Int. J. Nonlinear Anal. Appl. 13 (2022) 2, 1183–1200
ISSN: 2008-6822 (electronic)
http://dx.doi.org/10.22075/ijnaa.2022.6356

Evaluation of machine learning approaches for sensor-based human activity recognition

Hala Muhanad Yousif*, Dhahir Abdulhade Abdulah

Department of Computer Science, College of Science, University of Diyala, Baqubah, Iraq

(Communicated by Madjid Eshaghi Gordji)

Abstract

Human Activity Recognition (HAR) systems used in healthcare have attracted much attention in recent years. A HAR system consists of a wearable device with sensors. HAR has been used to suggest several machine learning (ML) algorithms. However, only a few research have looked at how to evaluate HAR to identify physical activities. Nevertheless, obtaining an explanation for their performances is complicated by two factors: the lack of implementation specifics and the lack of a baseline evaluation setup that makes comparisons unfair. For establishing effective and efficient ML–HAR of computers and networks, this study uses ten common unsupervised and supervised ML algorithms. The decision tree (DT), artificial neural network (ANN), naive Bayes (NB), k-nearest neighbor (k-NN), support vector machine (SVM), random forest (RF), and XGBoost (XGB) algorithms are among the supervised ML algorithms, while the k-means, expectation-maximization (EM), and self-organizing maps (SOM) algorithms are among the unsupervised ML algorithms. Multiple algorithms models are presented, and the turning and training parameters in ML (DT, ANN, NB, KNN, SVM, RF, XGB) of each method are investigated in order to obtain the best classifier assessment. Differ from earlier research, this research measures the true negative and positive rates, precision, accuracy, F-Score as well as recall of 81 ML-HAR models to assess their performance. Because time complexity is a significant element in HAR, the ML-HAR models training and testing time are also taken into account when evaluating their performance efficiency. The mobile health care (M_HEALTH CARE) dataset, which includes real-world network activity, is used to test the ML-HAR models. In general, the XGB outperforms the DT-HAR, k-NN-HAR, and NB-HAR models in recognizing human activities, with recall, precision, and f-scores of 0.99, 0.99, and 0.99 for each, respectively, for health care mobile recognition.

Keywords: Machine Learning, Artificial Neural Network, Benchmarking, Supervised Learning Algorithms, k-means
2020 MSC: 62M45, 68T07, 68Q04

1 Introduction

IoT devices are tangible things that interact in a certain manner with the actual world. It could be a wireless gadget or a sensor on an assembly line. In any situation, the device is sensing what is going on in the real world. IoT devices vary in terms of functionality and different smart architectures such as buildings, healthcare, environment, smart city, efficient energy, mobility, manufacturing and smart agriculture [29]. As a result, IoT technology is evolving in the healthcare monitoring system in order to provide patients with appropriate emergency services [24]. Also,

*Corresponding author
Email addresses: scicompms21@uodiyala.edu.iq (Hala Muhanad Yousif), dhahair@uodiyala.edu.iq (Dhahir Abdulhade Abdulah)

Received: January 2022 *Accepted:* March 2022

it is being employed as an E-health application for various purposes. This includes early identification of medical problems, emergency notification, and computer-assisted rehabilitation. Sensors are included in the medical gadgets to keep track of the subject's health [33]. This sensor-based surveillance system collects a variety of signals or data from diagnostics and wards equipment and mines it for efficient and automatic healthcare control [4]. The IoT healthcare system allows for effective monitoring and tracking, which aids in improving people's health [27]. Furthermore, cloud computing is employed to manage healthcare data and cater to resource-sharing benefits such as flexibility and remote access to monitor patient data [31]. Currently, one of the most significant and vital studies has been to monitor human activity such as lying down, sitting, and walking. This is accomplished by using a sensor worn to monitor the patients, with the data being sent to professionals for analysis at the information center [5]. Because acceleration and angular velocity change with human movements, they can be used to extract human activities. Mobile sensors, in contrast to fixed sensors, are flexible and tiny, allowing them to be integrated into body gear or mobile devices. Mobile sensors are also advantageous since they are less expensive, require less energy, possess greater capabilities, and are less affected by the environment. As a result of the widespread use of mobile sensors in everyday life, there has been a surge in interest in mobile sensor-based activity detection, having a number of studies devoted to determining the suitability of mobile sensors for identifying human activities approach [34]. Human activities detection utilising mobile device sensors has traditionally been viewed as a multivariate time series classification problem. A significant methodology has been investigated on lightweight network design based on Machine learning to perform recognition tasks. The main measurements of recognition are accuracy and classification report including precession, recall, f-score—moreover, training and testing time as an important factor to enhance the lifetime of battery for IoT sensor. Although several ML monitor approaches exist, their accuracy remains a concern; accuracy is dependent on false and true positives. To lessen false positives and enhance proper categorization class, the accuracy issue must be addressed. This idea was the impetus for this investigation. In this paper, Decision Trees (DT), Artificial Neural Networks (ANN), Naive Bayes (NB), k-Nearest-Neighbors (k-NN), Support Vector Machine (SVM), Random Forests (RF), Expectation-Maximization (EM) clustering, XGBoost (XGB), Self-Organizing Maps (SOM) and K-means clustering, are used; these approaches have been shown to be effective in addressing the classification problem. The following is how this article is structured: Section 2 examines ML algorithms from a theoretical standpoint and briefly highlights important current studies in ML-HAR recognition. Section 3 examines the benchmarking approach for evaluating ML-HAR performance as well as the evaluation criteria used to assess classifier efficacy. Section 4 evaluates and tests evaluates the chosen ML algorithms to implement ML-JAR, as well as the data activities that are described and recommended for training and testing ML-HAR evaluation. Section 5 summarises the findings and makes recommendations for future research.

2 Background and Related Work

In this section, overview sensor types based on the functionality of the sensor, besides focusing on recent works, are related to machine learning for human activities recognition models. The performance of recognition models is measured by accuracy, precision and recall as a statistical test.

2.1 Sensor Types

2.1.1 Ambient sensor-based HAR (ASHAR)

Wearable sensor-based systems have achieved wide applications in HAR due to the ease of deployment and use of low-cost and satisfying performance. However, WSHAR can only provide the recognition of specific activities without giving the ambient context. Typical ambient sensors can instead provide rich contextual information relating to human daily activities, and ambient sensor-based HAR (ASHAR) systems have also been widely used in HAR.

A wide range of ambient sensors are available and are explored for HAR, including cameras, light sensors, reed switch sensors, RFID, PIR, temperature, flow sensor, pressure sensors, etc. Body-Worn Sensors.

A common HAR modality is body-worn sensors (for example, magnetometers, gyroscopes, and accelerometers). Given the variations in angular velocity and acceleration, these sensors can capture data on human actions. Several studies have used body-worn sensors in deep learning for HAR, with the majority focusing on accelerometer data. In addition, magnetometers and gyroscopes are frequently used in conjunction with accelerometers to detect activities of daily living (ADL) as well as specific sports activities.

2.1.2 Hybrid sensory-based HAR (ASHAR)

A single sensor modality, such as wearable or ambient alone, is often used in a HAR system. Each sensor modality has its own set of strengths and limitations, and in fact, single sensor modalities are sometimes unable to cope with

complicated scenarios. This establishes the groundwork for further research towards hybrid sensory HAR systems. Different sensor modalities offer diverse information and varied performances for specific tasks. For example, cameras deliver precise and direct information while coupled with privacy issues or working in a constrained space defined by the camera position and settings; ambient sensors (such as the temperature or light sensor) can provide important contextual information, whilst this can only give limited information for activity detection; door switches and other binary sensors are inexpensive and easy to install, but the captured ambient information is simple and limited to detect high-level activities; the accelerometer, the gyroscope and other wearable sensors are miniature-sized and can be flexibly worn on the body to capture sufficient motion-related information. However, they are unable to supply contextual information and face the issue of arbitrary data as a result of actions [32].

2.2 Human Activity Recognition ML Methods

There are unsupervised and supervised learning algorithms for training a machine learning algorithm. Supervised learning is dependent on data examples being classified in the training phase. Decision Trees (DT), Artificial Neural Networks (ANN), Naive Bayes (NB), k-Nearest-Neighbors (k-NN), Support Vector Machine (SVM), Random Forests (RF), as well as XGBoost (XGB) are examples of supervised learning techniques. In unsupervised learning, where clustering dominates the learning approach, data instances that are not labelled can be found. As illustrated in Figure 1, the unsupervised learning methods are K-means clustering, Expectation-Maximization (EM) clustering, and Self-Organizing Maps (SOM).

