Human Activity Recognition through Smartphone Inertial Sensors with ML Approach

Munid Alanazi
Department of Electrical Engineering and Computer Science, Florida Atlantic University, USA | Department of Business Informatics, College of Business, King Khalid University, Saudi Arabia
malanazi2020@fau.edu (corresponding author)

Raghdah Saem Aldahr
Department of Electrical Engineering and Computer Science, Florida Atlantic University, USA | Department of Computer Science, Taibah University, Saudi Arabia

Mohammad Ilyas
Department of Electrical Engineering and Computer Science, Florida Atlantic University, USA

Received: 2 November 2023 | Revised: 27 November 2023 | Accepted: 29 November 2023

ABSTRACT

Human Activity Recognition (HAR) has several applications in healthcare, security, and assisted living systems used in smart homes. The main aim of these applications or systems is to classify body movement read from the built in sensors such as accelerometers and gyroscopes. Some actions could be performed in response to the output of these HAR systems. The number of smartphone users increases, whereas the sensors are widely available in different sizes and shapes (internal or external sensors). Recent advances in sensor technology and machine learning have led researchers to conduct studies on sensor technology such as HAR. HAR systems typically use a combination of sensors, such as accelerometers, gyroscopes, and cameras, to collect images or signal data that can be classified by machine learning algorithms. HAR research has focused on several key challenges including dealing with variability in sensor data, handling missing data or noise, and dealing with large amounts of sensor-generated data. In this work, several machine learning algorithms were tested in predefined settings using the KU-HAR dataset in a series of experiments. Subsequently, various performance metrics were calculated to assess the chosen algorithms' performance. The experimental findings showed that the LightGBM classifier surpassed the other machine learning algorithms in performance metrics, such as accuracy, F1 score, precision, and recall. Although Gradient Boosting has lengthy training time, the other classifiers complete their training in an acceptable time period.

Keywords-Human Activity Recognition (HAR); accelerometer; gyroscope; machine learning; sensors

I. INTRODUCTION

The domain of Human Activity Recognition (HAR) is rapidly expanding, leveraging sensor data to autonomously detect and categorize human behaviors and actions, in numerous applications. The field of wearable computing is also experiencing rapid growth in activity recognition, thanks to sensors embedded in smartphones. Advances in smartphone technology have enabled these built-in sensors to collect time-series data in real-time, paving the way for a variety of applications in health monitoring, sports, gaming, and security. Table I shows that the number of smartphone users in the USA was 307 million in 2022 and is expected to reach 311.8 million in 2023. Every year, there are 4 million new users added to these statistics [1]. Thus, HAR applications will aid smartphone users in terms of health monitoring, sports performance, etc. Human activity tracking based on computer vision has been widely used, but infrastructure support is required to implement such systems, e.g. mounting some cameras in the monitoring areas [2]. Alternatively, inertial sensors available in smartphones—such as accelerometers and gyroscopes—can be worn on the body to measure acceleration and orientation [3]. An inertial sensor is a device that measures acceleration, angular velocity, and, sometimes, magnetic fields. The term encompasses a range of sensors, including accelerometers and gyroscopes [4]. A triaxial sensor specifically refers to a type of inertial sensor that measures these parameters along three orthogonal axes [5]. In essence, all triaxial sensors are inertial sensors, but not all inertial sensors are necessarily triaxial. Some may measure only one or two parameters and may not be oriented along three axes.

HAR based on smartphones is a supervised classification problem in which subjects perform activities to obtain a training dataset [6]. This approach involves collecting sensor data from smartphones worn by individuals during their daily routines. Using supervised learning techniques, the data acquired from smartphones are used to train algorithms that can accurately categorize and recognize different activities performed by users. The success of HAR lies in its ability to bridge the gap between raw sensor information and meaningful insights into human behaviors and movements.

TABLE I. SMARTPHONE USERS IN THE USA [3]

Year	Smartphone users (millions)
2022	307
2023	311.80
2024	316.20
2025	320.40
2026	324.40
2027	328.20
2028	331.80
2029	335.20
2030	338.50
2031	341.70
2032	344.70
2033	347.60

This study makes use of KU-HAR, a publicly available real-world HAR dataset [7], which contains 18 activities collected from 90 individuals. Our research focuses on refining the performance of activity recognition models through feature extraction and ML methods. Through this process, we aim to achieve optimal results, thereby enhancing the accuracy and effectiveness of our classification algorithms. This approach advances our understanding of activity recognition and significantly contributes to the development of robust and precise models for real-world applications. In conclusion, our goals have been met through the presentation of results obtained by employing 7 Machine Learning (ML) classifiers: Gradient Boosting (GB), K-Nearest Neighbors (KNN), Decision Tree (DT), Random Forest (RF), XGBoost, LightGBM, and Catboost. The contributions of this study are summarized as follows:

- First, we explore various relevant research areas for collecting HAR datasets and utilizing different machines as well as deep learning classifiers.
- We analyze the KU-HAR dataset, which includes a variety of dynamic and static activities and a large scale of samples. Additionally, we utilize ML algorithms to assess the classification performance of the HAR system.
- Lastly, we conduct a comprehensive evaluation of our system across various performance metrics such as precision, accuracy, recall, and F1-score. We also calculate the training and prediction durations to provide a thorough understanding of the system's operational efficiency.

