http://communications.science.ankara.edu.tr

Commun.Fac.Sci.Univ.Ank.Ser. A2-A3
Volume 65, Number 2, Pages 166–178 (2023)
DOI:10.33769/aupse.1306885
ISSN 1303–6009 E-ISSN 2618–6462

COMMUNICATIONS SERIES A2-A3

Research Article; Received: May 30, 2023; Accepted: July 11, 2023

CLASSIFICATION OF HUMAN ACTIVITIES BY SMART DEVICE MEASUREMENTS

Muruvvet KALKAN[1] and Yilmaz AR[1]

[1]Department of Computer Engineering, Ankara University, Ankara, TÜRKİYE

Abstract. The prevalence of activity detectors in users' personal mobile devices has been incorporated into an increasing interest in research into physical function recognition (HAR - Human Activity Recognition). With this research interest, different enterprises developed HAR systems working with measurement devices and still work on this subject. Although many HAR systems have been developed, there are still concrete practical limits. This situation is improved with modern techniques such as machine learning. A properly trained machine learning model predicts human activity from measured data. The data was measured at certain time intervals by sensors on smartphones. These different machine learning architectures were trained on sensor data that detected human activities, and their accuracy was calculated. A HAR system that predicts human activity is constructed separately with five approaches. KNN, Random Forest, Decision Tree, MLP and Gaussian Naive Bayes algorithms were used, and KNN produced the most accurate results.

1. Introduction

Detecting human motion is applied on various fields for scientific or commercial gain. Especially today, widely used mobile health applications come with the feature of detecting human activities. Various hardware of smart devices allows the data to be gathered. Modern ready-to-use smart phones and watches contain a diverse set of embedded sensors. For example, accelerometer, gyroscope, compass, WiFi, NFC and GPS [1]. The proliferation of such sensor-rich mobile devices is already in our daily lives. This provides an opportunity to unobtrusively capture information from human behavior in real time. It also provides easier development and the rapid growth of public mobile sensing applications. Thus, new possibilities are available for new mobile sensing research.

Among the sensors found on mobile device platforms, the accelerometer is one of the oldest and the most common. The accelerometer has gained immense popularity

Keywords. Human activity recognition, HAR, machine learning, accelerometer, gyroscope.
kalkanm@ankara.edu.tr-Corresponding author; 0000-0001-8056-1905
ar@ankara.edu.tr; 0000-0003-2370-357X.

in HAR research as it allows the recognition of a wide variety of human activities while having a relatively small energy consumption. Accelerometer-based HAR is used in numerous fields, including smart homes, healthcare, daily activity tracking, fitness tracking, elderly fall detection, and transportation mode detection [1]. Other motion-related sensors, such as the compass and gyroscope, are becoming more common and are often used to assist and complement the accelerometer. Especially gyroscope is useful in that sense, since it measures the state of the device according to gravity: oblique, vertical, horizontal. Motion sensors can also be paired with other sensors, such as GPS, GSM, WiFi and barometer, especially to recognize tasks beyond basic HAR.

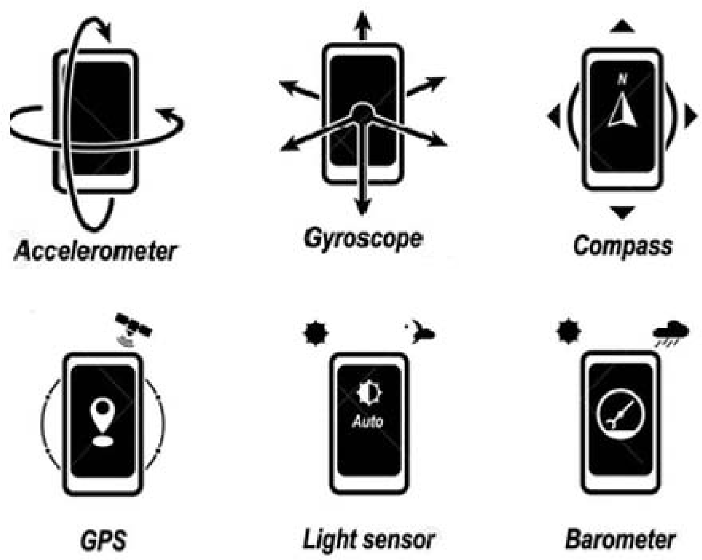

Figure 1. Sensors in smart mobile devices [2].