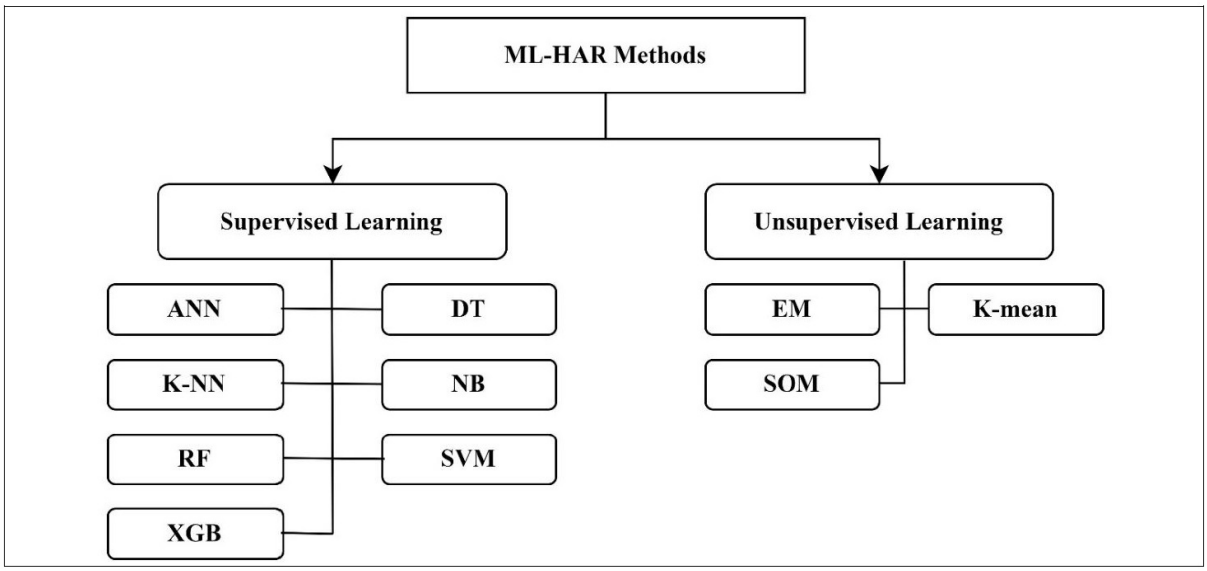

Figure 1: The ML-HAR models

2.2.1 Supervised Learning

In ML-HAR, eight different supervised machine learning algorithms will be assessed. The following diagrams show the essential notions of these algorithms.

Artificial Neural Networks (ANN):

In terms of visual representation, it is a weighted directed network with nodes and edges [1]. Linkages between artificial neurons are represented by artificial neurons and directed edges having weights (the strongest among neurons). Note that the output of a neuron is used as input by other neurons. They welcome input from the outside world in the form of a vector, which is akin to an image or pattern. Throughout the ANN's training, the weights are modified, which assists in the resolution of categorization issues. The ANN architecture is made up of three layers: input, output, and hidden. Each layer contains neurons. The input layer receives input from the outside world, meanwhile the output layer responds to the input layer's input based on its learning capacity. Moreover, the hidden layer acts as a link between the input and output layers, altering the input in order for the output layer to be used. Partially or entirely coupled layers are possible. The authors used a multilayer perceptron approach with backpropagation learning

in this study. Figure 2 depicts a general ANN design (I-H-O) for the c class, with I depicting the input nodes number, H representing the number of hidden layer nodes, and O representing the output nodes number.

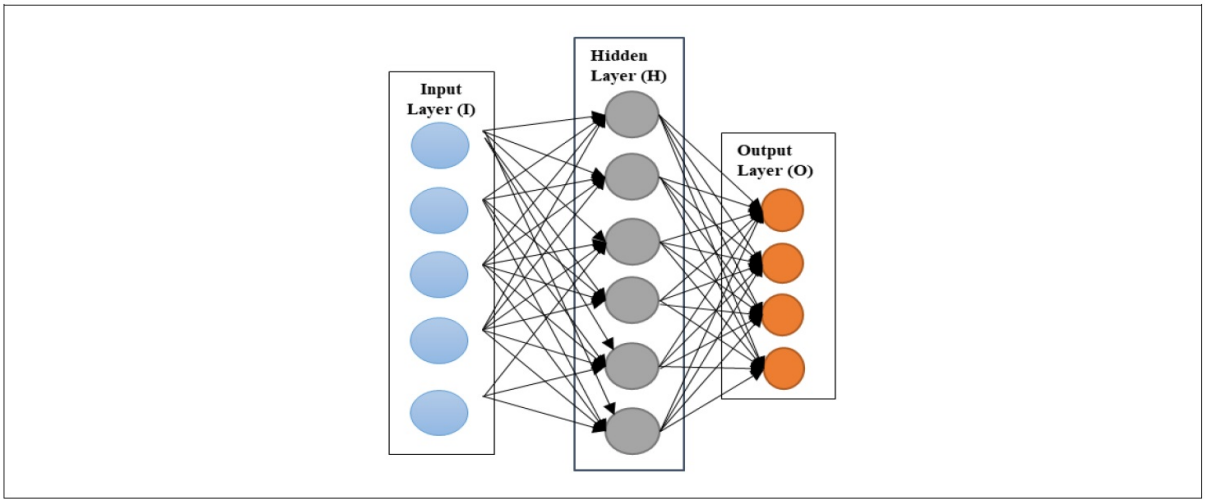

Figure 2: The architecture of the ANN

Decision Trees (DT):

A decision tree is one of the important algorithms in data science. There are many desirable decision techniques like CART, C5, and Quinlan's ID3. A decision tree describes the process formation and flow, in which every essential node indicates a test on an element, every transition represents a result of the test, and every leaf is associated with a class. Observations are split into parts to establish trees continuously.

They entail supervised learning algorithms commonly utilised to resolve machine learning classification issues. Tree models, often known as classification trees, are utilised when the target variable may accept discrete values as input. Branches, leaves, and nodes are examples of DT components. Branches indicate the set of attributes that appear in the class labels, whereas leaves represent the class labels. They are capable of working with both discrete and continuous data. The DT algorithm splits the samples into two or more homogeneous sets. Overfitting is a problem that DT has, which is addressed by Bagging and Boosting [23]. Over discrete data, the DT function performs well. A common structure example of a DT is indicated in Figure 3 [20].

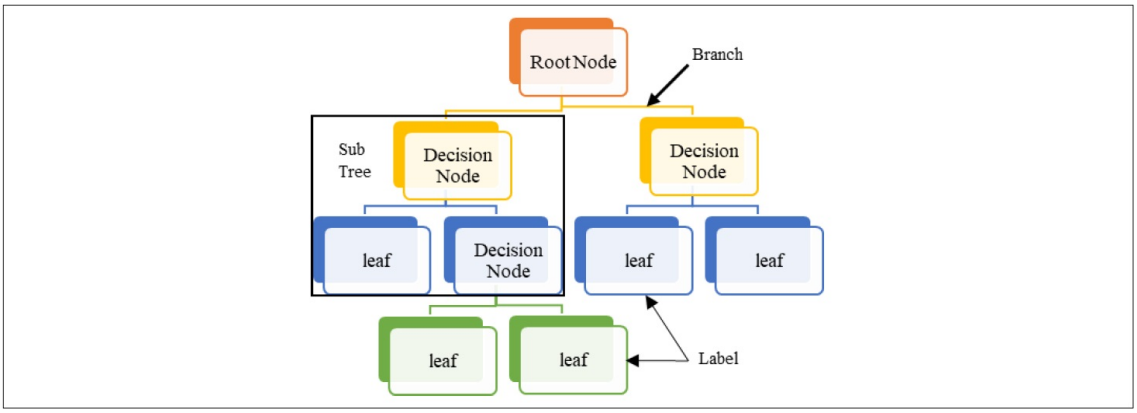

Figure 3: The architecture of the DT

K-nearest-Neighbors (k-NN):

It is a categorization system and case-based learning, according to [6]. The distance function of the k-NN method computes the correlations or differences between two instances or points. k-NN employs a variety of different distance

measures. Euclidean distance is a well-known and widely used distance measure. $D(a, b)$ is how Equation (2.1) represents it [16]:

$$D(a,b) = \sqrt{\sum_{i=1}^{r}(a_i - b_i)^2} \tag{2.1}$$

where a_i is the i^{th}-featured element of the instance a, b_i is the i^{th}-featured element of the instances b and r is the entire dataset features quantity. It is a non-parametric approach that does not make assumptions about how fundamental data is disseminated. The dataset is essentially used to determine the model's build. In practice, this is advantageous because the vast majority of the data is drawn from real-world datasets rather than mathematical hypotheses. A lazy algorithm, on the other hand, involves the creation of a model without the need for training data points. During the testing stage, all of the training data is utilized. It speeds up the training process, but it slows down and costs more during the testing process. Time and consumption will be affected by a costly testing stage. Hence, in the worst-case situation, the k-NN will need more time and memory to store training data and test all data.