II. RELATED WORK

HAR applications are utilized across various research fields. Researchers present HAR applications differently due to the variability in the modality of human activity data [7]. One such dataset was published in [8] (University of California, Irvine (UCI)– ML Repository). This dataset consists of 30 subjects performing 6 activities: standing, sitting, laying, walking downstairs, walking upstairs, and walking. A total of 10,299 samples were collected, each with 561 features in time-series format, using built-in smartphone sensors. This dataset has been widely used by researchers and has achieved high levels of accuracy. In 2013, a proposed system achieved an overall accuracy of 96% on the UCI dataset [9]. The Wireless Sensor Data Mining (WISDM) dataset [10] has become a benchmark in the field of HAR. The dataset consists of 5,424 transformed samples collected from 29 participants who performed 6 activities [10]. The authors trained models using robust algorithms such as J48, Logistic Regression (LR), and Multilayer Perceptron, achieving accuracies of 85.1%, 78.1%, and 91.7%, respectively. In [11], the authors applied the ADABOOST.M1 algorithm using 6 different classifiers: Hoeffding Tree, Decision Stump, Random Tree, J48, RF, and REP Tree on the same dataset, achieving accuracies of 87.84%, 57.31%, 95.69%, 97.83%, 94.44%, and 97.33%, respectively. In [12], the USC-HAD dataset focusing on daily activities was collected to enhance ubiquitous computing. The researchers enlisted 14 subjects to perform 12 different activities, resulting in a total of 840 samples. This dataset includes a diverse range of activities suitable for various applications in activity recognition. In [13], various techniques were employed using a Deep Convolutional Neural Network (DCNN) on the previously mentioned dataset, achieving a high accuracy of 97.01%.

Fall detection systems are essential for reducing healthcare costs and are a leading cause of severe injuries among seniors. The UMAFALL dataset [14] utilizes multisensory data from different body points of each participant, which enhances the performance of automatic fall detection systems. The UMAFALL dataset consists of sample data from 17 subjects using accelerometer, gyroscope, and magnetometer sensors. The authors employed a 200 Hz sampling rate to classify activities into 11 classes. Authors in [15] applied various methods to this dataset and achieved the highest accuracy of 98.49% using Shallow MLP. Additional notable accuracies include 97.8% with RF and 95.87% with KNN. The system in [16] relies on an accelerometer sensor module and collects HAR data using the Microsoft Band 2, positioned on the wrist. The sensors sample data at a frequency of 62 Hz. Time-series data are segmented using sliding windows with a 50% overlap. To capture a full cycle of human activity, the sliding windows are set to a size of 64, covering a time span of approximately 1 s. This experiment achieved an overall accuracy rate of approximately 90% using an RF model. Authors in [17] focused on optimizing classification algorithms for HAR systems that employ wearable devices. The paper takes an in-depth look at the utilization of RF algorithms for feature selection. According to the findings, using RF for this purpose enhances the memory efficiency of the model, particularly for mobile applications, although there is a slight compromise in accuracy. The study reports achieving a high accuracy rate of 92.7% when employing the Support Vector Classifier (SVC). Furthermore, accuracies of 89.99% and 92.79% were achieved using KNN and LR algorithms, respectively. Table II provides a comparative summary of the findings from previous research along with the results obtained from our proposed models.

TABLE II. COMPARISON OF OUR WORK WITH PREVIOUSLY PROPOSED MODELS

Ref.	Year	Dataset	Classes	Activities	ML/DL	Algorithms	Accuracy
[9]	2012	UCI-HAR	6	Standing, sitting, laying, walking_downstairs, walking_upstairs, walking	ML	SVM (MC-SVM)	96%
[10]	2012	WISDM	6	Jogging, walking, downstairs, upstairs, standing, sitting	BOTH	J48, LR, MLP	85.1%, 78.1%, 91.7%
[12]	2012	USC-HAD	12	Walking left, walking forward, walking upstairs, walking right, walking downstairs, jumping, standing, running forward, sitting, sleeping, elevator down, elevator up	DL	DCNN	97.01%
[14]	2017	UMAFALL	11	Climbing stairs down, climbing stairs up, squatting, light jogging, lying down and getting up from a bed, hopping, walking, sitting down and up from a chair, fall activities (forward, backwards, lateral)	BOTH	Shallow MLP, RF, KNN	98.49%, 97.8%, 95.87%
[16]	2017	Not Reported	6	Walking, going upstairs, going downstairs, jumping, running, keeping static	ML	RF	90%
[17]	2022	UCI-HAR	6	Standing, sitting, laying, walking_downstairs, walking_upstairs, walking	ML	KNN, LR, SVC	89.99%, 92.79%, 93.47%
Proposed	2023	KU-HAR	18	Stand, lay, sit, talk-stand, talk-sit, stair-up, lay-stand, stair-down, stand-sit, table-tennis, pick, push-up, jump, sit-up, walk-backward, walk, run, walk-circle	ML	RF, KNN, DT, XGBoosting, GB, LightGBM, CatBoost	94%, 77%, 87%, 95%, 94%, 96%, 91%

III. DATA ANALYSIS AND PRE-PROCESSING

The process of developing a HAR model using a specific dataset involves several key steps. Initially, a thorough understanding of the dataset is required, followed by the crucial task of data cleaning. Subsequently, the application of robust statistical methods is essential to ensure that the study's objectives are met and aligned with predetermined performance metrics. This section provides an overview of the approach used to analyze and pre-process the KU-HAR dataset.

