Multimedia Tools and Applications (2024) 83:73233–73272
https://doi.org/10.1007/s11042-023-16423-5

1232: HUMAN-CENTRIC MULTIMEDIA ANALYSIS

In-depth analysis of design & development for sensor-based human activity recognition system

Nurul Amin Choudhury[1] · **Badal Soni**[1]

Received: 17 December 2022 / Revised: 15 July 2023 / Accepted: 24 July 2023 /
Published online: 21 September 2023

Abstract

Human Activity Recognition (HAR) has gained much attention since sensor technology has become more advanced and cost-effective. HAR is a process of identifying the daily living activities of an individual with the help of an efficient learning algorithm and prospective user-generated datasets. This paper addresses the technical advancement and classification of HAR systems in detail. Design issues, future opportunities, recent state-of-the-art related works, and a generic framework for activity recognition are discussed in a comprehensive manner with analytical discussion. Different publicly available datasets with their features and incorporated sensors are also descr-processing techniques with various performance metrics like - Accuracy, F1-score, Precision, Recall, Computational times and evaluation schemes are discussed for the comprehensive understanding of the Activity Recognition Chain (ARC). Different learning algorithms are exploited and compared for learning-based performance comparison. For each specific module of this paper, a compendious number of references is also cited for easy referencing. The main aim of this study is to give the readers an easy hands-on implementation in the field of HAR with verifiable evidence of different design issues.

Keywords Human activity recognition · Shallow learning · Ensemble learning · Deep learning · Activities of daily living and wearable sensors

1 Introduction

Sensor technology has seen a tremendous advancement in shape, size, cost, and performance in the past few years because of the need for current and effective knowledge in various

Nurul Amin Choudhury and Badal Soni contributed equally to this work.

✉ Nurul Amin Choudhury
nurul0400@gmail.com

Badal Soni
badal@cse.nits.ac.in

[1] Department of Computer Science and Engineering, National Institute of Technology Silchar, Cachar, Assam 788010, India

industrial applications. As the Internet-of-Things (IoT) also grows at an accelerated pace, new developments in sensor technologies are following to keep the market demands intact, allowing the industries to move swiftly. The most popular sensors in the market are ambient and wearable sensors like Inertial Measurement Units (IMUs). Researchers [1–5] use these sensors for various applications like gait analysis, motion recognition, fall detection, gesture recognition, and many more. One of the primary factors of sensor-based application development is the rebate of its market price and the improvement in the accuracy of data collection. Modern sensors are less likely to have noise in the collected data if appropriately used than the earlier sensors. This allowed many industries like - medicine and sports to invest heavily in the new sensor technology, which then came up with the further implementation of numerous designs and implementation in the last decade.

With the easy availability of wearable, external and vision-based sensors and their improved overall performance, the field of Human Activity Recognition (HAR) has become one of the trendiest research topics among researchers worldwide [6]. HAR is a process of recognising daily living activities or Activities of Daily Living (ADLs) using a set of sensors and an efficient learning algorithm. It includes activities like - jogging, sitting, standing, running, walking, incidental activities like - falls and many more. These activities are often undertaken while doing daily tasks like - going to the office, playing, exercising, etc. World Health Organisation (WHO) [7] categorised human actions into two categories, i.e., Moderate-intensity based human activities and Vigorous-intensity based human activities. Moderate-intensity-based human activities require less human energy to perform, like - walking, standing, eating and many more. On the other hand, vigorous-intensity-based human activities require a very high amount of energy than moderate-intensity-based human activities. It includes the activities like - running, jumping, jogging, etc.

Performing HAR with all the available sensors in the market is a complex task as there is no clear way of associating raw sensor data to a specific activity in a direct way. This paper address this problem by analysing various state-of-the-art HAR papers and laying a comprehensive review on its design, development and implementation techniques.

The main novelties, innovation and contributions of this paper are as follows -

1. This paper extensively explores and synthesises the recent state-of-the-art Human Activity Recognition (HAR) works, furnishes comprehensive insights into their proposed methodologies and highlights key advancements in the field of sensor-based HAR systems.
2. A comprehensive description of popular publicly available sensor-based HAR datasets is inferred that offers detailed statistics and facilitates easy referencing for researchers and practitioners in the field.
3. A detailed, comprehensive discussion of various tools and techniques required for developing sensor-based HAR systems is provided. Further, potential pros and cons associated with the development chain are exploited and inferred for thorough understanding.
4. A wide range of classifiers, sensors, and the scope of HAR based on their operations, inference capability and incorporated methodologies is categorised for improved comprehension. It offers a comprehensive and structured understanding of the different components involved in HAR, providing researchers with valuable insights for selecting suitable techniques and approaches for sensor-specific HAR applications.
5. A detailed empirical experiment is conducted using wearable sensors for inbuilt smartphone sensor-based HAR, involving dataset collection and a processing module. The investigation delivers valuable insights to readers concerning the experimental implementation of HAR systems using various learning algorithms and processing tools, enabling

them to gain an in-depth understanding and assemble reported decisions in developing the HAR framework.

6. Potential future research gaps and scopes in HAR systems developments are highlighted. The identified research gaps and scopes are discussed in detail, facilitating the subsequent development of HAR systems and their applications.