Naive Bayes (NB):

The Bayes theorem is used to develop a set of probabilistic classifiers known as NB methods. It considers naïve independence hypotheses for each pair of attributes or features [18]. The NB can compete with the latest complex algorithms within its domain, such as the SVM and ANN, using an application before processing training data. A supervised learning structure makes it simple to train. The parametric computation for the NB models uses the technique of maximal probability in a number of real-world implementations that have been identified. In a nutshell, the NB model may work with or without Bayesian probability. The Bayes theorem is encapsulated in the following equation (2.2):

$$p(A|B) = \frac{p(A|B)P(A)}{p(B)} \tag{2.2}$$

in which B represents the active of the predictor attribute or antecedent event, while A represents the active of the target attribute or dependent event. Note that $P(A)$ denotes the prior probability of A, $P(A|B)$ denotes the posterior probability of B. Meanwhile, $P(B|A)$ is the likelihood of B if hypothesis A is true.

Random Forests (RF):

There are several supervised classification algorithms, and combining them may improve performance. The Random forest algorithm uses this understanding to generate an ensemble of many decision tree classifiers known as the for-est of the decision 165 of trees. Dr. Leo Breiman proposed the Random Forest method [31]. All the decision trees in the forest participate and the final results are drowned by the majority vote. Therefore, a higher number of trees in the forest give high accuracy results [2].

DT has difficulty with overfitting. An RF, on the other hand, effectively resolves the problem by allowing the average of many deep decision trees [8]. The RF algorithm is an ensemble-learning technique utilized to handle classification and regression problems. The development of several DT within the training timeframe is one of its responsibilities. During the execution of a classification function, the output includes the classes' mode of a specific DT. As a result, the RF achieves better results than the DT. Figure 4 depicts an RF architectural instance.

Support Vector Machine (SVM):

Its goal is to find a hyper-plane that divides all training cases into different groups (multi-class classification or binary classification). The SVM method, according to [15], accepts the stated instances and corresponding outputs, which are binary or N-ary. The model is further built to allow fresh instances to be classified into different groups. The training instance input sets are linearly separated by mapping the training instances into points in coordinate space. There are a variety of hyper-planes to choose from in order to separate the training instance sets. The longest distance from the most proximal occurrence of any class, on the other hand, is an ideal pick. P categorizes the instances correctly, although it has a smaller range away from the most proximal instance than the other two hyper-planes. Q, on the other hand, has a maximum range away, but it has a minor classification error, in which the hyper-plane P is selected instead. In addition, SVM works well in high-dimensional environments.

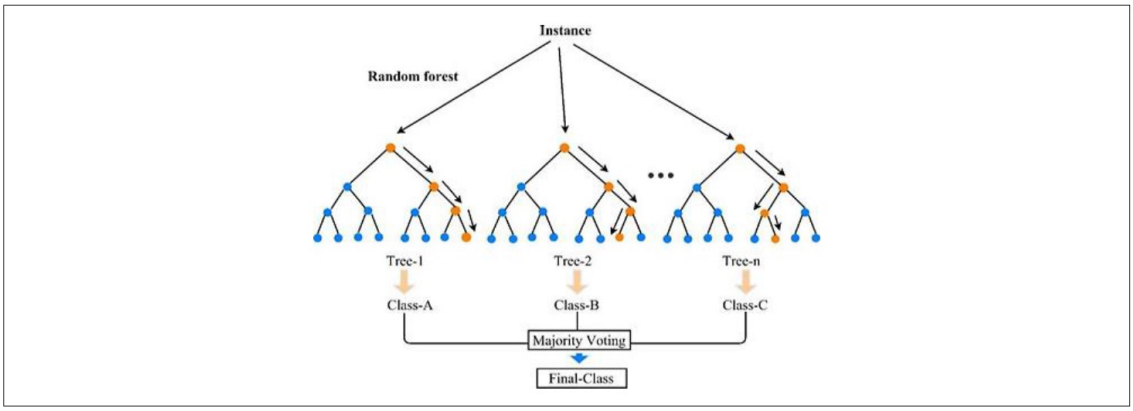

Figure 4: The architecture of the RF

XGB

Gradient Boosting Machine, which integrates gradient descent with boosting, is the starting point for Extreme Gradient Boosting (GBM). Boosting is an ensemble learning algorithm that assigns different weights to each iteration's training data distribution. Note that each boosting iteration modifies the training data distribution by adding weight to miss-classified error samples and subtracting weight from correct-classified error samples [11].

2.2.2 Unsupervised Learning Algorithms

There are three selected unsupervised machine learning algorithms to be assessed in ML-HAR. The following diagram illustrates the core notions of these algorithms.

Expectation-Maximization (EM):

It is extremely similar to k-means [21]. In two ways, Expectation-Maximization improves on the basic k-means clustering algorithm. First, according to one or more probability distributions, the EM algorithm computes the cluster membership probabilities. Based on the final clusters, it aims to maximize the data's total probability.

K-means:

From a distance-based perspective, it is one of the most basic unsupervised learning strategies. The n instances are divided into k clusters, with each instance operating as a collective inside the cluster with the closest mean. The biggest disadvantage of this strategy is: it demands the number of clusters k be pre-specified. k-signifies clustering seeks to partition p instances into $(k \leq p)$ sets$Z = \{Z_1, Z_2, ..., Z_k\}$ to minimize variance given a set of examples $(p_1, p_2, ..., p_n)$, in which each instance is a d-dimensional real vector. Then, k-means is determined as in the equation below (2.3) [13, 22].

$$a_z \min \sum_{i=1}^{k} \sum_{p \in Z_i}^{\max} \|p - m_i\| a_z \min \sum_{i=0}^{k} |Z_i| Var Z_i \tag{2.3}$$

provided a is an argument, μ_i denotes the mean of points in set Z_i.

Self-Organizing Maps (SOM):

It is predicated on the neural network models' unsupervised learning class. SOM can cluster data without knowing the input data class categories [17]. It provides a topology preserving mapping for mapping neurons that is derived from a high-dimensional data space (units). The distance between places is kept in mind during the mapping. In the SOM, neighboring maps units are mapped to mutually proximal places. The SOM network can recognize previous inputs. SOM is depicted in Figure 6.

$$y = \sigma(w^T + b) = \sigma\left(\sum_{k=1}^{n} w_k x_k + b\right) \tag{2.4}$$