In this study, measurements are taken at small time intervals to make numerical calculations from the accelerometer and gyroscope. The body's acceleration/angular velocity is calculated from the mean, standard deviation of device acceleration/angular velocity by these measurements are also included in the original measurement data. Therefore, the success of ML classification algorithms performed on recognizing human activities with this data is tried in this work.

In the following sections certain points are made. In Problem Definition, the subject of the research is given. In Literature Review, the past scientific studies are examined on the subject of the domain and the proposed method. In Data set, input data with the size, type and the way it is produced is described in detail.

In Methodology, the proposed method with algorithms is given step by step and explained in detail. In Experimental Results, Discussion and Conclusion sections, the results of the experiment are displayed, compared and discussed, then all the work done is concluded.

2. Literature Review

Human activity recognition has always been an interested topic for all researchers whose area of expertise is feasible. Therefore, there are a lot of different methods to perform HAR depending, like video surveillance, sound tracking, image processing, motion detection with sensors etc. Depending on the input type, the methods narrow accordingly but overall they are various [3]. In this study, the method of motion detection with sensors is preferred, so sensor data is used as input and machine learning classification algorithms: KNN, Random Forest, Decision Tree, MLP, Naive Bayes are chosen for processing the data and performing HAR.

K-Nearest Neighbor (KNN) is one of the simplest algorithms belonging to the unsupervised class of machine learning algorithms. There are two important things to know about KNN. First, KNN is a non-parametric machine learning algorithm, excluding the number of nearest neighbors (K). This means that no assumptions are made about the data set when the model is used. Rather, the model is built entirely from the data provided. Second, KNN makes no generalizations between a training and test set, so all training data is also used when the model is asked to make predictions. KNN models calculate similarity using the distance between two points on a graph. The greater the distance between the points, the less similarity. There is more than one way to calculate the distance between points, but the most common distance measure is Euclidean distance. When applying this process, it assumes the similarity between the new case/data and existing cases and puts the new case in the category most similar to the existing categories. For this reason, KNN is known as a lazy machine learning algorithm. For this data set, the number of nearest neighbors is selected as five (K = 5) [4].

The Random Forest algorithm consists of the required number of decision tree algorithms that work in a collective just like the forests consisting of trees. Random Forest, which can be used for both regression and classification, belongs to the supervised category of ML algorithms. In this algorithm, which consists of a large number of decision trees, the decision trees are fused, and each decision tree is trained on a different observation sample, then their results are combined to produce an overall correct result [5].

The Decision Tree algorithm belongs to the supervised class of ML algorithms. Unlike other supervised ML algorithms, this algorithm can also be used in regression and classification problems. The purpose of using a Decision Tree is to create a training ML model that can be used to predict the class or exact value of the targeted variable by learning the simple decision rules generated from the training data. In Decision Trees, one starts at the root of the tree to predict the class of a target record. The values of the root's attribute are compared with the attribute values of the target record. On the basis of the comparison, the branch that meets that value is followed and the next node is passed [6].

Multi-Layer Perceptron (MLP), a supervised ML algorithm, has a multi-layered network structure used especially in classification problems. When making forward calculations, a net input value is found by calculating the inputs transferred to the system. The output of the current phase is calculated by passing the obtained input value through the activation function. The calculated output is transferred to the next layer. These processes continue from the input layer to the next layer, the middle layer, and from the middle layer to the output layer. Finally, the output values are created in the output layer. Thus, the first stage of learning is completed. An error value will occur if the output value received from the network is different from the expected output value. With backward calculation, the error value is distributed and updates are made in each iteration and it is expected to get closer to the expected result [7].