A. Dataset

The KU-HAR Dataset [7] was collected by students from the Electronics and Communication Engineering Department of Khulna University. It consists of 18 different activities performed by individuals, such as stand, lay, sit, etc., and is presented in a time-series format. The dataset consists of three Comma-Separated Values (CSV) files: raw activity data, which include 1945 samples across 18 classes, trimmed and interpolated raw data, which are cleaned of natural noise occurring during the collection process, and a set of 20,750 subsamples extracted from the raw data. We intend to use these subsamples to build our proposed system. This dataset is segmented into 3 s intervals for each corresponding activity, capturing data from each axis (x, y, z) and each sensor (accelerometer and gyroscope). The data were collected from 90 subjects aged between 18 and 34 years using built-in smartphone sensors, namely accelerometers and gyroscopes. Figure 1 presents the frequency distribution of each class label. While the frequency of some class labels is moderately high, other models have comparatively lower. This variance in label distribution is what continues to make the KU-HAR dataset a subject of ongoing research interest, offering opportunities for contributions in both existing and new ML algorithms. Figure 2 represents a sample of accelerometer data for each activity to distinguish between static and dynamic activities. Static activities include standing, sitting, laying, and talk-standing, while dynamic activities encompass actions like walking, jumping, running, etc. Through sampling each activity, we observed that static activities exhibit higher data density compared to dynamic activities, thereby posing challenges for machine classification. For this study, we utilized Python 3 to process the data and develop models. We also employed various existing packages to facilitate the classification and evaluation processes. Exploratory Data Analysis (EDA) was applied to understand and analyze the dataset using statistical techniques and visualizations. This enabled us to discern relationships between features and identify patterns and anomalies. EDA techniques have been employed to ensure that the dataset is clean and ready for classification [18].

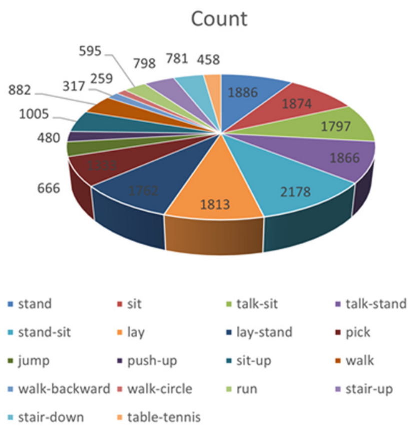

Fig. 1. Class frequency.

B. Description of Activities

In HAR systems, activities are generally categorized into static and dynamic. Static activities typically involve minimal or no movement, whereas dynamic activities entail significant motion or changes in position. Table III lists the static and dynamic activities featured in the KU-HAR dataset. Static activities involve relatively little or no change in position and include actions such as standing, sitting, laying, or talking while being stationary. Dynamic activities, on the other hand, entail significant motion or changes in position, exemplified by walking, running, jumping, or performing exercises like push-ups and sit-ups [19]. Categorizing activities as either static or dynamic is advantageous for designing and training HAR systems. This is because the features and algorithms employed for recognizing these activities may differ based on their motion characteristics.

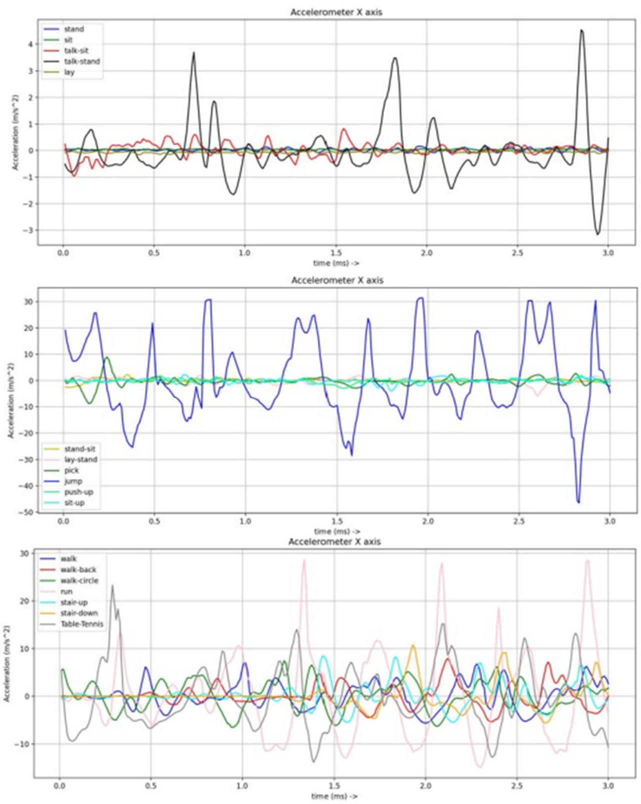

Fig. 2. Accelerometer samples for each activity.

