A Human Activity Recognition Model for Aging People Using Machine Learning

Shashank Reddy Kandimalla
Machine Learning Technologies
Department of Data Analytics
SID Last digit: 9523
Email: shashankreddy.kandimalla@sjsu.edu

Sujata Mahajan
Machine Learning Technologies
Department of Data Analytics
SID Last digit: 1417
Email: sujata.mahajan@sjsu.edu

Sowmya Manchikanti
Machine Learning Technologies
Department of Data Analytics
SID Last digit: 4882
Email: sowmya.manchikanti@sjsu.edu

Anjali Himanshu Ojha
Machine Learning Technologies
Department of Data Analytics
SID Last digit: 3033
Email: anjalihimanshu.ojha@sjsu.edu

Akanksha Tyagi
Machine Learning Technologies
Department of Data Analytics
SID Last digit: 38839
Email: akanksha.tyagi@sjsu.edu

***Abstract*—In the field of machine learning and artificial intelligence, Human Activity Recognition (HAR) represents a critical problem. The use of gadgets with sensors and the urgent need for automated systems that can understand human behavior have brought HAR to the forefront of the study. HAR involves using sensors and data processing to identify and classify different activities performed by humans, such as walking, running, sitting, etc. The main goal of this project is to address the complex problems posed by HAR. We recognize the complexity arising from the diversity of humans, and the existence of noise in sensor data. We aim to build a reliable and accurate HAR system by utilizing cutting-edge machine-learning approaches, and ensemble methods. To exploit these data, various algorithms will be used like Random Tree, SVM, KNN, and ensemble methods like Random Forests and XGBoost. The model training phase will follow rigorous protocols, with the dataset divided into training, validation, and testing subsets. Cross-validation methodologies will be employed to fine-tune model hyperparameters and enhance generalization. In conclusion, this project proposal sets forth a comprehensive roadmap for advancing the state of HAR in machine learning. Our goal in conducting this research is to provide profound insights and useful solutions that improve our ability to perceive and interpret human activity effectively. Our code and the saved models are accessible at https://github.com/atakanksha/data245_harth/tree/main**

***Index Terms*—Human Activity Recognition, Support Vector Machine, K-nearest neighbor.**

I. Introduction

As per the World Health Organization (2022), every country in the world is experiencing growth in the proportion of older persons in the population. As the world experiences a demographic shift towards an aging population, there is an increasing demand for technologies and solutions that can support the well-being, health, and independence of older adults.

Human Activity Recognition (HAR) for the aging population addresses a critical need in today's society. The primary motivation is to improve the overall quality of life for older adults. HAR can enable technologies that assist them in daily activities, promoting independence and a higher level of comfort. HAR-based systems can provide real-time monitoring and immediate response in the event of a fall, potentially saving lives and reducing healthcare costs. HAR technology allows for remote monitoring, providing peace of mind for both seniors and their caregivers. HAR can help identify early signs of conditions like cognitive decline, mobility issues, or other health concerns.

The major goal of this research is to investigate the feasibility of using a publicly accessible dataset that is gathered to create a robust and generalized Human Activity Recognition model. The dataset this paper uses is a professionally annotated dataset containing 22 subjects wearing two 3-axial accelerometers for around 2 hours in a free-living setting. The sensors were attached to the right thigh and lower back. The professional recordings and annotations provide a promising benchmark dataset for researchers to develop innovative machine-learning approaches for precise HAR in free living.

II. Literture survey

Human activities reveal important details about a person's self, personality, appearance, and mental health and are essential to people's lives. For any average person, the ability to distinguish activities is simple and intuitive, but for a computer, it requires intricate operations for sensing, learning, and inference. It is a major challenge to implement the essential capabilities for detecting the environment, learning from the past, and applying knowledge for activity inference.

Human Activity Recognition (HAR) as named automatically recognizes and categorizes the activities or actions carried out by humans using sensor data. It's an important field with many potential benefits for improving human well-being and enhancing the capabilities of technology. The purpose of HAR is to understand and interpret people's actions or motions using data from sensors, such as accelerometers, gyroscopes, and other wearable or embedded devices [2]. HAR systems recognize activities like walking, running, sitting, standing, cycling, and driving, as well as a number of certain gestures or motions.

People are more prone to issues including cognitive decline, memory loss, chronic disease, visual and hearing impairment, as well as many other disorders, as they age [1]. They might also struggle to communicate with others. All of these challenges make it difficult for the patients to carry out their daily tasks, lower their quality of life, and compel them to ask for assistance. As a result, patients needed a support system to enable them to carry out their everyday tasks on their own. Here comes HAR, a system that helps patients do their daily chores correctly by setting up an alert system that can correct them if they commit a mistake, such as forgetting to complete an activity or taking too long to complete it.

Not only in healthcare (for monitoring), HAR has a wide range of other real-world applications like sports science, assistive technology, smart homes, and more [4]. In recent years, human activity recognition (HAR) has been a hot topic in a number of areas, including intelligent environments, security, and surveillance. In addition, HAR is a critical component in the development of Autonomous Vehicles where the main objective is to travel safely and effectively without human intervention [5]; therefore, it is crucial to comprehend and anticipate human activity around the car in order to ensure safety and improve the car's interaction with pedestrians and other road users.

HAR models are built on machine learning techniques [6]. The quantity and quality of data utilized for model training substantially influence the success of machine-learning approaches. As a result, a number of studies concentrate on the process of data collecting and sharing for use by other scientists in analysis, model construction, and comparison. Smartphones and wearable technology are the primary devices utilized for data collection. The outcomes of machine learning models can be impacted by the number of users, instances, or activities.