Naive Bayes is a supervised ML algorithm based on the theorem put forward by Thomas Bayes and can be used to classify data. It is a classification algorithm based on probability methods. Predicts which class the target record or target data point may belong to, using probability calculations. The target record/data point is assumed to belong to the class with the highest value from the calculated probabilities. The more data entered into the algorithm for training purposes, the higher the accuracy of predicting the result. Naive Bayes is the oldest known ML classification algorithm, which is primitive, but easy to implement [8].

There are many academic studies on recognizing human activities. Ganapati Bhat and his team collected their data by getting the measurements from special wearable IoT devices they designed rather than smartphones [9]. They then trained their models on this data with reinforcement learning. Allan Stisen and his research associates studied heterogeneity in the measurements of the devices by taking measurements from different HAR devices [1]. In an evaluation run by Jindong Wang and his team, experiments on HAR using deep learning were discussed, compared and interpreted [10]. Also, Yilmaz and his colleague did a study on HAR using deep learning with genetic algorithms and proposed a novel approach [11]. In two studies, special HAR systems were developed to assist and support elderly people [12, 13]. There is also a recent study that delves into HAR field with smart phones sensors as input [14]. Lastly, a specific HAR system designed for the processors of mobile devices is designed using SVM (support vector machine) [15].

3. Methodology

The data set of this experiment consists of sensor measurements made with smartphones and human activities that they describe through these measurements. The measurements are made with the accelerometer and gyroscope inside the smart devices. The data set consists of 10 thousand rows. There are 563 columns in total. The columns include:

- Acceleration from the triaxial accelerometer and estimated body acceleration.
- Angular velocity measured by the three-axis gyroscope.
- Variable values held in 561 columns, measured and calculated at specific time intervals (with frequencies).
- Subject: The code of the person doing the activity, namely which user it is.
- Activity: The activity that the user is doing.

The activity column is preferred to classify the data set. With the processed measurement data, it is determined which activity the movement is. For this purpose, the classification algorithms of machine learning (ML) were applied on the data set by accepting the Activity column as the class column.

In the data set, the Activity column can have one of the six classes in Table 1.

Table 1. The classes of the data sets.

Class	Activitiy
WALKING	Walking
WALKING_UPSTAIRS	Walking Upstairs
WALKING_DOWNSTAIRS	Walking Downstairs
SITTING	Sitting
STANDING	Standing
LAYING	Laying

The data set was created by a certain number of volunteers, between the ages of 18-49. Each of them carried out six different activities with a smart phone he wore around his waist. Experiments were video-recorded to tag actions later as they were performed. The data taken from the accelerometer and gyroscope were processed by passing through noise filters. All results were combined and the final version of the data set was divided into two as 70% training and 30% test data set.

In the experiment certain libraries of Python are preferred and all the calculations were done by them. They are Scikit-learn, NumPy, Pandas and Matplotlib. "Scikit-learn is one of the most popular ML libraries used by the data science community. Available in Python programming language, scikit-learn is very effective for supervised or unsupervised ML applications and data processing. However, Scikit-learn allows developers to use many ML algorithms. Scikit-learn is built on commonly known data processing Python libraries such as NumPy, Pandas and Matplotlib." [16].

All models were run sequentially with the algorithm steps given on the data set. These steps in the program is given in the following Algorithm 1.

Algorithm 1. *Building the experimental model, training and evaluation.*

procedure MODEL CONSTRUCTION

Fetch the data set

Split them as training, test and validation data sets

Approximately 70% training, 20% validation, 10% test

model ←*get the model (KNN, Decision Tree, Random Forest, MLP, Gaussian Naive Bayes)*

$model.trainable \leftarrow True$

$$metrics \leftarrow [accuracy, loss, precision, recall, f1Score, roc]$$

$model.compile(metrics)$

Evaluate model by the test metrics

end procedure

The Algorithm 1 can be explained with following steps.