TABLE III. STATIC AND DYNAMIC ACTIVITES IN KU-HAR

Dynamic Activities	Static Activities
Pick	Sit
Jump	Lay
Push-up	Stand-sit
Walk-circle	Talk-stand
Walk	Stand
Walk-backwards	Talk-sit
Sit-up	
Run	
Lay-stand	
Stair-up	
Stair-down	
Table-tennis	

C. The Proposed Model

After reviewing the related work in HAR, we devised the model depicted in Figure 3. Initially, for the KU-HAR dataset, we utilized the 3 s subsamples created by [7] for each activity. We then applied a feature extraction technique to derive a new set of time-domain features, which are detailed later (Table VI). Subsequently, we conducted data cleaning and duplicate checks to ensure the dataset is ready for training. The data were then partitioned into training and testing sets, with a distribution of 70% for training and 30% for testing. We trained our HAR model using this dataset and evaluated its performance using various metrics, which will be elaborated below.

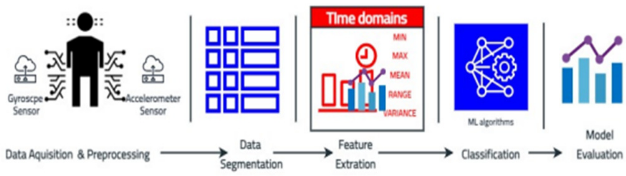

Fig. 3. The proposed model.

D. Evaluation Metrics

We evaluated our work using a variety of metrics, namely precision, accuracy, F1-score, recall, training time, and prediction time. We acknowledge that a high-accuracy model alone does not guarantee the reliability of the classifier's predictions. To assess the consistency of our system's results, we employed additional methods, such as the confusion matrix. Table IV provides a simplified representation of a confusion matrix for a hypothetical 3-class problem, which in our case extends to an 18-class problem [20].

TABLE IV. CONFUSION MATRIX

	Predicted Class		
Actual Class	**Class A**	**Class B**	**Class C**
Class A	TP-A-A	FP-A-B	FP-A-C
Class B	FN-B-A	TP-B-B	FP-B-C
Class C	FN-C-A	FN-C-B	TP-C-C

TP_A_A demonstrates the true positives for Class A (have correctly predicted as Class A), FP_A_B demonstrates the false positives for Class A when predicted as Class B, FN_A_B demonstrates the false negatives for Class A when the actual class is B. The evaluation metrics can be calculated using the formulas outlined in Table V.

- Precision: This metric quantifies the proportion of accurately identified instances of each specific activity among all instances classified under that particular activity. This is evaluated across the 18 different activities.
- Accuracy: This represents the percentage of activity records correctly identified, taking into account all 18 classes.
- F1-score: This composite metric calculates the harmonic mean of recall and precision for each of the 18 classes in the HAR problem, offering a balanced measure of classification performance.
- Recall: This metric measures the proportion of correctly predicted instances for each specific activity relative to all instances of that activity. It provides a comprehensive assessment of the classifier's ability to accurately identify each of the 18 activities.

- Training time: Denotes the time required to train the model using a specific algorithm across the entire dataset.
- Prediction time: Indicates the time taken by a specific algorithm to predict activities across the entire dataset, differentiating among various types of activities.

E. Feature Extraction

From each sample mentioned above, a feature vector is derived. These features are based on standard methods commonly used in HAR studies considering smartphones [21]. For instance, metrics like mean and standard deviation were utilized [22]. A total of 66 features, spanning both frequency and time domains, were extracted to describe each activity window [23, 24]. To derive features from the data, we utilized a window-based approach rather than relying on raw data, which would require classification for each individual data point [25]. Table VI presents the extracted features along with their corresponding mathematical equations.

TABLE V. EVALUATION METRICS

Metric	Mathematical equation
Accuracy (ACC)	$\frac{TP+TN}{TP+TN+FP+FN}$
Precision (Pr)	$\frac{TP}{TP+FP}$
Recall (Rc)	$\frac{TP}{TP+FN}$
F1-Score (F1)	$\frac{2TP}{2TP+FP+FN}$

TABLE VI. MATHMATICAL REPRESENTATION OF THE EXTRACTED FEATURES

Feature	Equation
Mean	$\mu_s = \frac{1}{n}\sum_{i=1}^{n} S_i)$
Max	$\max(S_1, S_2, \ldots, S_n)$
Min	$\min(S_1, S_2, \ldots, S_n)$
Median	$\text{median}(S_1, S_2, \ldots, S_n)$
Standard Deviation (SD)	$\sigma_s = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(S_i - \mu_s)^2}$
Skewness	$\frac{1}{n\sigma_s^3}\sum_{i=1}^{n}(s_i - \mu_s)^3$
Kurtosis	$\frac{1}{n\sigma_s^4}\sum_{i=1}^{n}(s_i - \mu_s)^4)$
Signal magnitude area	$\frac{1}{n}\sum_{i=1}^{n}(\lvert x_i\rvert + \lvert y_i\rvert + \lvert z_i\rvert)$
Interquartile range	$\text{percentile }(s, 75) - \text{percentile }(s, 25)$
Average energy	$e = \frac{1}{N}\sum_{i=1}^{N} w_i^2$
Entropy	$H(X) = -\sum_{i=1}^{n} p(x_i)\log_2 p\,(x_i)$

F. Data Processing and ML Tool

In this experiment, we utilized the Jupyter Notebook running on Python version 3.11. Although the Anaconda platform offers various notebook applications, we chose Python for its efficiency, scalability, and robustness. Additionally, Python offers a wide array of evaluation metrics that aid in assessing the performance of the models we used. To process the data, train the models, and evaluate their performance, we employed several libraries, including Sklearn, Numpy, SciPy, Pandas, CatBoost, LightGBM, and XGBoost [26–32].