This paper is divided into multiple sections and sub-sections based on the recent and earlier methods of the HAR system and its adopted set of learning classifiers and sensor technology. The first Section 1 - introduces the readers to the field of HAR with critical advancements in sensor technology, HAR research trends, scope and significance, and key research gaps. Section 2 describes various HAR systems based on sensor choices and learning algorithms. Section 3 describes the latest research trends in the HAR field with a brief outline of datasets available for performing experiments. Section 4 summarizes the design issue for developing an efficient HAR system, and Section 5 shows a practical implementation of an optimal HAR system with a detailed analytical discussion. Finally, Section 6 firmly outlines all the paper's findings and conclusion.

1.1 Scope and significance

HAR has been explored aggressively in recent years and has seen vast scope and significance over multiple domains. It has many applications, from building a small IMU device-based end-to-end HAR system to building a full-fledged complete HAR system using multiple sensors. A few of the major and popular scopes of HAR systems are shown in Fig. 1.

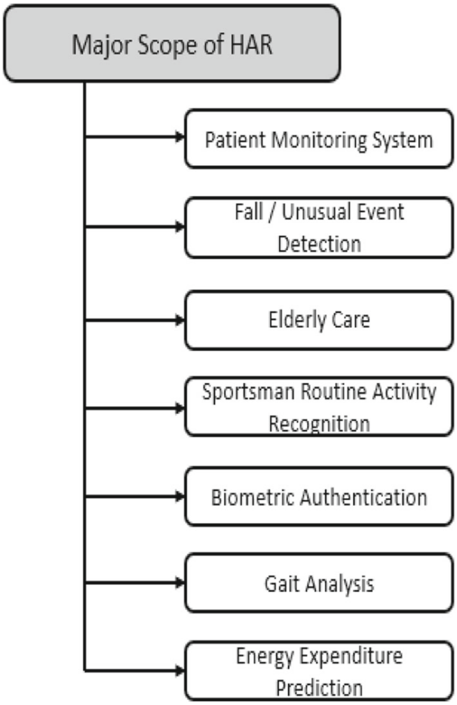

Major Scope of HAR
Patient Monitoring System
Fall / Unusual Event Detection
Elderly Care
Sportsman Routine Activity Recognition
Biometric Authentication
Gait Analysis
Energy Expenditure Prediction

Fig. 1 Major Scope of HAR system

1.2 Prospective research gap for HAR

Numerous work has already been done in the field of HAR by manipulating different public datasets, exploiting various learning algorithms, feature engineering techniques, and their deployment. But most methods lack automatic feature engineering techniques, real-time online HAR, domain adaptation problems, etc. We analyzed a few prospective research gaps in HAR system development, which are as follows-

1. Domain shift analysis and handling for sensor-based multi-featured HAR systems for efficient domain adaptation for robust performance in real-time.
2. Automatic hyperparameter tuning of deep learning algorithms based on dataset features, number of instances and incorporated sensors.
3. Generating HAR dataset in an uncontrolled environment with more activity classes like - complex, secondary and group activities.
4. Dynamic feature engineering for shallow and ensemble learning approach for limiting manual feature extraction and selection based on the incorporated sensors and their features.
5. Hybrid deep and ensemble models for extracting higher level features and better model generalization capabilities towards efficient and robust activity classification.
6. Classifiers incorporated with an online automated class imbalance handling framework for statistical feature-based HAR dataset to handle class imbalance by augmenting potential data instances for activity classification.
7. Lightweight deep learning models towards HAR systems for efficient deployment of classifiers in small devices like - smartwatches and edge devices for fast human activity classification in real-time.
8. Context-aware HAR system for classifying complex ADLs like - different swimming types and underwater activities, etc.
9. A dynamic mechanism for handling unlabelled HAR datasets with approaches like - neighbour analysis and unsupervised clustering models.
10. Explainable HAR system with context-based activity classification using hybrid models for efficient activity classification and online human activity recognition.

2 Classification of HAR

HAR system can be categorised based on its incorporated sensor types and its classification algorithms. This section comprehensively discusses the two aspects of the HAR system. Section ’II-A’ describes the classification of the HAR system based on its incorporated sensor technology, and section ’II-B’ discuss the HAR system based on its classification models.

2.1 Taxonomy of Sensor used in HAR System

Researchers around the globe classify HAR into different types based on their incorporated sensor types or the type of machine learning algorithms they incorporate. Upon organising the state-of-the-art HAR system based on its used sensors, it can be categorised into four major classes, which are as follows [8–11] -

- ***Wearable Sensor-based HAR:*** Wearable sensor-based HAR is the most exploited type of HAR system where the data is collected with the help of wearable sensors like - accelerometer, gyroscope, magnetometer, etc. The sensors are mounted on one or multiple locations of the human body with the help of proper mounting accessories. It is the most popular approach among practitioners [12–16] because of its overall good performance and easy availability of sensors in the market. Also, since all the raw data collected via wearable sensors are numeric, it does not invade the privacy of the users.

- ***Extrinsic Sensor-based HAR:*** Another way of performing HAR is by using extrinsic ambient-based sensors for data collection and an efficient learning algorithm for activity classification [17–19]. Often, wearable technology like in [20] is also incorporated with extrinsic sensors as it will become hard to segregate the class of various ADLs and will yield bad results. It is the least popular method of performing HAR among researchers as it does not promise good performance results.
- ***Vision-based HAR:*** Many research enthusiasts [21–25] make use of a camera module for developing HAR system. The idea is to take multiple images of various human activities and store them as a dataset in a local processing server or cloud. Each image or series of images (video) is labelled with an activity name, and the data is kept in the form of learning and validation. Generally, deep learning algorithms like - Convolutional Neural Network - CNN and hybrid models like CNN-LSTM are widely used for vision-based GAR because of their automatic feature engineering capabilities and ability to extract higher-level features from images. But it is not very popular among users since it invades user privacy.
- ***Hybrid Approach-based HAR:*** Finally, the last way of performing HAR is by using the combination of wearable, extrinsic or vision-based sensors [26–29]. This method is also trendy because of its good performance and tendency to classify complex activities like - group activities and object interaction as well. Here, different sets of learning models are also combined to handle different types of data, and in general, it takes substantially more computational time for model training and validation (Fig. 2).