(1) The data set is fetched.
(2) The data set is partitioned to train and test data sets (70% - 30%, respectively).
(3) The chosen ML model is loaded from Scikit-learn (One of KNN, Random Forest, Decision Tree, MLP or Naive Bayes)
(4) The model is trained with training data set
(5) Trained model produce predictions by the test data set and these predictions are saved.
(6) Trained model produce prediction probabilities by the test data set and these prediction probabilities are saved.
(7) Using produced predicted results, prediction probabilities and actual labels of the test data set, the following evaluation metrics are calculated: Accuracy, loss, precision, recall, F1-score and ROC AUC.
(8) All steps from 3 to 7 are repeated for each algorithm and all of their scores are saved.
(9) Saved scores are displayed in graphs with Matplotlib.

4. Experimental Results

Algorithms run on the data set with 10.300 tuples were analyzed according to some evaluation metric results. The metrics examined are accuracy, loss, precision, recall, f1-score and ROC AUC values. Except for the loss, all of them are calculated from the values of prediction systems by 2:

- **TP:** Number of True Positives
- **TN:** Number of True Negatives
- **FP:** False Positives
- **FN:** False Negatives

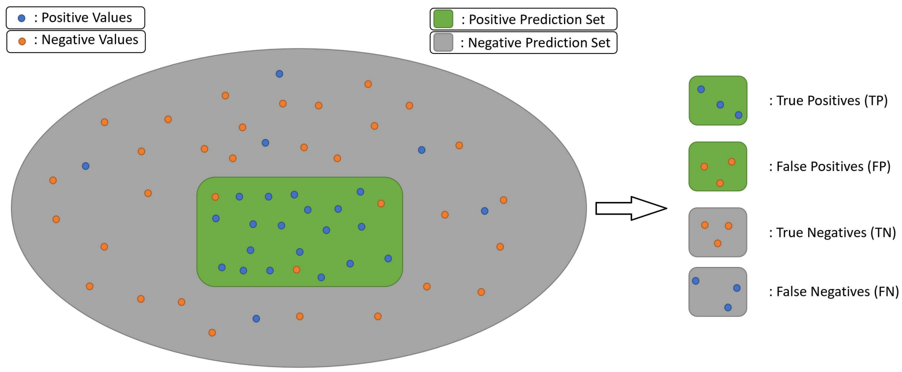

FIGURE 2. Prediction value definitions.

Accuracy is defined as the ratio of correctly detected data in the model to all data. It is not sufficient on its own to measure the relevance of the study, but it is the most important value, Equation 1.

Loss measures the difference between the true value of the sample and the predicted value in the model. The greater the difference, the greater the loss. It indicates how far are the predictions from real values.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{1}$$

Precision is the ratio of actual positives to all predicted positives. It measures the precision of positively predicted values, Equation 2.

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

Recall shows how well the system is at not missing actual positives. The higher it is, the more true positives are accurately predicted, Equation 3.

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

The F1-score value is the harmonic mean of the precision and recall values. Harmonic averaging is used to measure the balance between often contradicting precision and recall. F1-Score formula is given in Equations 4 and 5.

$$F_1 = 2 \bullet \frac{precision \bullet recall}{precision + recall} \tag{4}$$

$$F_1 = \frac{TP}{TP + \frac{1}{2}(FP + FN)} \tag{5}$$

Area Under the Curve of Receiver Operating Characteristic (ROC AUC or AUC of ROC), the area under the ROC curve is expressed as AUC, see Figure 3. The larger this value in a model, the better the machine learning model is at classification. ROC curve is expressed on the 2D space with x-axis as the False Positive Rate (FPR) and y-axis as the True Positive Rate (TPR) of the model, see Equations 6 and 7.

