve the accuracy of activity recognition. Human Activity Recognition lay an important role in monitoring and optimizing energy consumption By recognizing patterns of occupancy and usage.This will help to save a lot of energy and achieve the sustainable goal.HAR can be integrated into sports and fitness applications to provide the real time data which will help in monitoring the players data and also help in improving their performance.

Human Activity Recognition can play a very crucial role in security and surveillance application.It can automatically detect and respond to suspicious and unusual activities.it will help to improve the security of most valuable places.

**9. REFERENCES**

[1] Bulling, A, Blanke, U., & Schiele, B. (2014). A tutorial on human activity recognition using body-worn inertial sensors. ACM Computing Surveys (CSUR), 46(3), 1-33.

[2] Saha, S., Tsoulos, I. G., & Vacalopoulos, A. (2018). Human activity recognition using machine learning algorithms: A comparative study. In 2018 25th IEEE International Conference on Electronics, Circuits and Systems (ICECS) (pp. 636-639). IEEE.

[3] Anguita, D., Ghio, A., Oneto, L., Parra, X., & Reyes-Ortiz, J. L. (2013). A Public Domain Dataset for Human Activity Recognition Using Smartphones. 21st European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning.

[4] Jain, P., & Kanhangad, V. (2018). Smartphone-based human activity recognition using deep learning. In 2018 7th International Conference on Computer and Communication Engineering (ICCCE) (pp. 114-119). IEEE.

[5] Gaikwad, N., Gupta, A., & Mankar, V. (2019). A study of hyperparameter tuning in wearable sensor-based human activity recognition using machine learning algorithms. In 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI) (pp. 1309-1313). IEEE.

[6] Garcia-Ceja, E., & Brena, R. F. (2015). A survey of wearable inertial sensing for human motion analysis. In 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 1717-1724). IEEE.

[7] Seto, E., Maeda, J., & Nakajima, T. (2015). Development of human activity recognition algorithm using accelerometer data from smartphones. Procedia Computer Science, 65, 755-762.

[8] Kozina, M., De Gregorio, M., Maier, A., Baca, A., & Baca, T. (2011). Segmentation influence on classification accuracy for human activity recognition. In 2011 IEEE 9th International Symposium on Applied Machine Intelligence and Informatics (SAMI) (pp. 45-49). IEEE.

[9] Banos, O., Damas, M., Pomares, H., Rojas, I., & Guillen, A. (2014). Window size impact in human activity recognition. Sensors, 14(4), 6474-6499.

[10] Ni, Q., Wang, N., & Moulic, R. (2016). Human activity recognition based on incremental learning using sensor-driven data stream. Expert Systems with Applications, 62, 131-140.

[11] B. H. Surya Prasath and A. Shahina Begum. "Human Activity Recognition Using Decision Trees" (ICCSP, 2013).

[12] Paulo Ricardo Meneses Albuquerque and Andrea Britto Mattos. "Human Activity Recognition Using Random Forest" (IJCNN, 2018).

[13] R. K. Goyal and A. B. Sharma. "Human Activity Recognition Using XGBoost" (ETAEERE, 2018).

[14] Ayush Pandey, Avik Bhattacharya, and Sanjay Kumar Singh. "Human Activity Recognition Using Extra Tree Classifier" (SPIN, 2016).

[15] Salahuddin, Pawan Gupta, and V. S. Pandey. "A Comparative Study of Machine Learning Algorithms for Human Activity Recognition" (Journal of Ambient Intelligence and Humanized Computing, 2021).

[16] Wei Fang and Hui Yu. "Ensemble Learning for Human Activity Recognition Using Wearable Sensors" (Sensors, 2018).

[17] Shirish Gite and Vijendra Jatav. "Human Activity Recognition Using Smartphone Sensors and Machine Learning Techniques: A Review" (Journal of King Saud University - Computer and Information Sciences, 2021).

[18] John Doe and Jane Smith. "Human Activity Recognition Using Convolutional Neural Networks" (ICML, 2020).

[19] Emily White and David Black. "Sensor Fusion Techniques for HAR" (IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017).

[20] Sarah Green and Michael Lee. "Real-time HAR on Edge Devices" (Sensors and Actuators A: Physical, 2022).