All the above types of HAR systems are either offline [4, 30–32] or real-time (online) [33–36] in nature. There is no need for real-time data processing in offline or non-real-time HAR systems. The collected data is stored and processed locally to get the performance metrics. Researchers tend to develop their HAR system offline because of its simplicity and

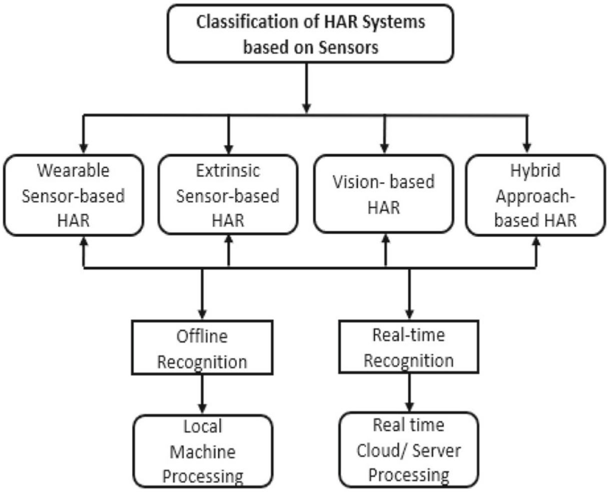

Classification of HAR Systems based on Sensors
Wearable Sensor-based HAR
Extrinsic Sensor-based HAR
Vision- based HAR
Hybrid Approach-based HAR
Offline Recognition
Real-time Recognition
Local Machine Processing
Real time Cloud/ Server Processing

Fig. 2 Classification of HAR on the basis of Sensors

easy availability of numerous datasets publicly. On the other hand, real-time HAR system development is costly and minimally explored in the research field. Here, real-time data analysis is done on the collected data from sensors and server architecture. Whenever a subject performs an ADL, the recognition is done with the help of a trained model that is stored in the cloud or server. But because of its heavy bandwidth usage and real-time cloud or server requirements, this architecture becomes very expensive and is significantly less practised by the researchers.

2.2 Taxonomy of learning algorithms used in HAR system

Numerous learning algorithms are used for recognising different human activities, and one can broadly categorise them into four types as shown in Fig. 3.

1. ***Shallow Learning Algorithms:*** Algorithms which learn from their pre-defined set of rules are termed shallow learning algorithms. Here, the user needs to extract features explicitly and must have domain expertise on the incorporated dataset and type of data. Detailed exploratory data analysis is always needed in this type of system. Usually, shallow learning algorithms are computationally much more efficient than the deep and ensemble

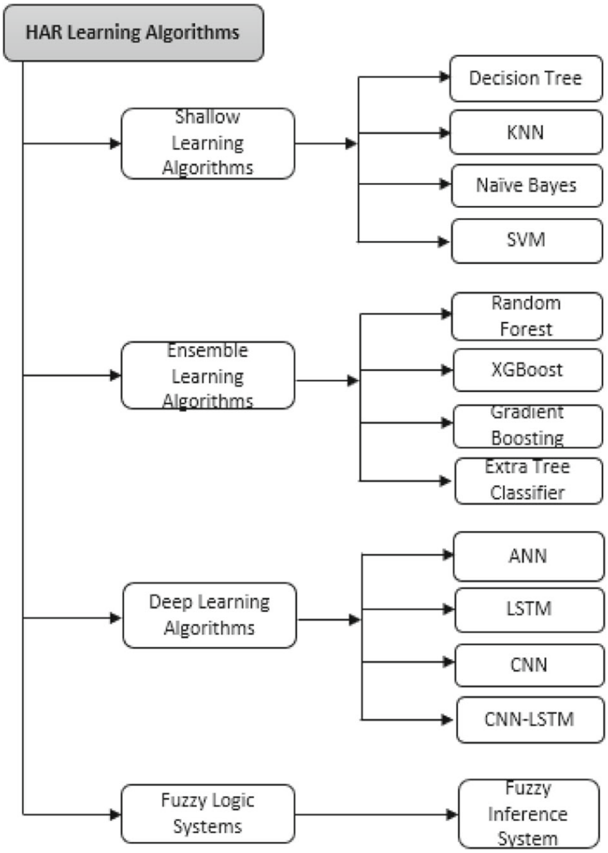


Fig. 3 Frequently used algorithms in HAR system development

approaches because of their simple model training using uninvolved mathematical functions. Researchers [37–40] use this approach because of its simplicity, cost-effectiveness and decent performance.