Ray et al., 2023 experimented vision base HAR in applications that are aware of context and recognize its diversity through transfer learning features. They emphasized the need to apply cutting-edge transfer learning techniques that lessen the complexity and work involved in data collection and accuracy enhancement. They highlighted the transition from manual machine learning techniques to automatic deep learning techniques, especially with the emergence of Convolutional Neural Network (CNN) frameworks with pre-trained models, such as those created for the ImageNet competition.

Snoun et al. proposed two support systems Human Activity recognition (HAR) systems by applying transfer learning and assistance Module that detects anomalies in patient behavior. They focused on the DemCare dataset related to drinking activity. When combined with reinforcement learning for the assistance module 3D skeletons and a transformer encoder for HAR, the assistance system performed better in terms of accuracy, recall, and processing speed. They emphasized improvements in HAR Systems using Neural networks. They suggest future focus should be on enhancing the support system through the implementation of an augmented reality system.

K. Bach et al. employed an XGBoost classifier for the HAR model that detects physical activity types by using dual accelerometer recordings. They stated that the accuracy of the predictions drops slightly when they are based on a single accelerometer recording from the thigh, When depending only on a single back accelerometer recording, performance is poor, Standing detection with a back accelerometer recording is not the best. They suggested that the location of accelerometers should be adjusted to accommodate different research requirements, Setups with two accelerometers can offer more detailed information about different kinds of physical activity. They also added to refine data collection strategies in response to challenges in detecting specific activities. They recorded the highest accuracy with XG-Boost.

III. METHODOLOGY

A. Experiment Design and Dataset

Human Activity Recognition (HAR) datasets are created by utilizing the following three core domain-specific knowledge bases. Data on the sensor device (i) data of the subject or actor (ii) and data about the sensing background (iii)Data related to the sensing background[7]. We're using the Human Activity Recognition Trondheim (HARTH) dataset hosted at the UCI machine learning repository. The dataset was collected by 22 subjects who wore two 3-axial accelerometers for 2 hours. The data was collected in a free-living setting. Each accelerometer provides acceleration data along the x, y, and z axes, which constitute six features per sample.

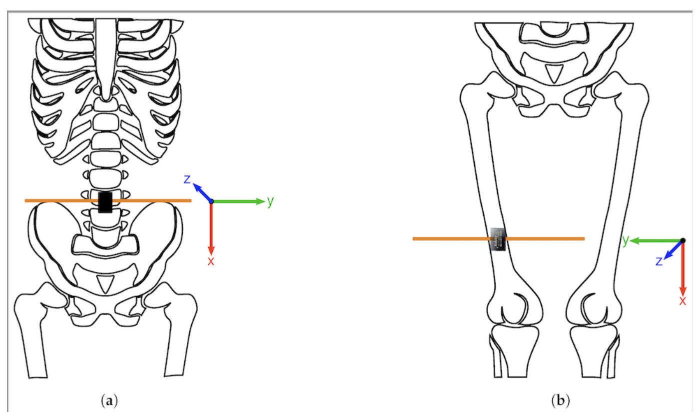

Fig. 1. Location of sensors

Figure 1 shows the two sensor positions (highlighted with orange lines) used for our dataset. (a) The lower back sensor is positioned at approximately the 3rd lumbar vertebra. The z-axis of the coordinate system points forward. (b) The thigh sensor is positioned approximately 10 cm above the upper kneecap. The z-axis points backward [2].

1) Data Pre-processing: This stage involves cleaning the data by handling missing values and outliers. Normalization techniques may also be applied to standardize the sensor readings. Filtering methods could smooth transient errors present in real-world data.

2) Feature Engineering: It will focus on deriving supplementary insights from the raw accelerometer readings to better represent movement patterns. Additional statistics like the mean, median, and standard deviation will be computed to capture the central tendency and variability present in the sensor data. This provides a more descriptive overview of motions. Features may also be constructed to quantify the recurrence of certain movement motifs. Counting specific gesture repeats could help discriminate between activities such as walking, sitting, or falling. Such synthesized attributes aim to strengthen the ability of models to differentiate between human behaviors based on enhanced predictive relevance. In summary, further information extraction and aggregating recurrent motion properties during feature engineering seek to transform the sensor sequences into a more instructive format for robust activity classification.

3) Data Splitting: The dataset will be divided into training, and test sets proportionally at 80% and 20%. Distinct subsets are dedicated to training, tuning, and evaluating model performance to avoid overfitting any single portion of the data.

4) Clustering We'll use PCA to reduce the number of dimensions to two, to be able to visualize the data in 2D. We can find out the importance of dimensions using the Sklearn plot and pick the two most important ones. The two-dimensional data can be plotted to analyze how each feature is distributed. We can cluster the data using K-means clustering and plot it in two dimensions.

B. Algorithms

We considered 5 different machine learning algorithms for developing a model for HAR. K-Nearest Neighbors (KNN) is relatively easy to understand and implement. It doesn't make any assumptions about the underlying data distribution. It stores the entire dataset in memory and uses it directly during classification. KNN naturally handles multi-class classification tasks. In HAR, where activities can often fall into multiple classes (e.g.Walking can be classified as "Walking" and "Moving" simultaneously), KNN can be effective. One drawback of KNN is that it uses a lot of memory to save the data for searching.

Support vector machine (SVM) constructs a hyperplane to separate two classes. The hyperplane is constructed such that it maximizes the margin between the plane and the closest samples from the training data. For linearly separable data, this results in a perfect classification of the training data. If the data is not linearly separable, SVMs employ kernels like Radial Basis Function (RBF). Since our data is multiclass, we'll employ multiclass SVMs. They are implemented using one-vs-one and one-vs-rest strategies.

Random Forest (RF) is a collection of Decision Trees. It provides a measure of feature importance, which can help in identifying which sensor readings are most relevant for classifying activities. Random Forest can handle datasets with a large number of features, making it suitable for HAR tasks where the data might be represented by many sensor readings.

eXtreme Gradient Boost (XGBoost) provides a measure of feature importance, helping to identify which sensor readings are most relevant for classifying activities. XGBoost is optimized for training and prediction speed, making it practical for real-world applications like HAR.