2.3 Related Work

Even though certain HAR techniques can be generalized to any sensor modality, most are specialized and restricted. Ambient sensors, body-worn sensors, and object sensors are the three categories of modalities [25]. Several types of research were proposed the machine learning approach as significate recognition for human activities such as SVM, RF, and KNN [35]. The Adaboost ensemble classifiers have achieved performance for automated human activity recognition utilizing human body sensors, according to the experimental data. The precision of motion of 7-activities of the users' bodies was around 99%. This work has restrictions in terms of training and testing methods; nevertheless, they failed to include the dataset statistics, the quantity of training and testing samples, and the realistic qualities were removed by ignoring part of the classes [26]. The deep learning architecture is proposed with deep convolutional neural networks to conduct HAR utilising smartphone sensors via extracting characteristics of activities and signal dimension of time-series signals. This also offers a way to automatically and data-adaptively retrieve robust features from raw data having high accuracy [25]. The suggested RNN can combine the positive time direction (forward state) and the negative time direction (back state). Second, residual connections between stacked cells serve as gradient shortcuts, preventing gradient vanishing. The model was evaluated using the opportunity dataset and the public domain UCI dataset, and it scored 93.5 for accuracy and recall rate [35]. Employing wearable body sensor data to solve the problem of human activity detection as a classification problem. The researchers suggested that for good human activity recognition, they use a Deep Belief Network (DBN) classifier. They extract the crucial initial features from the raw body sensor data, then do a Kernel Principal Component Analysis (KPCA) and Linear Discriminant Analysis (LDA) to better handle the features and make them more robust so that they can be used in recognition training. To test the deep learning algorithm's performance, researchers used a real-world wearable sensor dataset. The results demonstrate that for 11500 samples of 12 activities or labels, the recommended adequate activity recognition performance is roughly 97% accurate [12]. Researchers proposed a hybrid deep learning model given the LSTM recurrent units and an ELM classifier. It was more appropriate to categorize the extracted features as well as shorten the runtime. Then, the measurement was time for train and test with low accuracy for nine activities based on the OPPORTUNITY dataset [28]. Their proposal a deep-learning model based on a Convolutional Long Short-Term Memory (ConvLSTM) network to categorise human activities inside the indoor localization situation employing smartphones with smartphone inertial sensor data. The accuracy was 73% for nine activities with few samples [30]. This dataset is indoor activity depending on the GPS of the person and designing for tracking the person only. improved convolutional neural networks (CNN) for the use of HAR task with local loss, the authors evaluated their methodology with a group of dataset recognition activities based on global loss, the results were significant the few samples of the dataset as WISDM, UCI HAR but not clearly measurement with others dataset. For healthcare applications, the proposed improved deep learning-based approach may distinguish both specific activities and transitions between two distinct activities of short duration and low frequency. To enhance the HAR identification rate, they first built a deep convolutional neural network (CNN) to extract features from sensor data, and then a long short-term memory (LTSM) network to capture long-term correlations between two events. A wearable sensor-based model is suggested that can reliably distinguish activities and their transitions by combining CNN and LSTM. Based on the HAPT dataset [32], the experimental findings showed that the suggested approach could achieve a recognition rate of up to 95.87% and a recognition rate for transitions of more than 80%. The suggested end-to-end model uses a Deep Neural Network-based model that utilises CNN and Gated Recurrent Units to perform automatic feature extraction and classification of activities. The studies in this paper were conducted utilizing raw data from wearable sensors having no pre-processing, and no customised feature extraction approaches were used. On the UCI-HAR, WISDM, and PAMAP2 datasets, the accuracies were 96.20%, 97.21%, and 95.27%, accordingly [7]. Nonetheless, despite the encouraging results, it is still hard to measure the activity efficacy identification systems, particularly those involving deep learning. The lack of information on data pre-processing, and sometimes even the implementation of AI recognition, might make reproducing reported performances difficult. It is also challenging to evaluate the various ways due to a lack of consistency in the benchmark HAR dataset(s), as well as the classification report and testing time, which makes for an unfair comparison or discussion of recognition methodologies based on wearable sensors. In this work, benchmarking evaluation is proposed to measure the ability of ML algorithms to recognize the human activity with accuracy, classification report and training and execution time. The article [14] depicted a model of a perfect healthcare system. Network technology, sensor technology, and data processing technology are the primary high technologies for the ideal healthcare system. Diagnostic software, digital imaging processing, and electronic health record systems are examples of data processing. Recent IoT research has opened up more opportunities than in the medical field.

3 Benchmarking of ML-HAR in Mobile Health Care

Since the beginning of this decade, ML-HAR has sparked a lot of scientific interest. Researchers are still concerned about building a HAR using ML techniques or AI to develop a secure network that can withstand physical activities. Many studies use various metrics to improve HAR and analyze their findings. The behavior of most machine learning algorithms is parameterized, meaning that it cannot be predicted from the processed data. Random parameters also have a significant impact on the AIDS' models performance [10]. The parameters behavior must then be fine-tuned in order to attain an adequate evaluation. Figure 5 depicts a potential benchmarking methodology for testing and assessing ML-HAR models, as well as related operations.

3.1 Pre-processing method

As per their histogram, the values of various mobile health care attributes have a wide range and are non-distributed. We standardized all attribute values within the interval $[-3, 3]$ to put them on the same scale as below:

$$X = -3 < \min(z_{\max}, \max(z, z_{\min})) < 3 \tag{3.1}$$

$$\text{Where Standardization} : z_i = \frac{x - \mu}{\sigma} \tag{3.2}$$

$$\text{mean} : \mu = \frac{1}{N}\sum_{i=1}^{N}(x_i) \tag{3.3}$$

$$\text{Standard deviation} : \sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2} \tag{3.4}$$

3.2 Benchmarking Evaluation Methodology

The investigation of ML-HAR has continued since the beginning of this decade. As a consequence, it assists in creating a HAR utilizing AI or machine learning approaches that produce a secure network important research issue. Many studies aim to improve HAR and assess their outcomes using various criteria. The behavior of most machine learning algorithms is parameterized, which means that it cannot be predicted from the processed data. Moreover, the random parameters have a considerable impact on the model's performance [10]. As a result, their behavior can be fine-tuned for proper evaluation. Table 1 describes a method to evaluatie the performance of the ML-HAR models and establishing benchmarking results.

Many strategies exist to generate hyperparameters, including Trial and Error as a common strategy [9]. Cross-validation with k-folds is another option. The dataset is separated into training and testing segments in this technique. To test reliability and generalization as well as the effectiveness of the ML-HAR models, these parts are specifically chosen to a particular threshold of training and testing percentages (for instance, 40%-60%, 50%-50% or 60%-40%) that are not visible in the training stage. Note that the training began by adjusting a set of parameters for each algorithm, as described in the next section. Then, the outcome was assessed, and the process was repeated with different parameters. The benchmarking technique and related procedures to evaluatie as well as test the ML-HAR models are shown in Figure 5.

3.3 Evaluation Metrics

Since most real data is difficult to read visually, more quantitative criteria (accuracy, recall, F1, and the confusion matrix) must be utilised to assess a model and determine which classes it is likely to confuse. These are the measures that should be used to evaluate the HAR using performance metrics like precision, accuracy, F-Score, sensitivity (recall), prediction time and training time, as shown below

- **Accuracy** is the proportion of right activity forecasts for TN and TP relative to the number of instances tested.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FN + FP} \tag{3.5}$$

Table 1: The benchmarking ML-HAR Algorithm

♯	**Step**
1	Divide the dataset based on 5-folds cross-validation of training and testing without removing any instance or feature to ensure the test robustness of the ML-HAR models;
2	Turn parameters of an ML-HAR model manually, then train and test the model;
3	Evaluate the results of the model by using the proposed evaluation metrics;
4	Repeat steps 2 and 3 until hyperparameters of the model are obtained based on the best result.
5	Conclude the hyperparameters of the ML-HAR model.
6	Repeat steps 2 to 5 for all ML-HAR models;
7	Present the benchmarking results of the ML-HAR models based on the evaluation metrics;
8	Identify the pros and cons of each model and choose the best model.

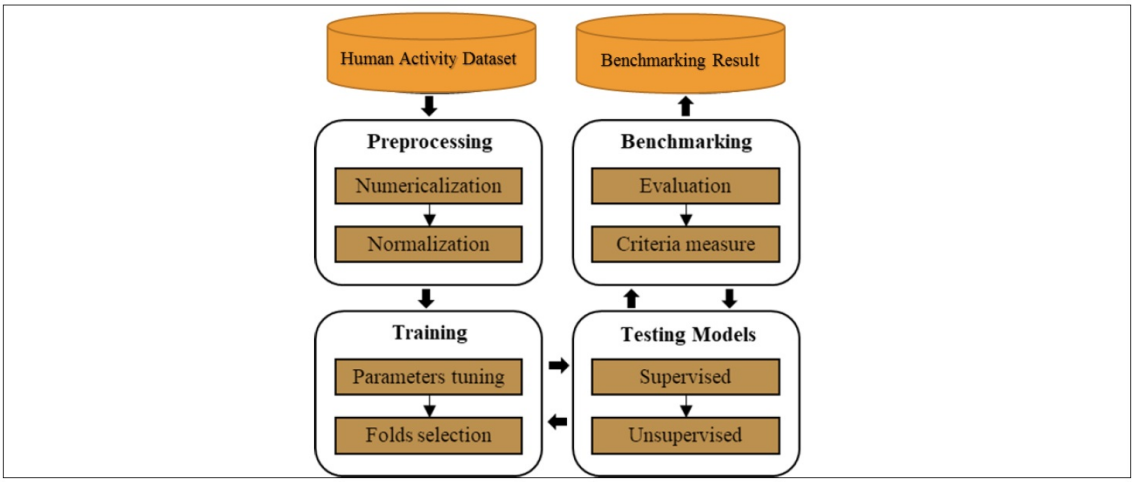

Figure 5: The benchmarking methodology of the ML-HAR

- **Precision** (true positive rate): It is used to calculate the proportion of correctly identified positives, as shown in (3.6):

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3.6}$$

- **Sensitivity** (Recall) calculates the number of correct classifications minus the number of missing items as shown in (3.3):

$$\text{Sensitivity} = \frac{TP}{TP + FN} \tag{3.7}$$

- **F1_Score**: As in (3.4), a measure to achieve a balance between Precision and Recall:

$$\text{F1\_Score} = 2 * \frac{\text{Precision} + \text{Recall}}{\text{Precision} * \text{Recall}} \tag{3.8}$$

- **Training time (T1)**: This indicates how long it took a method to train the entire data set in order to develop the best-fit NIDS model as described in (3.5):

$$\text{T1} = end_{time}^{training} - start_{time}^{training} \tag{3.9}$$

- **Testing time (T2)**: This indicates how long it took an approach to forecast the entire data set as normal or activity, as in (3.6):

$$\text{T2} = end_{time}^{testing} - start_{time}^{testing} \tag{3.10}$$

The number of activities adequately identified as activity is described as the True Positive (TP) measures. False Positive (FP) measures, on the other side, are described as the number of normal connections that are mistakenly identified as activity connections.