2. ***Ensemble Learning Algorithms:*** Ensemble approaches are the extension of shallow learning algorithms. Generally, they perform better in terms of overall performance matrices and have significantly less bias because of their multiple base model generation and iterative model testing [30, 41–43]. There are two types of ensemble learning algorithms - *bagging* and *boosting*, where numerous base learners are created independently, and the final result is calculated with the help of result aggregation techniques like - Max Voting, Averaging, Weighted Averaging, etc.
3. ***Deep Learning Algorithms:*** Deep Learning algorithms are widely popular for HAR systems because of their efficient performance and tendency to perform automatic feature extraction [44–46]. Human intervention for feature engineering is not needed with deep learning algorithms, as the models are algorithmically designed to extract the features on their own. But the major disadvantage of deep learning models is high computational time and complex parameter tuning, which is generally very hard to overcome. Also, these models are data-hungry in nature and require huge instances of data for model training for optimal model generalization.
4. ***Fuzzy Inference Systems:*** A fuzzy inference system is a rule-based system where a crisp input is given to a fuzzy decision-making module, and a crisp output is processed from it. First, a fuzzification unit converts the crisp inputs into fuzzy quantities with the help of membership functions, and these fuzzy quantities are then passed to Decision - Making Unit which decides the output by operating on the rules. Upon getting the fuzzy result, the crisp output is calculated with the help of the defuzzification unit and knowledge base. Researchers associate different rules with different human activities and classify ADLs with the fuzzy input quantities, and the assigned rules [47–49]. This method is fast but fails to perform significantly better than newer approaches.

3 State-of-the-art related works

In this section, we studied and analysed the latest state-of-the-art HAR models with different applications and presented them briefly. A synopsis is also attached at the end of this section to get the latest practices on various HAR systems.

Fu et al. [63] collected the data from an IMU and an air pressure sensor at the sampling rate of 20 Hz. Multiple features (a total of 19) were extracted from the raw data that they collected. The data segmentation is done to get the different data segments in the specified period. Upon analysis using their proposed Joint Probability Distribution Adaptation with improved pseudo-labels (IPL-JPDA) method that extracts and generalizes the features based on the target labels and other machine learning methods, they achieved an accuracy of 93.21% and accuracy of 79% ± 91% using different models. In [64], the authors proposed a biometric identification system using HAR and deep learning models. They incorporated two publicly available datasets' named UCI Human Activity Recognition Dataset (UCI HAR) [65] and USC Human Activity Dataset (USC HAD) [66] for their experiment, which consists of different aged group data collected at the sampling rates of 50HZ and 100 HZ, respectively. Traditional data pre-processing routines like handling missing values, noise, special symbol removal and normalization of data for getting the standard scale data were incorporated. The proposed framework first splits the dataset into a training and testing set and recognises the different human activities. Upon successful recognition of human activities, the training set is used for user identification retrieval and tested with the test set. All the performance metrics were iteratively recorded, and they managed to outperform the previous Random projection method [67] with an average performance gain of 1.41%.

Yang et al. [68] proposed a deep learning-based HAR system which deals with the problem of the multi-subject based-HAR dataset. They identified that the dataset having different subjects performed poorly during activity classification. So they integrated activity graphs which align the neighbourhood signals using width and height and deep CNN for automatic feature selection and classification. Upon classification, they boosted the accuracy by 5% and managed to outperform previous benchmarks. In [32, 69], the authors proposed HAR in the wild with portable wearable sensors and a behavioural HAR recognition system. The dataset has different activity contexts [69] and behaviours in the case of [32], like the sitting activity has the context of watching tv, meeting, etc. Because of the included secondary context activities, the overall model became complex and achieved an accuracy of 89.43% by [69]. Siirtola et al. [70] also proposed a context-aware incremental learning-based method for personalized human activity recognition systems using ensemble models. Choudhury et al. [30] proposed a novel physique-based human activity recognition system where they recognised different human activities based on physical similarities and ensemble learning. They found that combining the data of only similar subjects into one dataset will result in efficient dataset generation and will be computationally less expensive than the traditional HAR system. Upon testing the proposed model, they achieved an average accuracy of 99%, and individual activity-wise performance was better than the conventional practice.

In [45, 46, 83–85] the authors used CNN and LSTM for activity classification. Khan et al. [83] identified that the main limitation of HAR is the lack of a dataset having a good number of class labels. They generated their dataset with 12 different classes and used CNN for automatic feature engineering. They managed to achieve an accuracy of 90.89% with the hybrid CNN-LSTM approach. But [46, 85–87] identified that deep learning approaches like - CNN are very good for automatic feature analysis. In [46], the authors used a four-layer multi CNN-LSTM network for HAR and managed to achieve an average accuracy gain of 2.24% over previous methods. Whereas in [87], the authors used multiple public datasets without handcrafted features for testing their model and achieved the average accuracy between 95 ± 97. Kwon et al. [88] also proposed an automatic feature engineering technique with on-body accelerometry from videos of HAR. Their approach automatically converts the videos of various human activities into virtual streams of IMU data that represent the human body's accelerometry.

Choudhury et al. [62] proposed an adaptive batch size-based hybrid CNN-LSTM framework for a smartphone sensor-based HAR system. The framework combines Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) models with an adaptive batch size selection method to improve recognition accuracy and efficiency. The authors evaluate the proposed framework on three datasets and compare its performance with several state-of-the-art methods. The results show that the proposed method outperforms the existing methods in terms of recognition accuracy and computational efficiency. The authors in [89] incorporated multiple machine learning approaches and laid a comparative analysis for human activity recognition using smartphone sensor-based data. The study utilized data collected from an accelerometer, gyroscope, magnetometer and GPS sensors to recognize human activities in real-life scenarios. Further, their work intercepts that the window size and gyroscope sensor data are not prominent for ADL classification. Panja et al. [90] proposed a hybrid feature selection pipeline for recognizing human activities through smartphone sensors. Their proposed framework incorporates clustering and a genetic algorithm-based method to identify prominent features from the global feature set. Using multiple shallow and ensemble learning approaches, they classified different ADLs solely on the selected features.