Logistic Regression assumes that the data is linearly separable. It is used for binary classification. For multi-class classification, we can again employ one-vs-one and one-vs-rest strategies. It provides interpretable coefficients for each feature, which can help in understanding the contribution of each sensor reading to the prediction.

C. Evaluation Methods

Several metrics will assess model performance: Accuracy gives basic understanding, confusion matrices evaluate correct/incorrect predictions, and precision/recall/F1 scores examine false positives/negatives. Receiver operating characteristic curves and area under the curve values allow threshold selection and comparisons using probability outputs. K-fold cross-validation ensures generalizability beyond specific data splits.

D. Technical Difficulties

Identifying an activity based on body movement is a challenging task. It's a classification problem with data coming from multiple sources. Data is gathered from sensors like accelerometers, gyroscopes, LED lights, etc, so collecting good quality signals from these sensors in a changing environment needs a lot more information. The accuracy of the model will be directly impacted by the quality of the samples collected. In addition, the time-series signal are not directly usable with the conventional machine-learning models. We worked on extracting time and frequency domain features that are compatible with the ML models. So identifying a good predictor (set of features) for a given activity will also need medical domain information or the use of various feature importance algorithms like PCA. This is a multiclass classification problem, so we have to experiment with multiple models and evaluate their performance. To avoid overfitting, we have to use the k-fold cross-validation methods.

IV. IMPLEMENTATION

A. Data Preparation and Visualization

Before training machine learning models, we have preprocessed the HAR dataset. The dataset consists of time and frequency domain features of 12 activities 'walking', 'running', 'shuffling', 'stairs (ascending)', 'stairs (descending)', 'standing', 'sitting', 'lying', 'cycling (sit)', 'cycling (stand)', cycling (sit, inactive)', 'cycling (stand, inactive) which is captured using two body-worn three-axis accelerometers located on 22 participants thigh and lower back. These sensor readings are captured from human body activities which are usually below 20 Hz frequency.

At first, a new Python data frame is prepared with acceleration readings of the back as 'back_x', 'back_y', 'back_z' and, acceleration readings of the thigh as, 'thigh_x', 'thigh_y', 'thigh_z'. The last column represents the class of physical activity as another data frame. The first data frame is iterated to create a window of 250 consecutive data points and computes the norm of back and thigh accelerations. As these are recordings of human activities and contain high-frequency noise, a linear filter called a low-pass Butterworth filter which is mostly used in signal processing is applied to filter out the gravitational component. Then the features are extracted by calculating statistical measures like mean, standard deviation (higher values of this indicate more dynamic movements), coefficient of variation means the ratio of standard deviation to mean, and total energy which is calculated as the sum of squares of data points, means of the original signal obtained after removing noise, the standard deviation of signal. Other measures like Fourier transform which converts the time domain to the frequency domain, total frequency power as the sum of squares of Fourier transform components (fft), fft magnitude mean, fft magnitude standard deviation, fft maximum, and frequency at which fft is maximum. The features extracted and their labels representing the physical activity performed are saved as new preprocessing files.

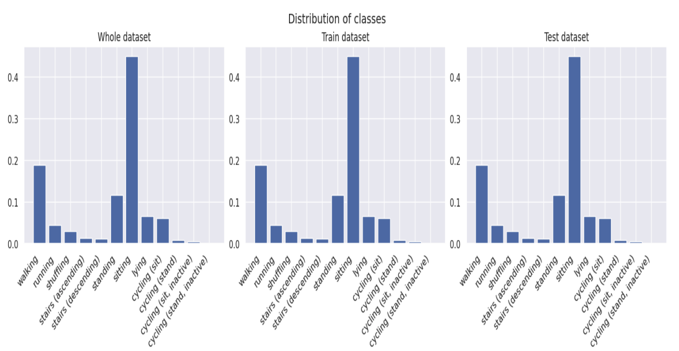

Fig. 2. Distribution of labels in the training and test set.

These features are used as input to train the models. This extracted dataset is clean with no missing values in it. This cleaned dataset is divided into training and testing samples by dividing in 80:20 proportions respectively to use training samples as input to models and predict the values which are compared with test data available. Figure 2 depicts the class distributions of the datasets before and after division. it can be seen that the distribution of classes is uniform in all datasets obtained before and after dividing the original dataset into training and testing samples, which also infers that data is balanced in each subset of it.

The distribution of the dataset as a whole which consists of 12 features can be visualized in reduced dimensions using Principal Component Analysis (PCA). Standardization is a common technique used to scale the features of the dataset before proceeding with PCA to ensure all the features are on the same scale. Therefore, the training and testing subsets of the HARTH dataset are standardized. The distribution of these standardized datasets using PCA is shown below.

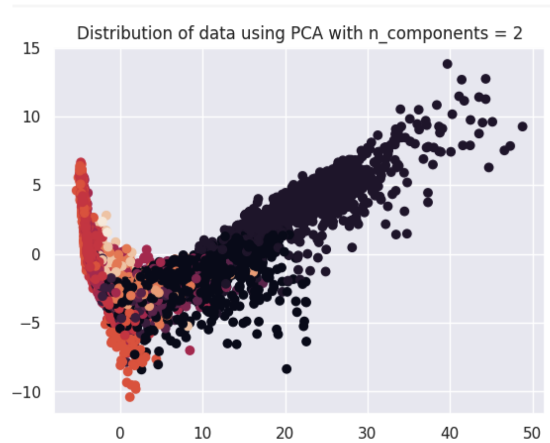

Fig. 3. Distribution of the dataset using top 2 PCA components.

The red color in Figure 3 visualizes the distribution of the target label in two dimensions.