$$TPR = \frac{TP}{TP + FN} \tag{6}$$

$$FPR = \frac{FP}{FP + FN} \tag{7}$$

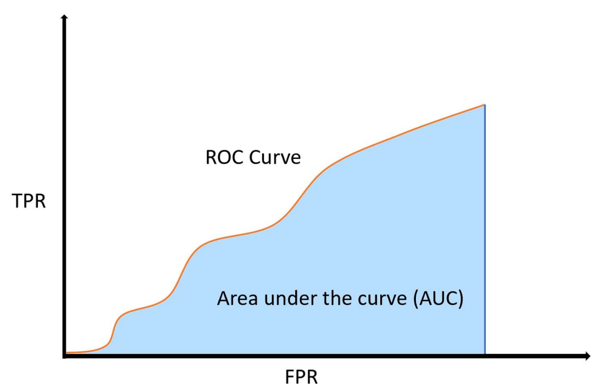

Figure 3. Area under the curve of receiver operating characteristic (ROC AUC).

All the metrics mentioned on the given models have been calculated separately. Analyzes are given according to the calculated values. Considering the accuracy values in the experiment, the KNN algorithm got the highest accuracy rate with 96.6%. MLP produced 94.56%, Decision Tree 89.09%, Random Forest 78.03% and Gaussian Naive Bayes algorithm 76.18% accuracies. The accuracy distribution of all algorithms is approximately between 76% and 96%. ROC AUC values range from 99% to 96%, and all models are close to each other, Figure 4.

When the loss data is examined, the MLP model got the least loss with a value of 0.15. Other models are followed by KNN with 0.16, Decision Tree with 0.31, and

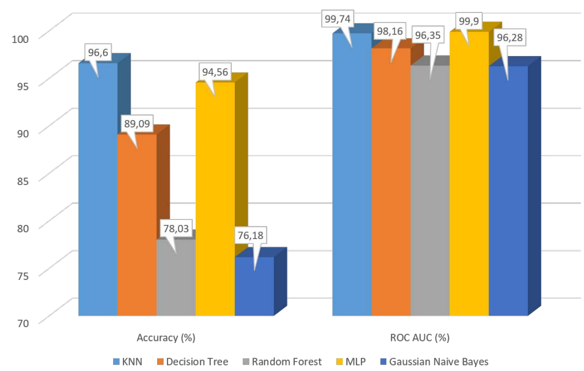

FIGURE 4. Graph of accuracy and AUC of ROC.

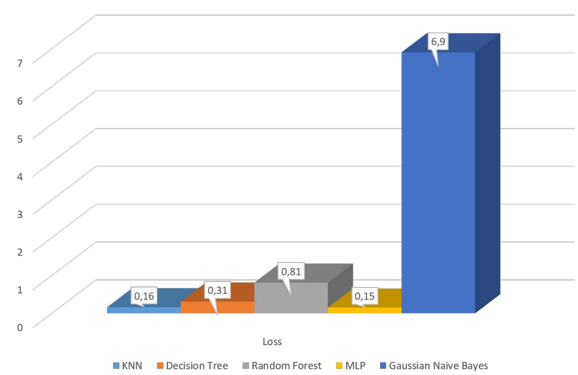

FIGURE 5. Graph of loss.

Random Forest with 0.81. With a value of 6.9, the Gaussian Naive Bayes algorithm has the highest loss, Figure 5.

Precision metric produced 96.82%, MLP 95.83%, Decision Tree 89.52%, Gaussian Naive Bayes 79.62%, Random Forest 78.07% on KNN. Recall values are 96.71% in the KNN model, 94.72% in MLP, 88.51% in Decision Tree, 77.59% in Random Forest, and 76.74% in Gaussian Naive Bayes. Finally, the F1-Score metric is produced as 96.75% by the KNN algorithm, 94.72% by MLP, 88.67% by Decision Tree, 77.4% by Random Forest, 75.42% by Gaussian Naive Bayes given in Table 2 and Figure 6.