3.4 Dataset Description

The MHEALTH (Mobile HEALTH) dataset contains recordings of ten participants' vital signs and body motions while undertaking various physical activities. Sensors attached to the subject's right wrist, chest, and left ankle to track the motion of various body parts, including the rate of turn, acceleration, and magnetic field orientation [3]. Furthermore, "the sensor on the chest can also take 2-lead ECG readings, which may be utilized for basic cardiac monitoring, screening for various arrhythmias, or examining the effects of exercise on the EC" [19]. The M_HEALTH CARE dataset used in the tests contains twelve types of classes. The class name, class label and support of class (number of instances) of the M_HEALTH CARE dataset are shown in Table 2. The number of instances (activities) that are tested is represented by the support ($support = N1 - (N1/N2)$, where $N1$ specifies the number of instances in the input dataset and $N2$ indicates the size of the testing dataset).

Table 2: Samples of the M_Health care testing

Class name	Class label	Support
Standing_still	C1	147
Sitting_and_relaxing	C2	154
Lying_down	C3	120
WALKING	C4	148
Climbing_stairs	C5	133
Waist_bends_forward	C6	124
Frontal_elevation_of_arms	C7	138
Knees_bending_(crouching)	C8	133
Cycling	C9	147
Jogging	C10	93
Running	C11	116
Jump_front_&_back	C12	44

4 Results and Discussion

The benchmarking classification algorithms for a multi-class label HAR M_HEALTH CARE dataset are explained in this section. The benchmarking comprises 10 ML algorithms, seven of which are supervised and three of which are unsupervised. DT, ANN, NB, k-NN, SVM, RF, and XGB have supervised learning algorithms. As discussed in section II, the unsupervised learning algorithms are EM clustering, SOM, and K-means clustering. Several models are proposed to develop some of the ML algorithms. The models are set up by rotating parameters for each method to find the best-fitting parameters based on their results. Those parameters are assumed to have certain initial values for training and testing. Anaconda 3 executes the ML-HAR algorithms in Python 3. OPTIPLEX 3010 DELL with Intel Core i3, 3.60 GHz processor, 4 GB main memory, and 2 GB GPU running Ubuntu 16.04. Depending on the ML algorithms used, the ML-HAR algorithms are put to the test in eight supervised and three unsupervised learning tests. Precision, accuracy, F1-Score, recall, training time (T1), and testing time (T2) are among the evaluation metrics used to assess ML-HAR's performance.

4.1 Results of Supervised Learning Algorithms

The following is a detailed description of the seven ML-HAR outcomes, which are based on proposed ML approaches such as un-supervision and supervision:

4.1.1 ANN:

The ANN classifier testing consists of three models, each of which is described by a set of parameters connected to the training model's creation. The default values for these parameters are activation='relu', alpha=0.0001, batch size='auto', number of hidden layers='12', Optimazer=' '. Table 3 shows the outcomes of the three ANN models. In principle, changing the ANN algorithm's training parameters yields various outcomes. One of the ANN models is able to recognize the $\{C1 - C12\}$ activity, and the other two models low to recognize the Cs activities. Solver=adam, loss= "categorical cross-entropy," and Epoch=100 are the settings for the ANN algorithm's best model. The model achieves 0.90 precision, 90.91 accuracy, 0.89 of f1_score, 0.88 recall, 225.54 seconds of training time, and 0.01 seconds of testing time.

Table 3: Performance results of the ANN algorithm

M_Healthcare Dataset												
Model Setting	**Measurement**	**C1**	**C2**	**C3**	**C4**	**C5**	**C6**	**C7**	**C8**	**C9**	**C10**	**C11**	**C12**
Solver = lbglfs = loss = categorical crosse ntropy	Accuracy	68.34											
	Precision	0.80	0.88	0.99	0.45	0.46	0.45	0.78	0.58	0.89	0.49	0.68	0.44
	Recall	0.99	0.99	1.00	0.43	0.26	0.64	0.76	0.43	0.87	0.50	0.66	0.10
	F1-Score	0.88	0.93	1.00	0.44	0.34	0.53	0.77	0.49	0.88	0.69	0.67	0.16
	Train (s)	79.75											
	Test (s)	0.04											
Solver = adam, loss = categorical crossentropy	Accuracy	90.91											
	Precision	0.99	0.99	1.00	0.88	0.78	0.94	0.92	0.85	0.99	0.82	0.86	0.81
	Recall	1.00	1.00	1.00	0.88	0.76	0.95	0.92	0.93	0.99	0.86	0.86	0.50
	F1-Score	1.00	1.00	1.00	0.88	0.77	0.95	0.92	0.89	0.99	0.84	0.86	0.60
	Train (s)	225.54											
	Test (s)	0.01											
Solver = sgd loss = categorical crossentropy	Accuracy	87.73											
	Precision	0.96	0.99	1.00	0.81	0.74	0.92	0.88	0.82	0.97	0.70	0.80	0.35
	Recall	1.00	1.00	1.00	0.81	0.64	0.90	0.95	0.87	0.98	0.73	0.87	0.21
	F1-Score	0.98	0.99	1.00	0.81	0.96	0.91	0.81	0.84	0.97	0.71	0.83	0.30
	Train (s)	212.750.21											
	Test (s)	0.03											

4.1.2 DT:

The DT classifier testing is made up of three models that are described by parameters relevant to the training models' development. The DT's two configurable parameters are maximum depth and feature type. The two-level value is defined via the max depth, which starts at 1 and 2 and ends at None. This value has an impact on the models' fit. For example, a larger max depth value results in good accuracy, whereas a smaller value causes underfitting, implying poor HAR performance. The features used are "gini" from the Gini impurity criterion or "entropy" from the information gain criterion. Table 4 displays the findings of the three DT models. In practice, changing the DT algorithm's training parameters produces various outcomes. For example, three of the DT models have a different level of nodes, which are able to recognize the $\{C1 - C12\}$, two models less to recognize the Cs activities. The best model of the DT algorithm has the settings of criterion = gini, max depth = None, where the model achieves an accuracy of 96.46, recall of 0.96, the precision of 0.96 F1- score of 0.96, training time of 10.27s and testing time of 0.01s.

4.1.3 k-NN:

Three models are denoted as 1, 2, and 3 k-neighbors in the k-NN classifier testing. Table 5 shows the three k-NN models outcomes. In general, the change in the k value of the k-NN algorithm shows different results. All of the (C1), (C2), (C3), and some other classes are recognized by three of the k-NN models. The 2-NN models less to recognize the Cs of activities. The best average results of all classes are obtained from the 1-NN and 3-NN models. The model achieves a precision of 0.92, an accuracy of 93.52, recall of 0.91, F1_score of 0.92, training time of 3.22s and a testing time of 0.33s.

4.1.4 NB:

There is only one default model in the NB classifier testing. C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11, and C12 activates are recognized by the model. It obtains a precision of 0.72, an accuracy of 70.94, an F1_score of 0.68, a recall of 0.68, a training time of 0.18s and a testing time of 0.1s. Table 6. illustrates the performance of the NB classifier that indicates a good recognition towards activities and consumes considerable time.