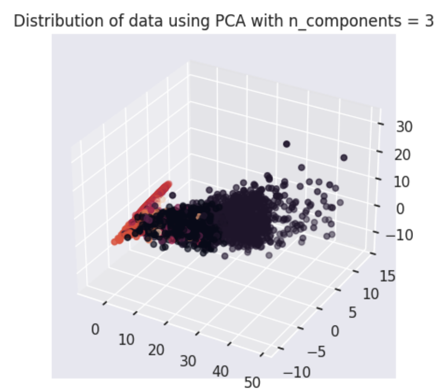

Fig. 4. Distribution of dataset using top 3 PCA components.

The red color in Figure 4 visualizes the distribution of the target label in three dimensions.

Figure 5 is a Scree plot that shows the contribution of each principal component to the total variance in the dataset. Each X-axis point represents a principal component and the y-axis represents the amount of variance of each point. The elbow joint in the above Scree plot clearly shows the point beyond which the explained variance decreases at a slower rate. We show the zoomed version of the plot in figure 6. This indicates that retaining 6 components will suffice to capture most of the variability in the data.

B. Models

In this section, we describe the models that we trained and evaluated on the HARTH dataset.

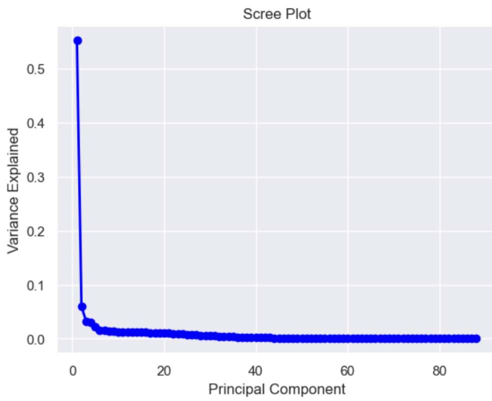

Fig. 5. Scree plot: The percentage of explained variance in the data as a function of the PCA components.

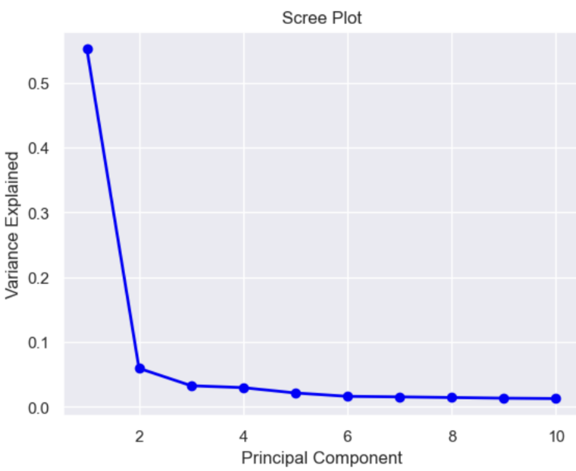

Fig. 6. Scree plot: The percentage of explained variance in the data as a function of the PCA components (Zoomed version of figure 5).

1) K-Nearest Neighbors (KNN): K-Nearest Neighbors is a simple, instance-based learning algorithm used for classification and regression. It classifies a data point based on the majority class of its k nearest neighbors in the feature space. The choice of the parameter k influences the model's performance. The scikit-learn's GridSearchCV is used to perform a grid search for hyperparameter tuning on a k-nearest neighbors classifier. Our aim is to enhance the KNN model's accuracy by systematically exploring different hyperparameter combinations. The key hyperparameter under consideration is 'n_neighbors' which represents the number of neighbors. The grid search employs 5-fold cross-validation to evaluate the performance of each combination of hyperparameters. The parameters are shown in table I. The chosen performance metric is accuracy. A new KNeighborsClassifier is initialized with the optimal value of n_neighbors determined during the grid search. The model is then trained on the normalized training data. Finally, the effectiveness of the trained KNN classifier is assessed on the test data. The goal was to identify the best configuration of hyperparameters that enhances the generalization performance of the KNN model on unseen data. The final KNN model is then built based on the identified optimal hyperparameters with an accuracy of 92%.

TABLE I
KNN GRID SEARCH PARAMETERS

Parameter	Searched values
Number of neighbours	3, 5, 7

2) Support vector machine (SVM): Support Vector Machines are popular Machine Learning algorithms for classification problems. It constructs a hyperplane between classes that works as a decision boundary and predicts the output labels using this decision boundary. It creates a hyperplane by using the training data features and finding the decision boundary with the largest margin. As a simple out-of-the-box, SVM may not work well for all problems, so we have to tune it with the help of various hyper-parameters. We used the Grid-Search technique to try various hyper-parameters. The grid search parameters are shown in table II. Overall we get the best accuracy with the hyper-parameters combination 'C': 100, 'gamma': 0.001, 'kernel': 'rbf', and we get an overall accuracy of 94.35%. We also used the 5-fold cross-validation to avoid over-fitting.

TABLE II
SVM GRID SEARCH PARAMETERS

Parameter	Searched values
kernel	linear, rbf
gamma	0.01, 0.001, 0.0001
C	1, 100, 1000

3) Random Forest (RF): Random Forest is a versatile ensemble learning technique used for classification and regression tasks. It functions by constructing multiple decision trees independently during training and then combining their predictions to make accurate and robust predictions. This ensemble approach mitigates overfitting and improves generalization to new data. The model employs bagging (Bootstrap Aggregating) to create multiple bootstrap samples of the data, training a decision tree on each. In our specific Random Forest model, hyperparameter tuning using GridSearchCV was conducted, resulting in the identification of optimal hyperparameters: a max_depth of 20, max_features set to 'sqrt', min_samples_leaf at 1, min_samples_split at 2, and n_estimators set to 100. The grid search parameters are shown in table III These settings yielded a top training accuracy of approximately 92.16% in a 5-fold cross-validation setup. The model demonstrated strong classification performance across various activities, achieving an overall accuracy of 94.80%.