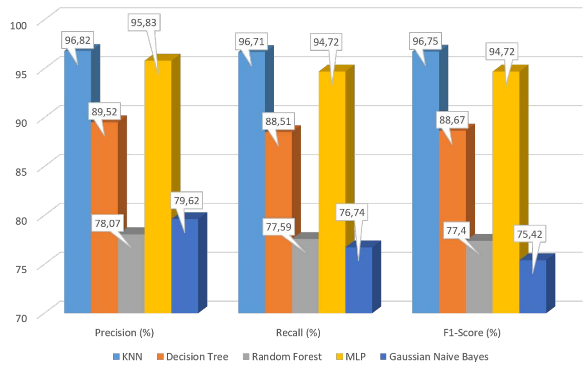

FIGURE 6. Graph of precision, recall and F1-score.

TABLE 2. The metrics results of the algorithms.

Algorithm	**Accuracy%**	**Loss**	**Precision%**	**Recall%**	**F1-Score%**	**AUC of ROC%**
KNN	96.60	0.16	96.82	96.71	96.75	99.74
Decision Tree	89.09	0.31	89.52	88.51	88.67	98.16
Random Forest	78.03	0.81	78.07	77.59	77.40	96.35
MLP	94.56	0.15	95.83	94.72	94.72	99.90
Gaussian Naive Bayes	76.18	6.90	79.62	76.74	75.42	96.28

5. DISCUSSION

When the study is evaluated in general, the model that produces the best result in almost all metric values is the KNN algorithm. While KNN is performing the classification operations, a distance is measured for each data to the neighbors of it, then a result is estimated by these distances. The label of the nearest neighbor is considered the class of the test data. Since the data set is a quite dense matrix, KNN with its own approach gave the best estimation values with a result of 96.6

The MLP classification algorithm iterates a lot of times to make itself to predict with the least error possible. Initially randomly assigned error weights are updated at each iteration and the network is optimized. In this way, with an accuracy rate of 94.56%, MLP can be taken as the second highest classification algorithm in the study. Certainly, it points that the data set of the experiment is a good fit for MLP's way of classification.

The Gaussian Naive Bayes algorithm calculates the proximity to other data for the data to be classified and assigns it to the class with the highest probability. While it can give very good results with a small data set, the classification algorithm that can produce low values for large-volume data sets is 76% in the experiment. It has the lowest accuracy rate of 18. Considering it is a primitive algorithm compared to its peers, such a result is to be expected.

The Decision Tree model, which is easy to understand and interpret, followed with a rate of 89.09%. The model used is average for the experiment. Although not as strong as KNN and MLP, it produced satisfactory values.

Random Forest yielded disappointing results with 78.03%. Although it allows to create multiple decision trees on the data set and train each one separately, Random Forest could not provide satisfactory results for this experiment. Although it would be expected of Random Forest to produce far superior results to Gaussian Naive Bayes algorithm, their results are close, so Random Forest can be considered as a poor choice for this task.

When all algorithms are examined, the most powerful algorithm for the considered data set is the KNN model. It provided the best values in almost all metrics. However, the Random Forest algorithm could not meet the desired results and Gaussian Naive Bayes got the worst scores, especially the highest loss rate with a 6.9. The models that make the best classification in the study can be taken as KNN and MLP.

6. Consclusion

In this study, Classification of Smart Mobile Devices in the Recognition of Human Activities is provided by KNN, MLP, Random Forest, Decision Tree, and Gaussian Naive Bayes classification algorithms using the Scikit-learn library. In the application, the movement performed with various parameters is classified into six activities in total. Classification algorithms are evaluated by accuracy, loss, precision, recall, F1-score, and AUC-ROC curve. The KNN algorithm gave the best accuracy for the data set used, and the Gaussian Naive Bayes algorithm showed the lowest rate. Random Forest performed worst considering expectations. For the Loss metric, MLP produced the most accurate result, and the Gaussian Naive Bayes model produced the worst result. KNN algorithm produced the best results in all metrics.