G. Data Partitioning

The dataset was divided into separate training and testing subsets in a 70:30 ratio. This separation was conducted using randomization and stratification methods to ensure that all classes were represented in both subsets. Following this division, we employed the training data to construct various classification models. Ultimately, these models were evaluated on the testing dataset to assess their performance [33, 34].

IV. MACHINE LEARNING CLASSIFIERS

This section offers a concise overview of various supervised ML algorithms, highlighting their significance in diverse domains like HAR. The ongoing advancement of cutting-edge technologies underscores the growing need for these algorithms, which are crucial for extracting knowledge from large datasets. In this study, we aim to utilize these ML algorithms to classify the extracted features from the KU-HAR dataset into their corresponding classes. Introduced in 2001, the RF algorithm has gained widespread use for classification and regression tasks. This algorithm involves combining multiple DTs that are generated in a randomized manner and then aggregating their predictions through averaging. It has proven especially effective when dealing with a high number of variables compared to the number of observations [35]. RF is well-suited for HAR applications, as it is an ensemble learning algorithm that can work with various types of datasets, including time series data. One of its primary advantages in the field of HAR is its ability to manage complex, high-dimensional datasets. Moreover, RF is robust to noise and outliers, making it valuable for reliable activity recognition in real-world scenarios [36]. KNN algorithm is a form of supervised learning that classifies new data points based on their proximity to the K closest existing points. The value of K is a significant hyperparameter that dictates how many neighbors should be considered [37]. The algorithm functions by initially calculating the distance between a new data point and every other point in the training dataset. The instances with the shortest distances are then identified as the K nearest neighbors. The new instance is subsequently labeled based on the most common label among its K closest neighbors. One of the most commonly used distance metrics is the Euclidean distance:

$$d(x_i, x_j) = \sqrt{\sum_{l=1}^{m}(x_{il} - x_{jl})^2}$$

DT is a powerful algorithm used for classification and prediction [38]. Essentially, it is a tree-like model that serves as a flowchart, consisting of branches, nodes, and leaves. The algorithm divides a dataset into smaller subsets while simultaneously constructing an associated DT.

LR assumes a linear relationship among the features of a dataset. This categorizes it as a parametric learning algorithm, adhering to a predefined structure for the model's parameters. While LR is typically used for predicting outcomes with continuous values, it can also be adapted for classification tasks. Given that time-series data consist of continuous values, LR could yield favorable results for classification [39]. In our case with 18 classes, the model calculates the probability of an input belonging to a specific class *j*, as expressed by:

$$P(y = j|\boldsymbol{x}) = \frac{e^{x \cdot w_j + b_j}}{\sum_{k=1}^{18} e^{x \cdot w_k + b_k}}$$

GB is a powerful ML technique used for both classification and regression tasks. This ensemble learning approach constructs a predictive model by sequentially combining the outputs of multiple weaker models, often DTs. The overall performance is continually enhanced by applying a loss function at each iteration and optimizing it through the Gradient Descent algorithm [40]. The general mathematical form of the Gradient Descent algorithm is given by:

$$F_m(x) = F_{m-1}(x) + \alpha \cdot h_m(x)$$

XGBoost is a form of supervised learning in the ML domain. It uses boosting techniques to create accurate models. Essentially, it employs DTs for making predictions. The term boosting refers to the construction of a sequence of models, where each new model aims to address the shortcomings of the previous one [41].

LightGBM is a sophisticated ML technique that employs a histogram-based algorithm and a leaf-wise strategy to enhance model accuracy [42]. One of the main advantages of LightGBM is its capability for GPU acceleration, enabling the model to train and make predictions faster than some other algorithms.

CatBoost is a supervised ML algorithm tailored for classification and regression tasks. Its name is derived from Categorical Boosting. The algorithm incorporates advanced techniques to improve predictive model accuracy, making it a valuable tool in the field of ML for structured data analysis.

V. MODEL PERFORMANCE AND DISCUSSION

Table VII presents the results obtained from the various ML classifiers discussed above. After the evaluation of the performance with precision, recall, accuracy, and F1-score, the LightGBM classifier emerges as the top performer, followed by XGBoost, RF, GB, CatBoost, DT, and KNN. Notably, XGBoost, RF, and LightGBM excel in terms of higher accuracy and well-balanced precision and recall. DT, GB, and CatBoost also show commendable performance, while KNN lags slightly in accuracy and other metrics.

During the experiment, we also measured the training and prediction times, as shown in Table VIII. It's evident that KNN requires a significantly longer duration for both training and testing phases. Upon analyzing the training and prediction times across various classifiers, we observed distinct differences in computational efficiency. While KNN is notable for its relatively quick training time, it demands considerably more time for making predictions. In contrast, RF has a lengthier training phase but excels in prediction, delivering quick results once trained. The DT classifier offers a balanced approach with both short training and prediction times, making it an overall efficient choice. Ensemble methods such as GB, XGBoost, LightGBM, and CatBoost require longer training durations due to their intricate ensembles but make up for it with efficient prediction times. This highlights the trade-offs between training and prediction speeds.