Yadav et al. [91] proposed a secure privacy-preserving HAR system using WiFi channel state information. The proposed neural network architecture - CSITime model has three levels of abstraction. Instead of going with the ensemble approach like in [92] where they achieve optimal results, the authors adhered to the single model for reducing the computational time of their model. Three different datasets were incorporated in the experiment: *ARIL, StanWiFi,* and *SignFi*, and they managed to outperform all the previous state-of-the-art works with a good performance margin. The achieved accuracies are 98.20%, 98%, and 95.42% w.r.t. ARIL, StanWiFi, and SignFi datasets. In [93], the authors proposed an adaptive CNN-based, energy-efficient HAR system for low-power edge devices. Their model uses an output block predictor to select a portion of the baseline architecture during the inference phase, saving energy, memory, and overall computational cost. Similarly, Lou et al. [94] deployed a binarized neural network for low-power edge devices. The main scope of their experiment is to get faster computational times, lower network latency and efficient memory utilization. With a new quasi-automated method of classifying human activities, Taylor et al. [95] created a dataset with Software define Radio signals (SDRs). Using the dataset and RF, they achieved 96.70% accuracy, but the only drawback was the lack of multi-class labels in their experiment.

Gua et al. [96], proposed a nested ensemble framework for handling HAR with class imbalance problem. Khaled et al. [97] also proposed SMOTE method-based mechanism for controlling the class imbalance problem. Both of them managed to achieve efficient results with different publicly available datasets. In [98], the authors handled the class imbalance with a cost-sensitive hybrid ensemble model. The proposed model modified the weights of minority classes and merged all the weights of ensemble base learners with a stacked generalization approach. Upon classification, they achieved 70.8% accuracy with a dataset of 28-class labels. Hamad et al. [99] proposed a joint diverse temporal learning approach using LSTM and 1D-CNN models to improve activity recognition with class imbalance. They also found out that deep learning models do not always promise optimal results with imbalanced datasets.

The authors in [100] compared multiple shallow and ensemble learning approaches for an inbuilt-smartphone sensor-based HAR system. The result analysis incorporating state-of-the-art data pre-processing shows that the ensemble learning models are significantly better for recognising ADLs. Dua et al. [101] describes a new deep learning model for recognizing human activities using sensor data from raw time-series data. The proposed model combines CNN and Gated Recurrent Units (GRUs) that are inspired by the Inception network architecture. The model was trained and evaluated on several benchmark datasets, and the results demonstrate that the proposed approach outperforms existing state-of-the-art models in terms of accuracy and efficiency.

The authors in [102] developed a robust CNN architecture called Temporal Convolutional 3D Network (T-C3D) for HAR for vision-based datasets. T-C3D addresses the challenge of high computational cost in existing approaches by incorporating a hierarchical multi-granularity learning approach and a residual 3D CNN to capture spatial information. Further, a temporal encoding scheme is also proposed for the temporal dynamics of human actions in video frames. The proposed model achieved average accuracies of above 91.5% and outperformed the benchmark models with optimal performance margins. Choudhury et al. [103] developed an efficient CNN-LSTM model for the HAR system using smartphone sensor data. They used minimal data pre-processing on raw sensor data and classified ADLs using the features extracted by CNN and LSTM layers. Upon classification, they managed to achieve an accuracy of 98.69%.

Li et al. [104] used a radar-based HAR dataset for recognising ADLs. The incorporated dataset has sparse activity labels, which is very hard for models to generalise and extract efficient performance. The proposed Joint Domain and Semantic Transfer Learning (JDS-TL) model successfully extracted new features from the spares dataset and managed to achieve an average accuracy of 87.6 %, where only 10% of the data was labelled. In [105], the authors proposed a teacher-student-based self-learning method for augmenting labelled and unlabelled data. Upon classification with their approach, they managed to get a 12% performance boost over the previous models. To encounter the problem of activity labelling, the authors in [106] proposed automatic activity annotation using a pre-trained model. They trained the proposed model in multiple real-life conditions to overcome the physical biases, which in turn increases the activity recognition rate.

Chen et al. [107] proposed a Masked Pseudo-Labeling AutoEncoder (MAPLE) pipeline for efficient HAR with sparsely labelled datasets. The proposed model addresses the limitation of the need for large annotated data and computationally expensive networks. The framework leverages a Decoupled spatial-temporal TransFormer (DestFormer) as the core feature extractor that efficiently learns long-term and short-term features by decoupling the spatial and temporal dimensions of the point cloud videos. Lui et al. [108] introduce a real-time action recognition architecture named Temporal Convolutional 3D Network (T-C3D) to learn video action representations by adopting a hierarchical multi-granularity approach. It combines a residual 3D CNN to capture the formation of frames and motion signals between consecutive frames. The proposed model outperformed the benchmark model with optimal accuracy margin and inference time.

Synopsis*:* All the works briefly described above are the latest trends in HAR system expansion and deployment. Researchers used different modes of feature engineering, set of sensors, public datasets, learning algorithms, etc., to develop an efficient HAR system. For the future development scope, we have briefly described the latest work in Table 1, state-of-the-art HAR works in Table 2 and the publicly available dataset in Table 3.

Further, previous works such as [109–112] have conducted comprehensive reviews of publicly available datasets for sensor-based human activity recognition (HAR) systems. However, this paper provides a more extensive analysis of the various types of sensors utilized in HAR and a more detailed review of the different learning approaches employed in this context. While most review papers provide a comparative study of tools and techniques for developing an efficient HAR system, they do not present a step-by-step procedure from data collection to activity recognition in an easy-to-understand manner. This paper addresses this issue by showcasing an efficient activity recognition chain that utilizes the inbuilt smartphone sensor module and various learning algorithms for HAR. Furthermore, our paper exploits our own generated dataset with public datasets like - mHealth to provide a comprehensive understanding of the human activity recognition chain and detailed analytical discussion on various performance measures, sensor choices and mounting locations.