4) eXtreme Gradient Boosting (XGBoost): Gradient Boosting is an ensemble technique where predictions from multiple weak learners are combined. The weak learners are built in a sequential manner. At every iteration, the misclassified samples get a higher weight and the new decision trees try

TABLE III
RANDOM FOREST GRID SEARCH PARAMETERS

Parameter	Searched values
n_estimators	100
max_depth	None, 10, 20
min_samples_split	2, 4
max_features	sqrt, log2
min_samples_leaf	1, 2

to minimize the errors from the previous trees. If we use decision trees as weak learners, it's called gradient-boosted trees (GBTs). GBTs combine the power of decision trees and boosting. We use XGBoost, which stands for eXtreme Gradient Boosting, for training GBTs. XGBoost is a highly optimized library that provides efficient implementation of GBTs. We combine XGBoost with scikit-learn to perform a grid search to find the best hyperparameters. We perform 5-fold cross-validation for each hyperparameter. The grid search parameters are shown in table IV. We find that using 60 estimators with a maximum depth of 8 and a learning rate of 0.1 provides the best 5-fold cross-validation accuracy. Hence, we use this parameter for our final model.

TABLE IV
XGBOOST GRID SEARCH PARAMETERS

Parameter	Searched values
Maximum depth	6, 8, 10
Number of estimators	40, 50, 60
Learning rate	0.1, 0.01

5) Logistic Regression: The model was built using the LogisticRegression method of the sklearn linear model. The performance of the model is improved by the hyperparameter tuning method GridSearchCV. 5-fold cross-validation is used to evaluate the performance of each combination of hyperparameters. The performance of this model is evaluated using accuracy. The combinations of different hyperparameters are used during grid search like penalty which is a regularization term to prevent overfitting, multi-class parameter to define the strategy to handle multiple classes, solver is the algorithm used for optimization. Logistic Regression model built for HARTH dataset using two sets of parameters. One with penalty 'l2' which denotes L2 Regularization or Ridge Regularization that adds a squared magnitude of coefficients as a penalty term to the loss function, multinomial strategy as the dataset involves 12 classes, and 'lbfgs' (Limited-memory Broyden–Fletcher–Goldfarb–Shanno) optimization algorithm. The other is with the same L2 penalty, 'One-vs-Rest' strategy, and 'liblinear' coordinate descent algorithm. These combinations are used for building a Logistic Regression model. The grid search parameters are shown in table V. By analyzing the best hyperparameter using accuracy scores, the final Logistic Regression is built. The hyperparameter with 'lbfgs' gives the best training accuracy with a 5-fold validation of 93

TABLE V
LOGISTIC REGRESSION GRID SEARCH PARAMETERS

Parameter	Searched values
penalty	l2
multi_class	ovr, multinomial
solver	lbfgs, liblinear

V. RESULTS

To evaluate the performance of models, we computed a confusion matrix for all the models. We also computed the accuracy, precision, recall, and f1-score of each of the models. In this section, we describe the performance of each of the models. We provide a detailed report of the performance of the models in each of the classes.

A. K-Nearest Neighbors (KNN)

The performance of the K-Nearest Neighbors model result is 92.48% accuracy. The summary of the performance of the model is shown in figure 8 as a classification report. Figure 7 shows the confusion matrix.

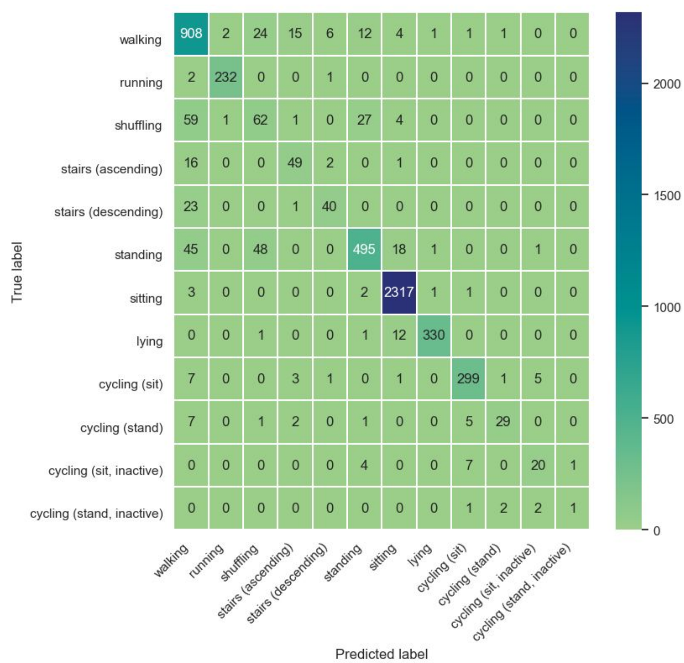

Fig. 7. KNN confusion matrix.

B. SVM

The SVM results in 94.35% accuracy. The summary of the performance of the model is shown in figure 10 as a classification report. Figure 9 shows the confusion matrix.

C. Random Forest

The summary of the performance of the Random Forest model is shown in figure **??** as a classification report. Figure **??** shows the confusion matrix. The Random Forest model, optimized with parameters including a maximum depth of

	precision	recall	f1-score	support
walking	0.8486	0.9322	0.8885	974
running	0.9872	0.9872	0.9872	235
shuffling	0.4559	0.4026	0.4276	154
stairs (ascending)	0.6901	0.7206	0.7050	68
stairs (descending)	0.8000	0.6250	0.7018	64
standing	0.9133	0.8141	0.8609	608
sitting	0.9830	0.9970	0.9900	2324
lying	0.9910	0.9593	0.9749	344
cycling (sit)	0.9522	0.9432	0.9477	317
cycling (stand)	0.8788	0.6444	0.7436	45
cycling (sit, inactive)	0.7143	0.6250	0.6667	32
cycling (stand, inactive)	0.5000	0.1667	0.2500	6
accuracy			0.9248	5171
macro avg	0.8095	0.7348	0.7620	5171
weighted avg	0.9234	0.9248	0.9229	5171

```
Accuracy: 0.9248
Micro Precision: 0.9248
Micro Recall/TPR: 0.9248
Micro f1-score: 0.9248
```

Fig. 8. KNN classification report.