TABLE VII. CLASSIFIERS' PERFORMANCE

Classifier	Accuracy	F1 Score	Precision	Recall
KNN	0.94008	0.940068	0.940751	0.94008
RF	0.771084	0.767235	0.771158	0.771084
DT	0.867149	0.867198	0.867774	0.867149
GB	0.94008	0.939989	0.940512	0.94008
XGBoosting	0.952129	0.952057	0.95238	0.952129
LightGBM	0.958072	0.957996	0.958361	0.958072
CatBoost	0.911004	0.910832	0.911173	0.911004

TABLE VIII. TRAINING AND PREDICTION TIMES

Classifier	Training Time (s)	Prediction Time (s)
KNN	0.005913	1.533868
RF	12.077085	0.121668
DT	1.793583	0.003753
GB	584.24576	0.145591
XGBoosting	50.149573	0.07401
LightGBM	10.617264	0.196468
CatBoost	21.677018	0.013404

After feature extraction, which yielded a total of 66 features, we compared the performance of this reduced dataset to the original segmented dataset. We evaluated the various ML classifiers on both datasets and the results are presented in Table IX. The findings indicate that the 66-feature dataset generally outperforms the 1800-feature dataset across all assessed metrics. Not only is the performance higher, but it is also more consistent for classifiers using the 66 features, as evidenced by lower variability. Figure 6 illustrates the difference in accuracy between each model using both the original and the extracted datasets.

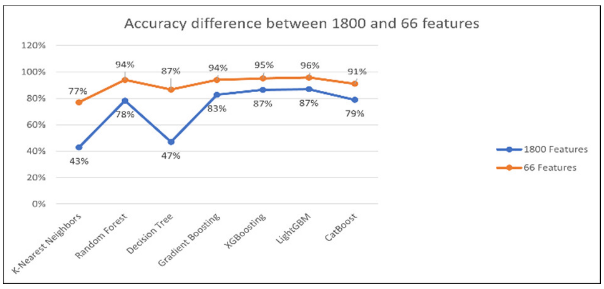

Fig. 6. Model accuracy comparison.

Table X offers a brief analysis of the results derived from the confusion matrix. Firstly, the misclassified labels encompass both FP and FN across all classes. Secondly, the error rate is determined as the ratio of the total number of misclassified samples to the overall count of samples in the confusion matrix. Our findings indicate that LightGBM has the lowest error rate (0.042), thereby making it the most accurate classifier based on this particular metric. Conversely, KNN has the highest error rate (0.228).

We also noted that LightGBM has the highest number of true positives (5964), signifying it correctly classified more instances compared to other classifiers. In contrast, KNN has the lowest number of true positives (4800).

TABLE IX. PERFORMANCE METRICS FOR ORIGINAL AND EXTRACTED DATASET

Classifier	#Features	Accuracy	F1 Score	Precision	Recall
RF	1800	0.783614	0.767101	0.803174	0.783614
RF	66	0.94008	0.940068	0.940751	0.94008
KNN	1800	0.429558	0.401098	0.506774	0.429558
KNN	66	0.771084	0.767235	0.771158	0.771084
DT	1800	0.470843	0.468254	0.467726	0.470843
DT	66	0.867149	0.867198	0.867774	0.867149
GB	1800	0.827952	0.824933	0.830015	0.827952
GB	66	0.94008	0.939989	0.940512	0.94008
XGBoosting	1800	0.865221	0.865221	0.862785	0.865221
XGBoosting	66	0.952129	0.952057	0.95238	0.952129
LightGBM	1800	0.869558	0.865922	0.873129	0.869558
LightGBM	66	0.958072	0.957996	0.958361	0.958072
CatBoost	1800	0.789237	0.785696	0.790512	0.789237
CatBoost	66	0.911004	0.910832	0.911173	0.911004

TABLE X. CONFUSION MATRIX ANALYSIS

Classifier	TP	Total elements	Misclassified	Error rate
Catboost	5671	6225	554	0.089
DT	5354	6225	871	0.14
RF	5845	6225	380	0.061
XGBoost	5550	5848	298	0.051
Light GBM	5964	6223	259	0.042
KNN	4800	6219	1419	0.228
GB	5487	5850	363	0.062

The results of this study can be summarized as follows:

- Based on all four evaluation metrics, the LightGBM classifier emerged as the best model among the ones considered.
- Despite having high training and prediction times, GB performs exceptionally well, showing results quite similar to those of LightGBM.
- XGBoost ranks second in performance, with a training time of 50 s, compared to LightGBM's 10 s.
- RF and LightGBM offer a good balance between reasonable training times and high performance.
- Although DT and KNN have the shortest training times, their overall performance metrics are the lowest.

VI. CONCLUSION

In this paper, a series of experiments in Human Activity Recognition (HAR) was conducted to assess the effectiveness and consistency of seven distinct machine learning algorithms, namely Gradient Boosting, Random Forest, Decision Tree, KNN, XGBoost, CatBoost, and LightGBM. These experiments leverage the publicly available KU-HAR dataset, which encompasses 18 activity classes grouped into static and dynamic categories. The results indicate that the LightGBM algorithm consistently outperforms the others across many performance metrics such as accuracy, F1 score, recall, and precision. While Gradient Boosting demonstrated higher effectiveness but required an extended training period, the remaining algorithms completed their training within a reasonable timeframe.

REFERENCES

[1] "Smartphone users in the US 2009-2040," *Statista.* https://www.statista.com/statistics/201182/forecast-of-smartphone-users-in-the-us/.