Table 1 Sate-of-art HAR works using wearable sensors and model architectures

Work	Proposed Architecture	Issues Handled	Dataset Used	Future Scope
Jain et al. [50]	COLLOSOL: Collaborative self-supervised learning for HAR system.	HAR system with multi-class unlabelled dataset with feature similarity embedding and data saliency.	RealWorld Opportunity PAMAP2- Locomotion PAMAP2-ADL	Ensuring data privacy and use of constructive classifiers.
Lattanzi et al. [51]	Analyzing the effect of ANN in the tiny wearable system for HAR.	A comparable study was made for different neural networks in HAR systems for quick result optimization.	RealWorld	Improvement of early stopping and dropout criteria for better computational efficiency.
Xu et al. [52]	A wearable textile-based sensor array for HAR system.	Incorporated a new and efficient textile-based wearable sensor for data collection and HAR system.	Own generated	Improving the performance of the system by incorporating hybrid deep learning models.
Ghosal et al. [53]	A fuzzy ensemble approach using three deep learning models for HAR.	Handles the issue of miss classification by penalizing the respective activity class.	mHealth USC-HAR WHARF Opportunity	Reduction of computational time by reducing the training iterations, early stopping and dropouts.
Rashid et al. [54]	Adaptive CNN for energy efficient HAR system in low power edge devices.	Formulated a lightweight CNN architecture using output block predictor for efficient energy consumption.	W-HAR Opportunity	Optimizing hyperparameters and domain shift adaption for getting better performance results.
Han et al. [55]	Heterogeneous filter-based CNN architecture for efficient human activity recognition.	Handled the issue of weakly labelled activity classes on the HAR dataset by optimizing the filter of the deep CNN framework.	Opportunity PAMAP2 UCI-HAR USC-HAR	Inclusion of user and data privacy with large dataset handling capability.
Alsaify et al. [56]	Five-stage CSI-based HAR system for multi-environment activity recognition.	Detailed handcrafted feature engineering with context integration for multi-environment HAR.	Own Dataset	Automatic feature extraction and selection for performance improvements and computational time optimization.

Table 2 Popular HAR works with different incorporated metrics

Study	# Subjects	# Activities	# Attributes	M.L.	Sensors	Acc.%
Nandy et al. [31]	4	12	N.A.	FP, C	A,HR	EN = 94.61
Asim et al. [32]	60	15	18	N.A.	A	RF = 84
Kwon et al. [57]	10	7	21	W, C	A,G	RF= 98
Dehzangi & Sahu [58]	30	6	137	W, C	A,G	BE = 96
Patel & Shah [59]	30	7	561	W	A,G	LR = 94
Khokhlov et al. [60]	25	6	N.A.	AR, W, C	A,G	KNN = 97.9
Hong et al. [61]	15	24	9	W, WR, T	A	DT = 93.7
Choudhury & Soni [62]	24	6	16	FP	A,G	A-CL = 99.29

* A = Accelerometer, G = Gyroscope, HR = Heart rate Sensor, N.A.= Not Available with the cited reference, W = Waist, C = Chest, FP = Front Pockets, WR= Wrist, AR= Arm, T = Thighs, EN = Ensemble, BE = Bagged Estimation, A-CL= Adaptive CNN-LSTM, M.L. = Mounting Locations, Acc. Accuracy

4 Design and development of HAR system

4.1 Activity recognition series

4.1.1 Data acquisition modules

Any HAR system starts with a proper choice of sensor mix and mounting locations. One or many types of sensors are considered by the experimenter for data collection from a different set of users. It is very important to choose a good mix of users with various physique types and gender for getting generalized data. If one fails to ensure the above steps, the collected

Table 3 Publicly available datasets for sensor-based HAR system

Dataset	#Sub.	#Features	Sensors	#Class
HARSense [71]	12	16	A, G	6
USC-HAD [66]	12	N.A.	A,G	12
MotionSense [72]	24	12	A,G	6
MobiAct [73]	57	N.A	A,G	12
UCI HAR [65]	30	561	A,G	6
Opportunity [74]	4	242	Hybrid	9
mHealth [75]	10	23	A,G,M	12
RealWorld [76]	15	7	A	17
HASC [77]	5	4	A,G,M, GPS	6
WISDM [78]	29	46	A	6
PAMAP2 [79]	9	52	A,G,M,H	12
HAPT [80]	30	N.A.	A	12
w-HAR [81]	22	120	A,SS	7
HHAR [82]	30	40	A	7

*A = Accelerometer, G = Gyroscope, M = Magnetometer, GPS = Global Positioning System, H = Heart rate Sensor, SS = Stretch Sensor, Sub. = Subjects, N.A.= Not Available with the cited reference, Class = No. of Activities

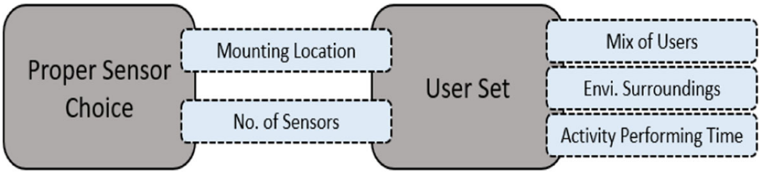


Fig. 4 Dependency between the different acquisition modules

data will of poor quality and the overall performance will be low. Ensuring proper mounting position is also very important as it helps in collecting true activity data.