Author Contribution Statements Conceptualization, M.K.; methodology, M.K.; software, M.K.; validation, M.K. and Y.A.; writing—original draft preparation, M.K. and Y.A.; writing—review and editing, M.K. and Y.A.; supervision, Y.A. All authors have read and agreed to the published version of the manuscript.

Declaration of Competing Interests The authors declare no conflict of interest.

References

[1] Stisen, A., et al., Smart devices are different: assessing and mitigating mobile sensing heterogeneities for activity recognition, *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, (2015), 127-140, https://doi.org/10.1145/2809695.2809718.

[2] Accelerometer, gyroscope, compass, GPS, light sensor, barometer. Important phone functions. Black icon, (2023). Available: https://stock.adobe.com/images/accelerometer-gyroscope-compass-gps-light-sensor-barometer-important-phone-functions-black-icon/170752038. [Accessed: May 2023].

[3] Vrigkas, M., Nikou, C. and Kakadiaris, I. A., A review of human activity recognition methods, *Front. Robot. Al*, 2 (2015), 28, https://doi.org/10.3389/frobt.2015.00028.

[4] Cover, T. and Hart, P., Nearest neighbor pattern classification, *IEEE Trans. Inf. Theory*, 13 (1) (1967), 21-27, https://doi.org/10.1109/TIT.1967.1053964.

[5] Breiman, L., Random forests, *Mach. Learn.*, 45 (1) (2001), 5-32, https://doi.org/10.1023/A:1010933404324.

[6] Quinlan, J. R., Induction of decision trees, *Mach. Learn.*, 1 (1) (1986), 81-106, https://doi.org/10.1007/BF00116251.

[7] Rosenblatt, F., The perceptron: a probabilistic model for information storage and organization in the brain, *Psychol. Rev.*, 65 (6) (1958), 386-408, https://doi.org/10.1037/h0042519.

[8] Vikramkumar, B. and Vijaykumar, T., Bayes and naive bayes classifier, arXiv.1404.0933, (2014), https://doi.org/10.48550/arXiv.1404.0933.

[9] Bhat, G., Deb, R., Chaurasia, V. V., Shill, H. and Ogras, U. Y., Online human activity recognition using low-power wearable devices, *2018 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, (2018), 1-8, https://doi.org/10.1145/3240765.3240833.

[10] Wang, J., Chen, Y., Hao, S., Peng, X. and Hu, L., Deep Learning for sensor-based activity recognition: A survey, *Pattern Recognit. Lett.*, 119 (2017), 3-11, https://doi.org/10.1016/j.patrec.2018.02.010.

[11] Yilmaz, A. A., Guzel, M. S., Bostanci, E. and Askerzade, I., A novel action recognition framework based on deep-learning and genetic algorithms, *IEEE Access*, 8 (2020), 100631-100644, https://doi.org/10.1109/ACCESS.2020.2997962.8.

[12] Chernbumroong, S., Cang, S., Atkins, A. and Yu, H., Elderly activities recognition and classification for applications in assisted living, *Expert Syst. Appl.*, 40 (5) (2013), 1662-1674, https://doi.org/10.1016/j.eswa.2012.09.004.

[13] Janaki, M., Geethalakshmi, Dr. S. N., An efficient system for human activity recognition and monitoring for elderly people using machine learning, *SJIS*, 35 (1) (2023), 1194-1206.

[14] Garcia-Gonzalez, D., Rivero, D., Fernandez-Blanco, E. and Luaces, M. R., New machine learning approaches for real-life human activity recognition using smartphone sensor-based data, *Knowl.-Based Syst.*, 262 (2023), 110260, https://doi.org/10.1016/j.knosys.2023.110260.

[15] Anguita, D., Ghio, A., Oneto, L., Parra, X. and Reyes-Ortiz, J. L., Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine, *Proceedings of the 4th International Conference on Ambient Assisted Living and Home Care*, (2012), 216-223, https://link.springer.com/chapter/10.1007/978-3-642-35395-6-30.

[16] SciKit-Learn, (2023). Available:https://scikit-learn.org/stable/about.html. [Accessed: May 2023].