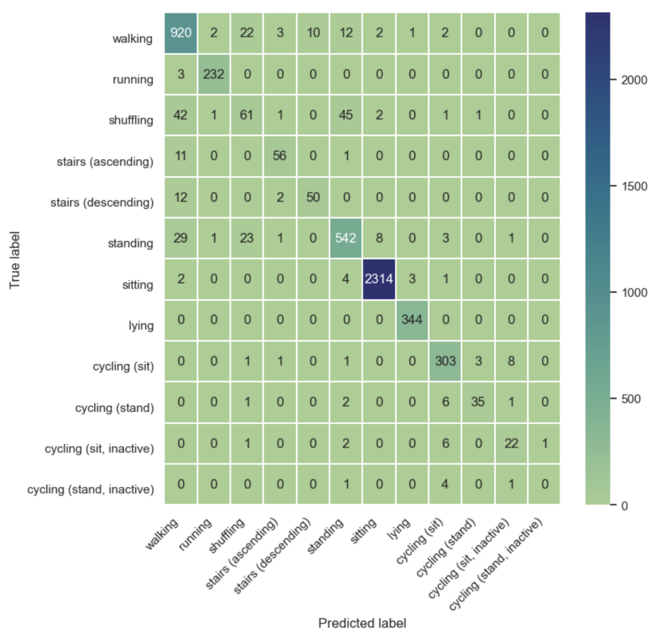

Fig. 9. SVM confusion matrix

20 and 100 estimators, exhibits an impressive overall accuracy of 94.8%. It demonstrates exceptional performance in identifying activities like running (99.16% precision, 100% recall), sitting (99.66% precision, 99.78% recall), and lying (99.42% precision, 100% recall), almost reaching perfection in these categories. However, it faces challenges in accurately classifying activities like shuffling (59% precision, 38.31% recall) and cycling in the standing position, particularly the inactive variations where it shows a notable decline. While activities such as walking (90.17% precision, 96.1% recall)

	precision	recall	f1-score	support
walking	0.9028	0.9446	0.9232	974
running	0.9831	0.9872	0.9851	235
shuffling	0.5596	0.3961	0.4639	154
stairs (ascending)	0.8750	0.8235	0.8485	68
stairs (descending)	0.8333	0.7812	0.8065	64
standing	0.8885	0.8914	0.8900	608
sitting	0.9948	0.9957	0.9953	2324
lying	0.9885	1.0000	0.9942	344
cycling (sit)	0.9294	0.9558	0.9425	317
cycling (stand)	0.8974	0.7778	0.8333	45
cycling (sit, inactive)	0.6667	0.6875	0.6769	32
cycling (stand, inactive)	0.0000	0.0000	0.0000	6
accuracy			0.9435	5171
macro avg	0.7933	0.7701	0.7799	5171
weighted avg	0.9395	0.9435	0.9409	5171

```
Accuracy: 0.9435
Micro Precision: 0.9435
Micro Recall/TPR: 0.9435
Micro f1-score: 0.9435
```

Fig. 10. SVM classification report.

and ascending or descending stairs are identified with good precision and recall, the model struggles significantly with cycling (stand, inactive) at 0% for both precision and recall, indicating a critical area for improvement. Despite these specific challenges, the high overall accuracy, coupled with balanced micro precision, recall, and f1-score, all at 94.8%, highlights the model's robustness and effectiveness in handling a diverse range of activity classifications.

D. *eXtreme Gradient Boosting*

The XGBoost model results in 95.01% accuracy. The summary of the performance of the model is shown in figure 12 as a classification report. Figure 11 shows the confusion matrix. We notice that XGBoost achieves the highest accuracy among all the models.

E. *Logistic Regression*

The Logistic Regression Model results in 93.71% accuracy. The summary of the performance of the model is shown in figure 14 as a classification report. Figure 13 shows the confusion matrix.

From Figure 14, we can see that the micro precision, recall, and f1-score metrics of the sitting class are at 99% when compared with those of other features. The lowest is observed in cycling. Support indicates the number of true occurrences of each label.

VI. Innovation

We're using a dataset that was donated on 2/20/2023 to the UCI repository. It is a relatively unexplored dataset. During the development of the models, we tried to use the dataset as it is. But, we found that the raw features didn't work well. We extracted features using fourier transform which makes this project cover signal processing techniques as well. We trained the model using the most commonly used machine