4.1.5 RF:

The RF classifier is made up of many subtrees that are built to distinguish different types of activities. The amount of subtrees and maximum tree-level in the RF effect the accuracy rate and time complexity. Hence subtree is used as a parameter in this experiment. The RF classifier testing is made up of two models that are depicted by parameters with respect to the training models' construction. The maximum depth and the number of estimators are the two RF

Table 4: Performance results of the DT algorithm

M_Healthcare Dataset													
Model Setting	**Measurement**	**C1**	**C2**	**C3**	**C4**	**C5**	**C6**	**C7**	**C8**	**C9**	**C10**	**C11**	**C12**
criterion = entropy, max depth=4	Accuracy	52.84											
	Precision	0.36	0.80	1.00	0.00	0.25	0.00	0.45	0.32	0.95	0.71	0.54	0.00
	Recall	0.99	0.88	1.00	0.00	0.54	0.00	0.73	0.17	0.80	0.19	0.63	0.00
	F1-Score	0.53	0.83	1.00	0.00	0.34	0.00	0.56	0.23	0.87	0.30	0.58	0.00
	Train (s)	5.34											
	Test (s)	0.01											
criterion = gini, max depth=4	Accuracy	46.63											
	Precision	0.33	0.44	0.89	0.00	0.00	0.00	0.61	0.24	0.95	0.43	0.42	1.00
	Recall	0.99	0.62	1.00	0.00	0.00	0.00	0.50	0.59	0.80	0.02	0.59	0.16
	F1-Score	0.50	0.51	0.94	0.00	0.00	0.00	0.55	0.34	0.87	0.04	0.49	0.27
	Train (s)	2.32											
	Test (s)	0.01											
criterion = gini, max depth= None	Accuracy	96.46											
	Precision	0.99	1.00	1.00	0.96	0.90	0.97	0.98	0.95	0.99	0.95	0.91	0.95
	Recall	1.00	1.00	1.00	0.95	0.91	0.97	0.98	0.95	0.99	0.93	0.95	0.89
	F1-Score	1.00	1.00	1.00	0.96	0.90	0.97	0.98	0.95	0.99	0.94	0.93	0.92
	Train (s)	10.27											
	Test (s)	0.01											

Table 5: Performance results of the k-NN algorithm

M_Healthcare Dataset													
Model Setting	**Measurement**	**C1**	**C2**	**C3**	**C4**	**C5**	**C6**	**C7**	**C8**	**C9**	**C10**	**C11**	**C12**
K=1	Accuracy	93.32											
	Precision	1.00	1.00	1.00	0.90	0.90	0.93	0.97	0.92	0.98	0.78	0.84	1.00
	Recall	1.00	1.00	1.00	0.93	0.86	0.98	0.99	0.92	0.98	0.85	0.80	0.64
	F1-Score	1.00	1.00	1.00	0.91	0.88	0.96	0.98	0.92	0.98	0.81	0.82	0.78
	Train (s)	3.22											
	Test (s)	0.33											
K=2	Accuracy	93.79											
	Precision	1.00	1.00	1.00	0.86	0.88	0.90	0.97	0.95	0.99	0.73	0.87	1.00
	Recall	1.00	1.00	1.00	0.97	0.85	0.99	0.99	0.88	0.96	0.88	0.72	0.59
	F1-Score	1.00	1.00	1.00	0.91	0.87	0.94	0.98	0.91	0.97	0.80	0.79	0.74
	Train (s)	2.95											
	Test (s)	0.28											
K=3	Accuracy	93.52											
	Precision	1.00	1.00	1.00	0.90	0.90	0.93	0.97	0.92	0.98	0.78	0.84	1.00
	Recall	1.00	1.00	1.00	0.93	0.86	0.98	0.99	0.92	0.98	0.85	0.80	0.64
	F1-Score	1.00	1.00	1.00	0.91	0.88	0.96	0.98	0.92	0.98	0.81	0.82	0.78
	Train (s)	2.93											
	Test (s)	0.34											

Table 6: Performance results of the NB algorithm

M_Healthcare Dataset													
Model Setting	**Measurement**	**C1**	**C2**	**C3**	**C4**	**C5**	**C6**	**C7**	**C8**	**C9**	**C10**	**C11**	**C12**
Non-Parameters	Accuracy	70.94											
	Precision	0.75	0.84	1.00	0.74	0.81	0.57	0.74	0.53	0.88	0.62	0.59	0.65
	Recall	0.97	0.56	1.00	0.65	0.41	0.83	0.82	0.55	0.83	0.66	0.73	0.25
	F1-Score	0.84	0.67	1.00	0.69	0.54	0.68	0.78	0.54	0.86	0.64	0.65	0.37
	Train (s)	0.18											
	Test (s)	0.1											

adjustable parameters (n estimators). Table 7 shows the findings of the two RF models. In general, the change in the training parameters of the RF algorithm shows a few different results. Two of the RF models are able to recognize all activities, one model less to recognize the samples of activities with a low level of nodes. The best model of the RF algorithm possesses the settings of max depth = None and n estimators =100. The model obtains a precision of 0.98, an accuracy of 98.80, f1_score of 0.98, recall of 0.98, training time of 160.3s and testing time of 0.1s. However, it is more time-consuming than other models with high testing time.

4.1.6 SVM:

The SVM classifier testing is made up of two models that are denoted by parameters linked to the training models' development. In such datasets, the SVM model's function is best described as a kernel. Table 8 shows the outcomes of the two SVM models. In general, changing the SVM algorithm's kernel function and training settings yields various results. Both SVM models are able to recognize the different activities with low performance. The best model of the SVM algorithm has the settings of kernel = RBF and max iter=100. The model achieves a precision of 0.38, an accuracy of 42.02, f1_score of 0.38, recall of 0.40, training time of 31.25s and testing time of 0.69s. However, it is acceptable time consuming than different models.

4.1.7 XGB:

The XGB classifier is made up of many subtrees that are built to recognize different kinds of activities. The combined subtree number in the XGB affects the accuracy rate and time complexity. The XGB classifier training and testing with default parameters are represented in the construction of the XGB model. As a result, the model is able to recognize the activities with high performance according to recall, precision, and f-score, as indicated in Table 9. In essence, the model obtains a precision of 0.99, an accuracy of 99.33, an F1_score of 0.99, a recall of 0.99, a training time of 343.97s and a testing time of 0.05s. However, it is more time-consuming to train with low testing time.

Table 7: Performance results of the RF algorithm

M_Healthcare Dataset												
Model Setting	**Measurement**	**C1**	**C2**	**C3**	**C4**	**C5**	**C6**	**C7**	**C8**	**C9**	**C10**	**C11**	**C12**
n estimators =100, max depth=6	Accuracy	78.02											
	Precision	0.60	0.96	1.00	0.70	0.87	0.66	0.93	0.58	0.97	0.77	0.69	1.00
	Recall	0.98	1.00	1.00	0.69	0.38	0.56	0.88	0.65	0.98	0.75	0.89	0.12
	F1-Score	0.74	0.98	1.00	0.70	0.53	0.61	0.90	0.61	0.98	0.76	0.78	0.22
	Train (s)	64.19											
	Test (s)	0.04											
n estimators =100, max depth = None	Accuracy	98.80											
	Precision	1.00	1.00	1.00	0.99	1.00	1.00	1.00	1.00	1.00	0.93	0.94	1.00
	Recall	1.00	1.00	1.00	1.00	0.99	1.00	1.00	1.00	1.00	0.95	0.97	0.85
	F1-Score	1.00	1.00	1.00	0.99	1.00	1.00	1.00	1.00	1.00	0.94	0.95	0.92
	Train (s)	160.3											
	Test (s)	0.1											

4.2 Results of Unsupervised Learning Algorithms

4.2.1 EM:

There is only one default model in the EM classifier testing. Although the model fails to distinguish $\{C1 - C12\}$ activities, it performs poorly in clustering. It obtains a precision of 0.08, an accuracy of 6.35, f1_score of 0.07, recall of 0.07, the training time of 136.06s and a testing time of 0.01s. Table 10 illustrates the performance of the EM classifier that indicates its poor ability recognition to the activities.

4.2.2 k-means:

Only one default model is used in the k-means classifier testing. The model is able to identify C2, C7, C10, C11 and C12 activities and fails to recognize C1, C3, C4, C5, C6, C8 and C19 activities. The model obtains a precision of 0.20, an accuracy of 4.41, f1_score of 0.17, recall of 0.20, the training time of 19.34s and a testing time of 0.10s. Table 11. illustrates the performance of the k-means classifier that indicates its poor recognition rate to the activities.

Table 8: Performance results of the SVM algorithm

M_Healthcare Dataset													
Model Setting	**Measurement**	**C1**	**C2**	**C3**	**C4**	**C5**	**C6**	**C7**	**C8**	**C9**	**C10**	**C11**	**C12**
kernel = RBF, max iter=-6	Accuracy	37.01											
	Precision	0.00	1.00	0.99	0.72	0.04	0.39	0.15	0.23	0.50	0.23	0.78	0.19
	Recall	0.00	0.18	1.00	0.12	0.02	0.10	0.09	0.85	0.98	0.75	0.25	0.07
	F1-Score	0.00	0.31	1.00	0.21	0.03	0.17	0.12	0.36	0.66	0.35	0.38	0.10
	Train (s)	78.2											
	Test (s)	2.62											
kernel = poly, max iter=100	Accuracy	42.02											
	Precision	0.38	0.56	0.98	0.18	0.12	0.39	0.37	0.39	0.75	0.23	0.23	0.05
	Recall	0.35	0.61	1	0.06	0.04	0.18	0.31	0.53	0.94	0.37	0.3	0.16
	F1-Score	0.36	0.58	0.99	0.09	0.06	0.24	0.34	0.45	0.84	0.28	0.26	0.08
	Train (s)	31.25											
	Test (s)	0.69											

Table 9: Performance results of the XGB algorithm

M_Healthcare Dataset													
Model Setting	**Measurement**	**C1**	**C2**	**C3**	**C4**	**C5**	**C6**	**C7**	**C8**	**C9**	**C10**	**C11**	**C12**
Non-Parameters	Accuracy	99.33											
	Precision	1.00	1.00	1.00	1.00	0.99	1.00	1.00	1.00	1.00	0.99	0.95	0.97
	Recall	1.00	1.00	1.00	0.99	1.00	1.00	1.00	1.00	1.00	0.96	0.98	0.97
	F1-Score	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.97	0.96	0.97
	Train (s)	343.97											
	Test (s)	0.05											

4.2.3 SOM:

There is only one default model in the SOM classifier testing. The model is able to recognize all classes except C4, C7, C9, and C11 fail to recognize activities. Furthermore, it has a higher rate of false alarms than previous algorithms. The model obtains a precision of 0.20, an accuracy of 6.35, f1_score of 0.17, recall of 0.20, the training time of 174.06s and a testing time of 0.05. Table 12 illustrates the performance of the SOM classifier that indicates its poor recognition rate to the activities.