Appendix

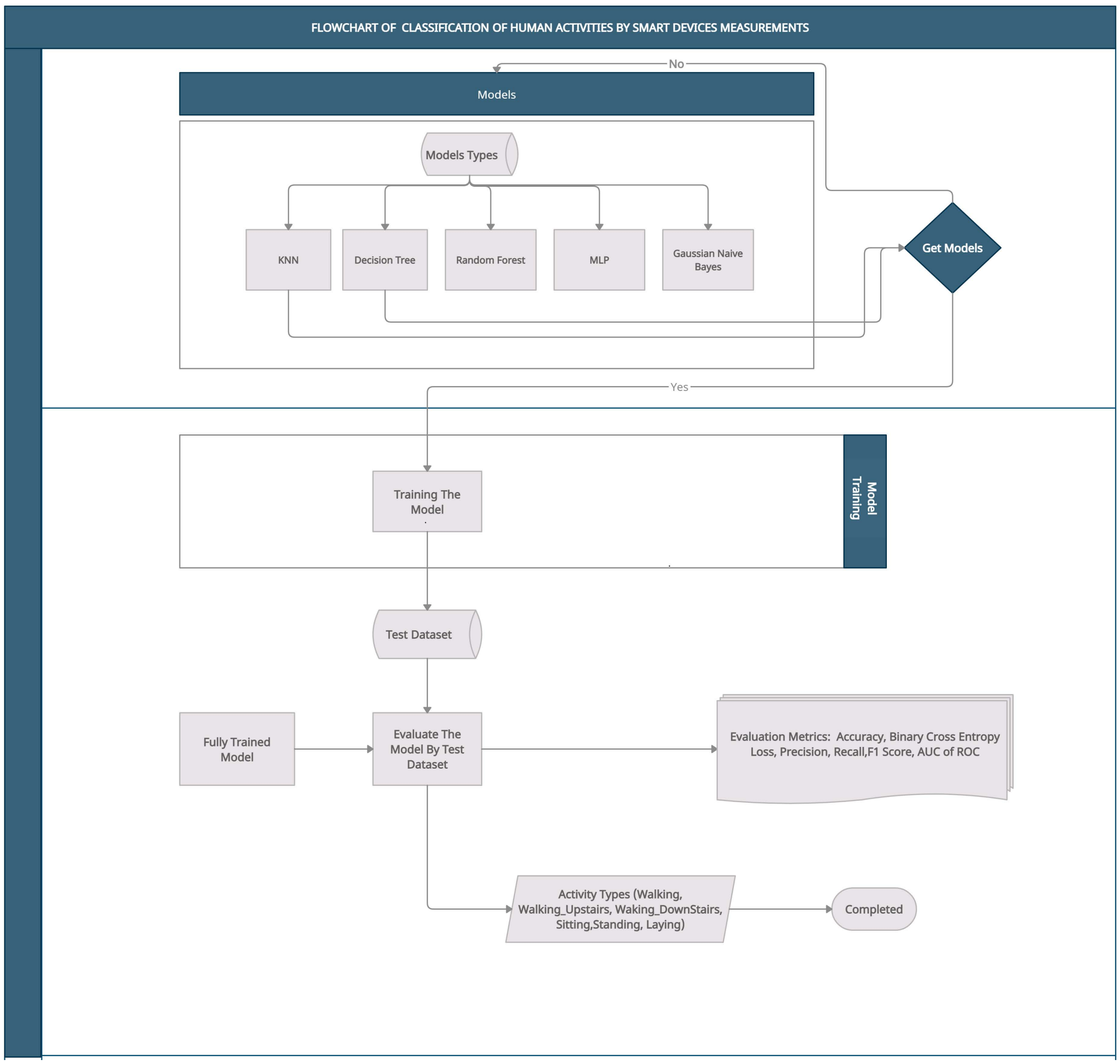

Figure 7. Flowchart representation of the experiment.