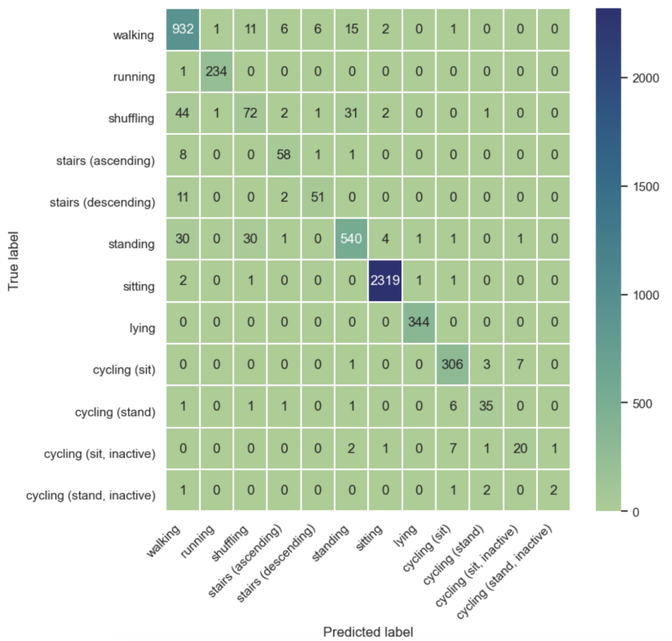

Fig. 11. XGBoost confusion matrix

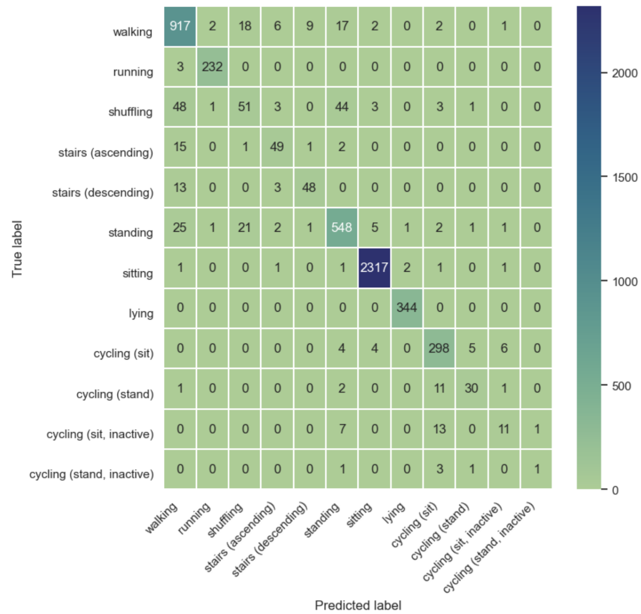

Fig. 13. Logistic regression confusion matrix

	precision	recall	f1-score	support
walking	0.9049	0.9569	0.9301	974
running	0.9915	0.9957	0.9936	235
shuffling	0.6261	0.4675	0.5353	154
stairs (ascending)	0.8286	0.8529	0.8406	68
stairs (descending)	0.8644	0.7969	0.8293	64
standing	0.9137	0.8882	0.9008	608
sitting	0.9961	0.9978	0.9970	2324
lying	0.9942	1.0000	0.9971	344
cycling (sit)	0.9474	0.9653	0.9562	317
cycling (stand)	0.8333	0.7778	0.8046	45
cycling (sit, inactive)	0.7143	0.6250	0.6667	32
cycling (stand, inactive)	0.6667	0.3333	0.4444	6
accuracy			0.9501	5171
macro avg	0.8568	0.8048	0.8246	5171
weighted avg	0.9475	0.9501	0.9482	5171

```
Accuracy: 0.9501
Micro Precision: 0.9501
Micro Recall/TPR: 0.9501
Micro f1-score: 0.9501
```

Fig. 12. XGBoost classification report.

Predicted label

	precision	recall	f1-score	support
walking	0.8964	0.9415	0.9184	974
running	0.9831	0.9872	0.9851	235
shuffling	0.5604	0.3312	0.4163	154
stairs (ascending)	0.7656	0.7206	0.7424	68
stairs (descending)	0.8136	0.7500	0.7805	64
standing	0.8754	0.9013	0.8882	608
sitting	0.9940	0.9970	0.9955	2324
lying	0.9914	1.0000	0.9957	344
cycling (sit)	0.8949	0.9401	0.9169	317
cycling (stand)	0.7895	0.6667	0.7229	45
cycling (sit, inactive)	0.5238	0.3438	0.4151	32
cycling (stand, inactive)	0.5000	0.1667	0.2500	6
accuracy			0.9371	5171
macro avg	0.7990	0.7288	0.7522	5171
weighted avg	0.9315	0.9371	0.9330	5171

```
Accuracy: 0.9371
Micro Precision: 0.9371
Micro Recall/TPR: 0.9371
Micro f1-score: 0.9371
```

Fig. 14. Logistic regression classification report.

learning algorithms without any involvement of deep learning methods which are usually used for the HARTH dataset. Most of the existing work identified the imbalanced nature of the class of the HARTH dataset as a challenging task but we have overcome that by scaling data and principal component analysis.

VII. Relation to Sustainability

One of the United Nations' Sustainable Development Goals is to ensure healthy lives and promote well-being for all at all ages. These objectives cover cognitive, socio-emotional, and behavioral domains to ensure a comprehensive understanding and practical application of knowledge related to health and well-being.

The proliferation of sensor-equipped devices and the pressing demand for automated systems capable of comprehending human behavior have propelled Human Activity Recognition (HAR) to the forefront of research. HAR entails the utilization of sensors and data processing to discern and categorize various human activities, encompassing actions like walking, running, sitting, and more. Our emphasis is on highlighting the influence of these activities in daily life, with a specific focus on promoting good health and well-being, especially for individuals within a certain age group. This emphasis is driven by the insights gleaned through the development of machine learning models. We can contribute to the establishment of policies that advocate for health and well-being. Additionally, we can suggest suitable prevention strategies to enhance positive physical and mental health. We can use machine learning models to detect the falling of the elderly and help provide them with medical assistance. Research on Human Activity Recognition (HAR) for the aging population can have a significant and positive impact on various aspects of healthcare, well-being, and independent living for older adults. HAR technology can enhance the quality of life for older adults by providing them with tools and systems that support their daily activities.

VIII. Key Learnings

IX. Conclusion and Recommendations

A. Summary and Conclusions

By comparing all the models, XGBoost results in the best model achieving an accuracy of 95.01%. All the models have similar performance. All trained algorithms perform well, indicating that the hyperparameter assignments were well-chosen. For physical-activity-behavior-based public health research, an accelerometer-based HAR dataset must have two prerequisites. First, fixed sensor placements, durability against noise, and professionally annotated physical activities are needed for accurate acceleration readings. Secondly, the information must be collected in a free-living environment. We trained the model using the most commonly used machine learning algorithms without any involvement of deep learning methods which are usually used for the HARTH dataset. Most of the existing work identified the imbalanced nature of the class of the HARTH dataset as a challenging task but we have overcome that by scaling data and principal component analysis. Our findings indicate that additional work needs to be done to create the latest machine-learning techniques that would enable more accurate human activity recognition in free-living environments.