Table 10: Performance results of the EM algorithm

M_Healthcare Dataset													
Model Setting	**Measurement**	**C1**	**C2**	**C3**	**C4**	**C5**	**C6**	**C7**	**C8**	**C9**	**C10**	**C11**	**C12**
Non Parameters	Accuracy	6.35											
	Precision	0.00	0.22	0.00	0.00	0.00	0.01	0.00	0.04	0.00	0.68	0.00	0.12
	Recall	0.00	0.25	0.00	0.00	0.00	0.01	0.00	0.04	0.00	0.40	0.00	0.24
	F1-Score	0.00	0.23	0.00	0.00	0.00	0.01	0.00	0.04	0.00	0.51	0.00	0.16
	Train (s)	136.06											
	Test (s)	0.01											

Table 11: Performance results of the k-mean algorithm

M_Healthcare Dataset													
Model Setting	**Measurement**	**C1**	**C2**	**C3**	**C4**	**C5**	**C6**	**C7**	**C8**	**C9**	**C10**	**C11**	**C12**
Non Parameters	Accuracy	4.41											
	Precision	0.00	0.17	0.00	0.00	0.00	0.00	0.13	0.00	0.00	0.48	0.02	0.06
	Recall	0.00	0.25	0.00	0.00	0.00	0.00	0.09	0.00	0.00	0.14	0.03	0.05
	F1-Score	0.00	0.20	0.00	0.00	0.00	0.00	0.11	0.00	0.00	0.21	0.03	0.06
	Train (s)	19.34											
	Test (s)	0.10											

Table 12: Performance results of the SOM algorithm

M_Healthcare Dataset												
Model Setting	**Measurement**	**C1**	**C2**	**C3**	**C4**	**C5**	**C6**	**C7**	**C8**	**C9**	**C10**	**C11**	**C12**
Non Parameters	Accuracy	6.35											
	Precision	0.70	0.22	0.40	0.00	0.25	0.01	0.00	0.04	0.00	0.68	0.00	0.12
	Recall	0.80	0.25	0.44	0.00	0.30	0.01	0.00	0.04	0.00	0.40	0.00	0.24
	F1-Score	0.50	0.23	0.43	0.00	0.25	0.01	0.00	0.04	0.00	0.51	0.00	0.16
	Train (s)	174.06											
	Test (s)	0.05											

4.3 Overall Evaluation

The purpose of this research is to see how effective ML-HAR algorithms are at detecting activity. The testing is done using the M healthcare dataset as a starting point. This study then offers the benchmarking of 10 classification methods, including supervised learning techniques (DT, ANN, NB, k-NN, SVM, RF, and XGB) as well as three unsupervised learning strategies (EM clustering, K-means clustering, and SOM). Several models are available for several of the ML-HAR algorithms. There are 18 models in total that have been tested, with 31 models being described in this paper. In general, when compared to other algorithms, the DT, RF, and ANN classifiers do better at recognizing activities. The supervised learning algorithms exceed the unsupervised learning algorithms in the ML-HAR algorithms' overall performance. XGB is the best-supervised learning algorithm that does not take into account training and testing time, and the ANN is the best-supervised learning algorithm that does take into account training and testing time. Without taking into account the training and testing time, the EM is the best-unsupervised learning method. This total result comprises all of the algorithms' models that have been tested, as well as the models' accuracy and classification report. The overall performance of the 10 ML-HARs evaluated is shown in Table 13.

Table 13: Overall performance results of the ML-ANTDS algorithms

Algorithms	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**T1(s)**	**T2(s)**
ANN	0.90	0.90	0.88	0.89	225.54	0.01
DT	0.96	0.96	0.96	0.96	10.27	0.01
k-NN	0.93	0.92	0.90	0.90	2.95	0.28
NB	0.70	0.72	0.68	0.68	0.18	0.10
RF	0.98	0.98	0.98	0.98	160.3	0.10
SVM	0.42	0.38	0.40	0.38	31.25	0.69
XGB	0.99	0.99	0.99	0.99	343.97	0.05
EM	0.60	0.08	0.078	0.07	136.06	0.01
k-means	0.41	0.071	0.046	0.05	19.34	0.1
SOM	0.63	0.20	0.206	0.17	174.06	0.05

Subsequently, the algorithms are exposed to false alarms because of the overlapping of activities problem. Figure 6 depicts the average accuracy, training time, and testing time of the 10 algorithms. In comparison to other algorithms, this figure confirms our conclusion that the DT, K-NN, FR, ANN, and XGB are the optimum ML-HAR algorithms for recognizing activities.

The ML model's building time that needs to create a trained model out of training data can be used to assess algorithm complexity. The time length should be kept to a minimum so that a trained model can recognize activities in the shortest amount of time possible.

5 Conclusion

This paper evaluates human activities recognition using several machine learning algorithms. On a simple dataset, the performance of associated ML-HAR models for recognizing activities is also evaluated. With a complex dataset, these models demonstrate limits in recognizing novel sorts of actions. Furthermore, the most of related research use accuracy as their primary assessment criterion, overlooking the issue of adjustable parameters in algorithms, preventing them from performing a fair comparison and evaluation of various ML-HARs. To deal with the problem, this research