[2] N. Gupta and B. B. Agarwal, "Recognition of Suspicious Human Activity in Video Surveillance: A Review," *Engineering, Technology & Applied Science Research*, vol. 13, no. 2, pp. 10529–10534, Apr. 2023, https://doi.org/10.48084/etasr.5739.

[3] M. T. Jan *et al.*, "Methods and Tools for Monitoring Driver's Behavior," in *International Conference on Computational Science and Computational Intelligence*, Las Vegas, NV, USA, Dec. 2022, pp. 1269–1273, https://doi.org/10.1109/CSCI58124.2022.00228.

[4] C. Collette *et al.*, "Review: Inertial Sensors for Low-Frequency Seismic Vibration Measurement," *Bulletin of the Seismological Society of America*, vol. 102, no. 4, pp. 1289–1300, Aug. 2012, https://doi.org/10.1785/0120110223.

[5] A. Bayat, M. Pomplun, and D. A. Tran, "A Study on Human Activity Recognition Using Accelerometer Data from Smartphones," *Procedia Computer Science*, vol. 34, pp. 450–457, Jan. 2014, https://doi.org/10.1016/j.procs.2014.07.009.

[6] M. M. Hassan, Md. Z. Uddin, A. Mohamed, and A. Almogren, "A robust human activity recognition system using smartphone sensors and deep learning," *Future Generation Computer Systems*, vol. 81, pp. 307–313, Apr. 2018, https://doi.org/10.1016/j.future.2017.11.029.

[7] N. Sikder and A.-A. Nahid, "KU-HAR: An open dataset for heterogeneous human activity recognition," *Pattern Recognition Letters*, vol. 146, pp. 46–54, Jun. 2021, https://doi.org/10.1016/j.patrec.2021.02.024.

[8] J. Reyes-Ortiz, D. Anguita, A. Ghio, L. Oneto, and X. Parra, "Human Activity Recognition Using Smartphones." UCI Machine Learning Repository, 2012, https://doi.org/10.24432/C54S4K.

[9] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "A Public Domain Dataset for Human Activity Recognition Using Smartphones," in *European Symposium on Artificial Neural Networks,*

Computational Intelligence and Machine Learning, Brussels, Belgium, Apr. 2013, pp. 437–442.

[10] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," *ACM SIGKDD Explorations Newsletter*, vol. 12, no. 2, pp. 74–82, Nov. 2011, https://doi.org/10.1145/1964897.1964918.

[11] K. Walse, R. Dharaskar, and V. M. Thakare, "A study of human activity recognition using adaboost classifiers on WISDM dataset," *The Institute of Integrative Omics and Applied Biotechnology Journal*, vol. 7, no. 2, pp. 68–76, Jan. 2016.

[12] M. Zhang and A. A. Sawchuk, "USC-HAD: a daily activity dataset for ubiquitous activity recognition using wearable sensors," in *ACM Conference on Ubiquitous Computing*, Pittsburgh, PA, USA, Sep. 2012, pp. 1036–1043, https://doi.org/10.1145/2370216.2370438.

[13] W. Jiang and Z. Yin, "Human Activity Recognition Using Wearable Sensors by Deep Convolutional Neural Networks," in *23rd ACM international conference on Multimedia*, Brisbane, QLD, Australia, Oct. 2015, pp. 1307–1310, https://doi.org/10.1145/2733373.2806333.

[14] E. Casilari, J. A. Santoyo-Ramon, and J. M. Cano-Garcia, "UMAFall: A Multisensor Dataset for the Research on Automatic Fall Detection," *Procedia Computer Science*, vol. 110, pp. 32–39, Jan. 2017, https://doi.org/10.1016/j.procs.2017.06.110.

[15] M. Saleh, M. Abbas, and R. B. Le Jeannès, "FallAllD: An Open Dataset of Human Falls and Activities of Daily Living for Classical and Deep Learning Applications," *IEEE Sensors Journal*, vol. 21, no. 2, pp. 1849–1858, Jan. 2021, https://doi.org/10.1109/JSEN.2020.3018335.

[16] L. Xu, W. Yang, Y. Cao, and Q. Li, "Human activity recognition based on random forests," in *13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery*, Guilin, China, Jul. 2017, pp. 548–553, https://doi.org/10.1109/FSKD.2017.8393329.

[17] D. Adeboye, "Human Activity Recognition in Wearables Using Random Forest Feature Selection to Improve Model Performance." Rochester, NY, USA, Oct. 25, 2022, https://doi.org/10.2139/ssrn.4257072.

[18] F. Hartwig and B. E. Dearing, *Exploratory Data Analysis*. Newcastle, UK: SAGE, 1979.

[19] A. O. Ige and M. H. Mohd Noor, "A survey on unsupervised learning for wearable sensor-based activity recognition," *Applied Soft Computing*, vol. 127, Sep. 2022, Art. no. 109363, https://doi.org/10.1016/j.asoc.2022.109363.

[20] E. Alalwany and I. Mahgoub, "Classification of Normal and Malicious Traffic Based on an Ensemble of Machine Learning for a Vehicle CAN-Network," *Sensors*, vol. 22, no. 23, Jan. 2022, Art. no. 9195, https://doi.org/10.3390/s22239195.