B. Recommendations for Future Works

Investigate new features that could record more data on human activity. More experiments are to be conducted by varying the window size and step size to generate features. More complex models using Ensemble methods can be experimented with. Optimizing models for real-time deployment in systems. Ethical consequences of deployment with regards to user privacy and consent. Incorporate User feedback to increase customization and flexibility throughout the model's training phase.

References

[1] Murman D. L. (2015). The Impact of Age on Cognition. Seminars in hearing, 36(3), 111–121. https://doi.org/10.1055/s-0035-1555115

[2] A. Logacjov, K. Bach, A. Kongsvold, H. B. Bårdstu, and P. J. Mork, "HARTH: A Human Activity Recognition Dataset for Machine Learning," Sensors, vol. 21, no. 23, Art. no. 23, Jan. 2021, doi: 10.3390/s21237853.

[3] K. Bach et al., "A Machine Learning Classifier for Detection of Physical Activity Types and Postures During Free-Living," Journal for the Measurement of Physical Behaviour, vol. 1, no. aop, pp. 1–8, Dec. 2021, doi: 10.1123/jmpb.2021-0015.

[4] Oyibo, K., Wang, K., & Morita, P. P. (2023). Using Smart Home Technologies to Promote Physical Activity Among the General and Aging Populations: Scoping Review. Journal of medical Internet research, 25, e41942. https://doi.org/10.2196/41942

[5] Human activity recognition in car workshop - thesai.org. (n.d.). https://thesai.org/Downloads/Volume13No4/Paper_95-Human_Activity_Recognition_in_Car_Workshop.pdf

[6] Snoun, A., Bouchrika, T. & Jemai, O. Deep-learning-based human activity recognition for Alzheimer's patients' daily life activities assistance. Neural Comput & Applic 35, 1777–1802 (2023). https://doi.org/10.1007/s00521-022-07883-1

[7] Ray, Abhisek, Maheshkumar H. Kolekar, R. Balasubramanian, and Adel Hafiane. "Transfer Learning Enhanced Vision-based Human Activity Recognition: A Decade-long Analysis." International Journal of Information Management Data Insights 3, no. 1 (2023): 100142. Accessed November 14, 2023. https://doi.org/10.1016/j.jjimei.2022.100142.

Appendix

A. CRediT statement

In figure 15, we show the CRediT statement (Contributor Roles Taxonomy).

Name	Role and Responsibilities
Anjali Himanshu Ojha	Data Collection and Preprocessing, Support Vector Machine (SVM) Model Development, Model Testing, Model Evaluation and Performance Metrics, Documentation of Development Process
Sowmya Manchikanti	Data Analysis and Visualization, Logistic Regression Model Development and Testing, Hyperparameter Tuning for Logistic Regression, Model Evaluation and Performance Metrics, Report Preparation
Akanksha Tyagi	Feature Engineering and Data Transformation, xGBoost Model Development and Testing, Ensemble Strategy Design and Testing, Model Evaluation and Performance Metrics, Preparing Interim Report of Findings
Shashank Reddy Kandimalla	Random Forest Model Development and Testing, Hyperparameter Tuning, Model Evaluation and Performance Metrics.
Sujata Mahajan	K-Nearest Neighbors (KNN) Model Development, Hyperparameter Tuning for KNN, Model Evaluation and Performance Metrics, Preparing Presentation

Fig. 15. CRediT statement

B. Rubric

In table VI, we describe how the rubric is met.

TABLE VI
RUBRIC

Criteria	How it is met
Code Walkthrough	Will be done during presentation.
Presentation Skills (Includes time management)	Will be done during presentation
Discussion / Q&A	Will be done during presentation
Demo	A python script is provided to run the demo. We'll run it during presentation.
Visualization Includes exploratory analysis (heat maps and other visuals)	We show the visualization of the dataset in 2D and 3D using PCA. We show the distribution of the labels for each class. We show the scree plot showing how much variance in the data is explained by the principle components. We show the heatmap of the confusion matrix to show how each model performs on each class.
Report Format, completeness, language, plagiarism, whether turnItIn could process it (no unnecessary screenshots), etc	A clean and concise report is provided.
Version Control	We used github. A screenshot of the commits is provided. We also provide the link to repo in the main report.
Relates to sustainability	Section VII describes how our project helps in sustainable development
Lessons learned	Section VIII describes the key learnings.
Prospects of winning competition / publication	We can explore the submission of our work on kaggle when there's a competition on this dataset. We believe our models will rank quite high.
Innovation	Section VI describes the innovations.
Evaluation of performance	Section V provides the evaluation of each model.
Teamwork	We worked as a team collaboratively to deliver the project.
Technical difficulty	Section III-D describes the technical difficulties in the project.
Practiced pair programming?	We used google collab to code together. The jupyter notebook on our github repository links the google collab notebook. Using pair programming enabled us to write reusable code for all the models
Practiced agile / scrum	We used trello https://trello.com/b/ferrusCz/msda245group6 We also provide screenshots.
Used Grammarly / other tools for language?	We used grammarly chrome extension. Screenshot is provided.
Slides	Slides have been submitted
Saving the model for quick demo	The models have been saved as pickle files. They are uploaded to github due to size constrains.
Used LaTeX	We used overleaf. The latex files are provided. We don't include screenshots due to size limits.
Used creative presentation techniques	Used prezi for presentation
Literature Survey	Section II provides literature survey.