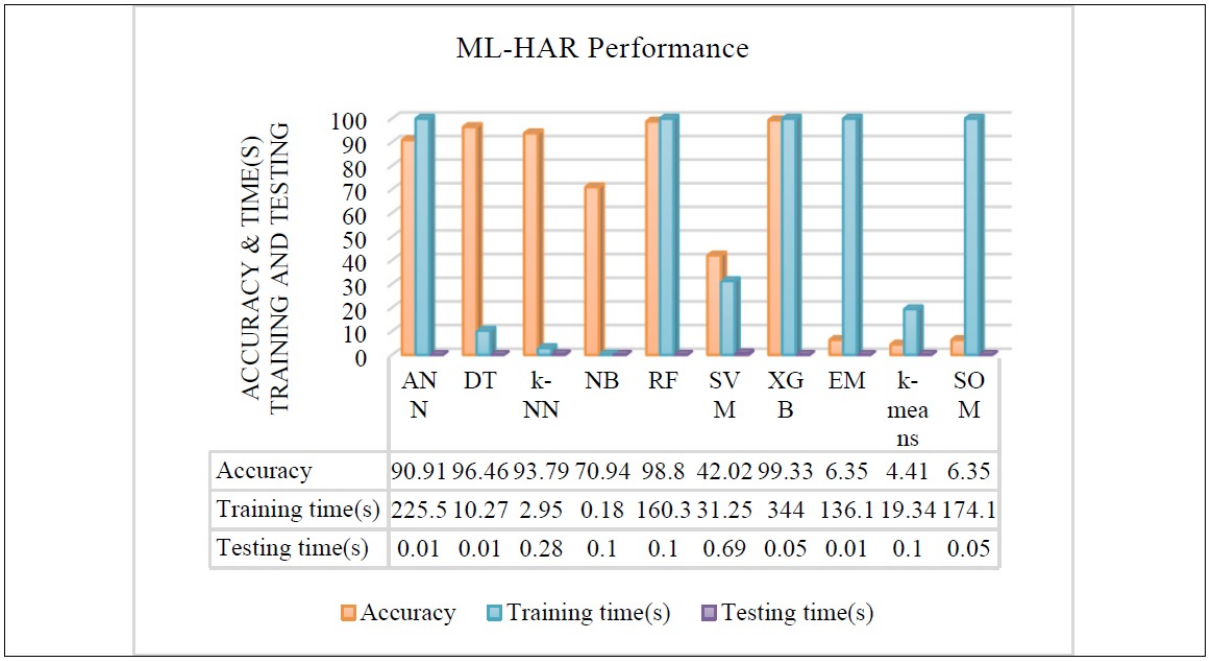

Figure 6: Diagram of components' factor loadings

provides a benchmarking approach that entails many steps and makes use of real data to ensure an accurate assessment of HAR performance using ML algorithms. Note that the assessment considers a variety of factors, including raw human activity information and suggested performance measures. Benchmarking tests employing supervised and unsupervised ML algorithms are also used to analyze the development of effective ML-HAR (for instance, DT, ANNNB, k-NN, SVM, RF, XGB, K-means, EM, and SOM). Human actions from the M-Health care dataset are used to conduct the experiments. The findings of the experiments reveal that there is no single machine learning algorithm capable of recognizing all forms of activities. Standing still, Sitting and relaxing, Lying down, and WALKING are among the most common human activities recognized by the XGB. The DT-HAR, K-NN-HAR, RF-HAR, and ANN-HAR models also have good performance; however, the EM-HAR and SOM-HAR models perform poorly due to their high FN and FP alarms. Researchers can use the suggested benchmarking approach to build a better AIDS model and compare their outcomes to those of this research. Future research should concentrate on assuring the influence of selecting features and taking into account new methodological approaches for constructing the deep learning-HAR model.

References

[1] S.M. Abdulla, N.B. Al-Dabagh and O. Zakaria, *Identify features and parameters to devise an accurate intrusion detection system using artificial neural network*, World Acad. Sci. Eng. Technol. **46** (2010), no. 10, 626-–630.

[2] Y. Alagrash, A. Drebee and N. Zirjawi, *Comparing the area of data mining algorithms in network intrusion detection*, J. Info. Secur. **11** (2020), 1–18.

[3] O. Banos, R. Garcia, J.A. Holgado-Terriza, M. Damas, H. Pomares, I. Rojas and C. Villalonga, *mHealthDroid: a novel framework for agile development of mobile health applications*, Int. Workshop on Ambient Assisted Living, Springer, Cham. 2014, p. 91–98.

[4] X. Chen, M. Ma and A. Liu, *Dynamic power management and adaptive packet size selection for IoT in e-healthcare*, Comput. Electr. Eng. **65** (2018), 357-–375.

[5] K. Chen, D. Zhang, L. Yao, B. Guo, Z. Yu and Y. Liu, *Deep learning for sensor-based human activity recognition: Overview, challenges, and opportunities*, ACM Comput. Surv. **54** (2021), no. 4, 1—40.

[6] T. Cover and P. Hart, *Nearest neighbour pattern classification*, IT **13** (1967), no. 1, 1–27.

[7] N. Dua, S.N. Singh and V.B. Semwal, *Multi-input CNN-GRU based human activity recognition using wearable sensors*, Comput. **103** (2021), no. 7, 1461–1478.

[8] N. Farnaaz and M.A. Jabbar, *Random forest modeling for network intrusion detection system*, In Proc. Comput. Sci. **89** (2016), 213-–217.

[9] A. Gozzoli, *Practical guide to hyperparameters optimization for deep learning models*, FloydHub, 2018.

[10] S. Hamouda, A. Hassan, M.E. Wahed, M. Ail and O. Farouk, *Tuning to optimize SVM approach for breast cancer diagnosis with blood analysis data*, Available SSRN 3537067, (2020).

[11] I. Hanif, *Implementing extreme gradient boosting (XGBoost) classifier to improve customer churn prediction*, Proce. 1st Int. Conf. Statist. Anal., ICSA 2019, Bogor, Indonesia, European Alliance for Innovation, 2019.

[12] M.M. Hassan, S. Huda, M.Z. Uddin, A. Almogren and M. Alrubaian, *Human activity recognition from body sensor data using deep learning*, J. Med. Syst. **42** (2018), no. 6, pp. 1—8.

[13] A.A. Hassan, W.M. Shah, M.F.I. Othman and H.A.H. Hassan, *Evaluate the performance of K-Means and the fuzzy C-Means algorithms to formation balanced clusters in wireless sensor networks*, Int. J. Electr. Comput. Eng. **10** (2020).

[14] J.-S. Jeong, O. Han and Y.-Y. You, *A design characteristics of smart healthcare system as the IoT application*, Indian J. Sci. Technol. **9** (2016), no. 37.

[15] J. Jha and L. Ragha, *Intrusion detection system using support vector machine*, Int. J. Appl. Inf. Syst. **3** (2013), 25-–30.

[16] D. Kaur, *A comparative study of various distance measures for software fault prediction*, arXiv preprint arXiv:1411.7474, 17 (2014), no. 3.

[17] T. Kohonen, *The self-organizing map*, Proc. IEEE, **78** (1990), no. 9, 1464–1480.

[18] D.D. Lewis, *Naive (bayes) at forty: The independence assumption in information retrieval*, Eur. Conf. Machine Learning, 1998, p. 4–15.

[19] Z.K. Maseer, R. Yusof, N. Bahaman, S.A. Mostafa and C.F.M. Foozy, *Benchmarking of machine learning for anomaly based intrusion detection systems in the CICIDS2017 dataset*, IEEE Access **9** (2021), 22351–22370.

[20] M. Mathuria, *Decision tree analysis on j48 algorithm for data mining*, Int. J. Adv. Res. Comput. Sci. Softw. Eng. 3 (2013), no. 6.

[21] T.K. Moon, *The expectation-maximization algorithm*, IEEE Signal Process. Mag. **13** (1996), no. 6, 47—60.

[22] P. Praveen and B. Rama, *A k-means clustering algorithm on numeric data*, Int. J. Pure Appl. Math. 117 (2017), no. 7.

[23] J.R. Quinlan, *Bagging, boosting, and C4. 5.* In Aaai/iaai **1** (1996), 725—730.

[24] A.M. Rahmani, T.N. Gia, B. Negash, A. Anzanpour, I. Azimi, M. Jiang and P. Liljeberg, *Exploiting smart e-health gateways at the edge of healthcare Internet-of-Things: A fog computing approach*, Futur. Gener. Comput. Syst. **78** (2018), 641-–658.

[25] C.A. Ronao and S.-B. Cho, *Human activity recognition with smartphone sensors using deep learning neural networks*, Expert Syst. Appl. **59** (2016), 235-–244.

[26] A. Subasi, D.H. Dammas, R.D. Alghamdi, R.A. Makawi, E.A. Albiety, T. Brahimi and A. Sarirete, *Sensor based human activity recognition using adaboost ensemble classifier*, Proc. Comput. Sci. **140** (2018), 104—111.

[27] V. Subramaniyaswamy, G. Manogaran, R. Logesh, V. Vijayakumar, N. Chilamkurti, D. Malathi and N. Senthilselvan, *An ontology-driven personalized food recommendation in IoT-based healthcare system*, J. Supercomput. **75** (2019), no. 6, 3184—3216.

[28] J. Sun, Y. Fu, S. Li, J. He, C. Xu and L. Tan, *Sequential human activity recognition based on deep convolutional network and extreme learning machine using wearable sensors*, J. Sensors **2018** (2018).

[29] H. Tahir, A. Kanwer and M. Junaid, *Internet of things (IoT): An overview of applications and security issues regarding implementation*, Int. J. Multidiscip. Sci. Eng. **7** (2016), no. 1, 14-–22.

[30] Q. Teng, K. Wang, L. Zhang and J. He, *The layer-wise training convolutional neural networks using local loss for sensor-based human activity recognition*, IEEE Sens. J. **20** (2020), no. 13, 7265—7274.

[31] P. Verma, S.K. Sood and S. Kalra, *Cloud-centric IoT based student healthcare monitoring framework*, J. Ambient Intell. Humaniz. Comput. **9** (2018), no. 5, 1293—1309.

[32] Y. Wang, H. Yu and S. Cang, *A survey on wearable sensor modality centred human activity recognition in health care*, Expert Syst. Appl. **137** (2019), 167–190.

[33] T. Wu, F. Wu, J.-M. Redoute and M.R. Yuce, *An autonomous wireless body area network implementation towards IoT connected healthcare applications*, IEEE Access **5** (2017), 11413—11422.

[34] K. Xia, J. Huang and H. Wang, *LSTM-CNN architecture for human activity recognition*, IEEE Access, 8 (2020), 56855-–56866.

[35] Y. Zhao, R. Yang, G. Chevalier, X. Xu and Z. Zhang, *Deep residual bidir-LSTM for human activity recognition using wearable sensors*, Math. Probl. Eng. **2018** (2018).