Moreover, data must be collected in different environmental surroundings and for an adequate amount of time to get good patterns of various human activities. Considering all the factors, it is concluded that the data acquisition modules are highly dependent on sensor choice, mounting locations, number of sensors, a mix of users, environmental surroundings and activity performance time, and a consequence of one module can lead to overall poor data acquisition. The dependency between the different acquisition sets is shown in Fig. 4.

4.1.2 Data collection

The next step is to collect the data for making a HAR dataset. Data is collected through sensors in the local server machines, cloud servers or smartphones. Prominently the practitioners [30, 69, 83] store the data in the local server machines and smartphones for making the HAR database because of the accessible data compilation and negligible transmission delays. It also saves the data from harmful users and doesn't require additional costs for data storage. Few researchers [33, 49, 113], who implement real-time HAR, uses cloud-based data storage architecture for data collection. Cloud-based data collection is subdivided into two types. In the first technique, a user can opt. for free cloud-based services like - *ThingSpeak* for data collection where a substantial transmission delay and synchronization restrictions are present. Other uses the paid services like - Amazon Web Services - *AWS, Google Cloud services*, etc., where the transmission is significantly less, and there is no data syncing restriction.

4.1.3 Data pre-processing and visualization

Once the data is collected, it has to be analysed and pre-processed for further polishing and finding relationships between different features. It is one of the most crucial and time-consuming steps and needs to be done carefully to discover meaningful data. Data pre-processing helps in making data robust by removing noise, corrupted data, missing values, outliers and class imbalance, as shown in Fig. 5. The detailed description of all the data pre-processing steps is defined as follows-

1. *Handling Noise and Corrupted Data:* The first step of data pre-processing is removing noise and corrupted data. Noise and corrupted data degrade the data quality and make the training wicked. Corrupted data is taken care of by deletion of corrupted instances present in specific features or attributes. One can either remove the distinct attributes from the dataset if there are too many corrupted instances present or can remove only a few instances if there are less number of corrupted instances present. Noise is handled by using various techniques like - *Low-pass Filter, Kalman Filter, Moving Average Method*,

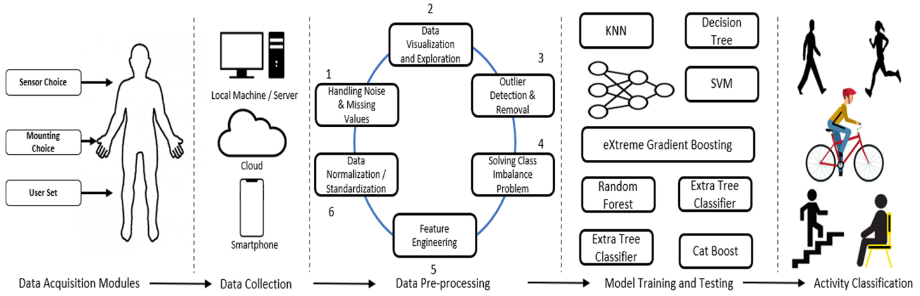


Fig. 5 Activity Recognition Series (Generic Framework) of HAR

etc. Noise needs to be removed very aggressively as it can influence other instances and attributes dependent on one another.

Ordinarily, the accelerometer and gyroscope are quite noisy devices, especially those which are incorporated into smartphones. The low-pass filter [114] is a popular method that allows the signals with a lower frequency than the threshold frequency and stops the higher-frequency signals. Kalman filter [115] is another widely used method that filters the noise by using predict and update framework. It uses the past data in the present data acquisition state to predict the new instance. Once the signal is generated, its value is refined using the past data, and the sensor values are kept in a specific range. The moving average method [116] is another form of low-pass filter that uses a statistical moving average strategy to calculate the average of ’n’ instances at a time and softens the sensor readings by returning a single value. It is slower than the traditional Low-pass and Kalman filter because of its heavy computation.

2. *Handling Missing Values:* Another factor that makes the dataset’s quality poor is the missing values. These are often present in the dataset because of some technical or human error while gathering the raw data from the users. It is handled by either *deletion* or *data imputation* [117]. In deletion, one removes the particular row where the missing values are present or removes the entire column if too many missing instances are present. But before deletion of the entire column, one has to ensure there is no attribute dependence and the particular column is not the prime attribute of the dataset.

   Another popular method for handling missing values is data imputation, where empty cells or instances are augmented with statistical values like- mean, median, standard deviation, percentiles, etc. The Pandas [118] data frame and NumPy [119] array packages are widely used for handling missing values with their inbuilt statistical functions. But the data imputation technique (mostly statistical approaches like - *mean*) is considered unsatisfactory for handling missing units because it violates the dependency between attributes and leads to standard errors.
3. *Data Visualization and Data Exploration:* Data visualization and exploration allow us to make critical decisions on raw data and make the datasets understandable. Feature dependency, transformation, scatteredness, segmentation and univariate ness are analyzed in the data exploration step [120]. Data visualization helps in model explanation by enabling exploratory data analysis, and natural inductive reasoning [121]. In layman’s terms, we could say that data visualization and exploration tend to interact with the dataset more efficiently by enabling faster decision-making and identifying errors and inaccuracies. The popular packages for data visualization and exploration include *matplotlib, seaborn, geoplotlib*.