[21] J.-Y. Yang, J.-S. Wang, and Y.-P. Chen, "Using acceleration measurements for activity recognition: An effective learning algorithm for constructing neural classifiers," *Pattern Recognition Letters*, vol. 29, no. 16, pp. 2213–2220, Dec. 2008, https://doi.org/10.1016/j.patrec.2008.08.002.

[22] V. Miskovic and D. Babic, "Implementation of a Flexible Bayesian Classifier for the Assessment of Patient's Activities within a Real-time Personalized Mobile Application," *Engineering, Technology & Applied Science Research*, vol. 7, no. 1, pp. 1405–1412, Feb. 2017, https://doi.org/10.48084/etasr.989.

[23] A. S. Altaher *et al.*, "Detection and localization of Goliath grouper using their low-frequency pulse sounds," *The Journal of the Acoustical Society of America*, vol. 153, no. 4, Apr. 2023, Art. no. 2190, https://doi.org/10.1121/10.0017804.

[24] A. J. Abidalkareem, M. A. Abd, A. K. Ibrahim, H. Zhuang, A. S. Altaher, and A. Muhamed Ali, "Diabetic Retinopathy (DR) Severity Level Classification Using Multimodel Convolutional Neural Networks," in *42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society*, Montreal, QC, Canada, Jul. 2020, pp. 1404–1407, https://doi.org/10.1109/EMBC44109.2020.9175606.

[25] S. O. Slim, A. Atia, M. M. A. Elfattah, and M.-S. M. Mostafa, "Survey on Human Activity Recognition based on Acceleration Data," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 3, pp. 84–98, Jan. 2019, https://doi.org/10.14569/IJACSA.2019.0100311.

[26] F. Pedregosa *et al.*, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, Oct. 2011.

[27] C. R. Harris *et al.*, "Array programming with NumPy," *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020, https://doi.org/10.1038/s41586-020-2649-2.

[28] P. Virtanen *et al.*, "SciPy 1.0: fundamental algorithms for scientific computing in Python," *Nature Methods*, vol. 17, no. 3, pp. 261–272, Mar. 2020, https://doi.org/10.1038/s41592-019-0686-2.

[29] W. McKinney, "Data Structures for Statistical Computing in Python," *Proceedings of the 9th Python in Science Conference*, pp. 56–61, 2010, https://doi.org/10.25080/Majora-92bf1922-00a.

[30] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "CatBoost: unbiased boosting with categorical features," in *32nd International Conference on Neural Information Processing Systems*, Montreal, QC, Canada, Dec. 2018, pp. 6639–6649.

[31] G. Ke *et al.*, "LightGBM: A Highly Efficient Gradient Boosting Decision Tree," in *31st Conference on Neural Information Processing Systems*, Long Beach, CA, USA, Dec. 2017.

[32] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, CA, USA, Aug. 2016, pp. 785–794, https://doi.org/10.1145/2939672.2939785.

[33] H. Cho and S. M. Yoon, "Divide and Conquer-Based 1D CNN Human Activity Recognition Using Test Data Sharpening," *Sensors*, vol. 18, no. 4, Apr. 2018, Art. no. 1055, https://doi.org/10.3390/s18041055.

[34] A. Altaher, Z. Salekshahrezaee, A. Abdollah Zadeh, H. Rafieipour, and A. Altaher, "Using Multi-inception CNN for Face Emotion Recognition," *Journal of Bioengineering Research*, vol. 3, no. 1, pp. 1–12, May 2021, https://doi.org/10.22034/jbr.2021.262544.1037.

[35] G. Biau and E. Scornet, "A random forest guided tour," *TEST*, vol. 25, no. 2, pp. 197–227, Jun. 2016, https://doi.org/10.1007/s11749-016-0481-7.

[36] Y. L. Ng, X. Jiang, Y. Zhang, S. B. Shin, and R. Ning, "Automated Activity Recognition with Gait Positions Using Machine Learning Algorithms," *Engineering, Technology & Applied Science Research*, vol. 9, no. 4, pp. 4554–4560, Aug. 2019, https://doi.org/10.48084/etasr.2952.

[37] P. Paul and T. George, "An effective approach for human activity recognition on smartphone," in *International Conference on Engineering and Technology*, Coimbatore, India, Mar. 2015, pp. 1–3, https://doi.org/10.1109/ICETECH.2015.7275024.

[38] M. Alsaidi, M. T. Jan, A. Altaher, H. Zhuang, and X. Zhu, "Tackling the class imbalanced dermoscopic image classification using data augmentation and GAN," *Multimedia Tools and Applications*, Oct. 2023, https://doi.org/10.1007/s11042-023-17067-1.

[39] E. Bisong, *Building machine learning and deep learning models on Google Cloud Platform*. New York, NY, USA: Springer, 2019.

[40] L. Mason, J. Baxter, P. Bartlett, and M. Frean, "Boosting algorithms as gradient descent," in *12th International Conference on Neural Information Processing Systems*, Cambridge, MA, USA, Dec. 1999, pp. 512–518.

[41] R. Mitchell and E. Frank, "Accelerating the XGBoost algorithm using GPU computing," *PeerJ Computer Science*, vol. 3, Jul. 2017, Art. no. e127, https://doi.org/10.7717/peerj-cs.127.

[42] D. Wang, L. Li, and D. Zhao, "Corporate finance risk prediction based on LightGBM," *Information Sciences*, vol. 602, pp. 259–268, Jul. 2022, https://doi.org/10.1016/j.ins.2022.04.058.