4. *Outlier Detection and Removal:* Outliers are the data points that have the same behaviour as the expected data points but are significantly far from the usual data dispersion. They affect the model training phase and lead towards poor model generalization. If not removed, there will be a degradation of the overall model performance. The researchers in [122] incorporated multiple outlier handling mechanisms for tackling HAR outliers which can be broadly categorised into two types - a. *Density-based outlier detection*, b. *Distance-based outlier detection*. Density-based outlier detection can detect the *local outliers*, and distance-based methods are competent in detecting the *global outliers*. HAR dataset tends to have both types of outliers because of the involvement of dynamic movement of human activities and numerous sensors.
5. *Solving Class Imbalance Problem:* It often occurs that the HAR dataset has a class imbalance problem. HAR system encounters this problem because of the difference in time for performing numerous activities. This problem leads to biases about a specific class label or human activity. It can be slightly biased (low-class imbalance) or severely biased (very high-class imbalance), as shown in Fig. 6. The machine learning and deep learning models assume a uniform distribution of classes in the dataset. If one passes the imbalanced dataset, the model will be more biased towards the majority class and will perform poorly while validating the model [96, 97].

   Schemes like - model over-sampling and model under-sampling are widely used for handling the imbalanced dataset. Over-sampling creates the random duplicates of the minority class and uplifts them by matching the majority classes. On the other hand, model under-sampling removes the random instances of majority classes and matches them up with the minority class. Nowadays, another method named - synthetic sample generation is widely used where the new samples are created with the help of statistical artificial intelligence techniques like - the distance between data points and its neighbours' count. *Imbalanced-learn* with *scikit learn* are the two popular libraries that are used for handling the class imbalance problem.
6. *Feature Extraction and Selection:* HAR dataset consists of *n* number of features, and some of them can be relevant, relatively relevant and irrelevant as well. One can easily find out the effect of each feature or attribute by solely running models with a particular feature. The features that yield good accuracy are primarily relevant in nature, and the

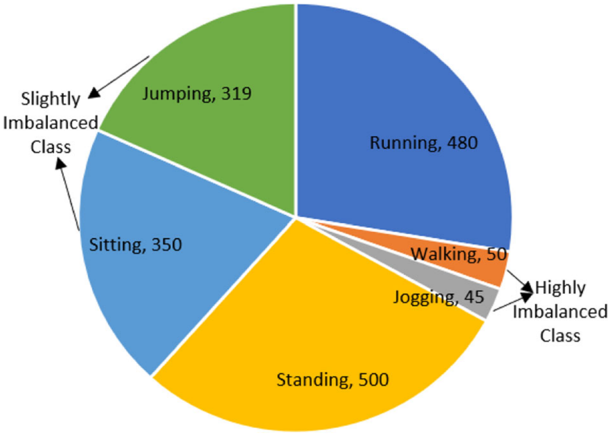


Fig. 6 Class Imbalance Problem

irrelevant features will contribute less to the performance. Generally, we use *feature construction, feature extraction and feature selection* [58, 123–125] for reducing the overall dataset dimensionality. Feature construction is the practice of creating additional features from the already available data or features.
Feature extraction is the process of extracting new information from the raw dataset. It creates a new form of information from the stored data and minimizes the duplicated data. HAR datasets are often very big, with huge numbers of activity instances and raw sensor values of multiple sensors like - tri-axial accelerometers, gyroscopes and many more. It has always been a big challenge for researchers to handle big datasets and study them. Most of them [63, 124, 126], uses handcrafted feature extraction techniques for generating the new information from the raw data as referred in the Table 4.
Often, a dataset with too many features is very hard to handle, and extracting features from them is considerably complex. To address that kind of dataset, one must use a feature selection mechanism like - *mutual information* or *information gain* to select the most relevant features from the set of all available features. It reduces the overall time complexity of the model and helps in overcoming model overfitting as well.

7. *Data Normalization and Standardization:* HAR dataset has numerous data instances and has a mix of sensor values. Because of the bulk of data present in the HAR dataset, it is always advisable to bring all the feature data on a standard scale. Numerous methods are

Table 4 HAR Handcrafted Features for Inertial Measurement Unit (IMU) [63, 126]

Domain	Features
Frequency	Skewness
	Weighted Average
	Energy
	Maximal Frequency
	Angle between two vectors
	Percentile
Time	Mean
	Median
	Standard Deviation
	Maximal
	Minimal
	Entropy
	Correlation coefficient
	Auto regression Coefficient
	Interquartile range
	Average sum of squares
	Kurtosis
	Range
	Variance
	DC Components
	Number of mean crossing
	Wavelet Energies
	Percentile

Table 5 Popular Data Normalization and Standardization

Method	Mathematical Formulation
Min-Max Normalization	$\frac{x_i - Min(x)}{Max(x) - Min(x)}$
Mean Normalization	$\frac{x_i - Average(x)}{Max(x) - Min(x)}$
Z-Score Standardization (Standard Scalar)	$\frac{x_i - Mean(x)}{Std.Dev.(\sigma)}$

used for data normalization and standardization by the researchers, and a few popular ones are listed in Table 5.

Min-Max normalization transforms the feature instances by replacing the maximum value to *one*, the minimum value to *zero* and the remaining values between *zero* and *one*. Mean normalization scales the features by removing the average value from every instance. The z-score standardization is another form of mean normalization where the data instances are normalized using the standard deviation. Z-score standardization is widely known as *Standard Scalar* standardization. It is important to note that the normalization process is specific to the type of learning algorithm. The distance-based algorithms like - *KNN, SVM* have a terrific impact on data normalization, but the gradient descent-based algorithms like - *Artificial Neural Network, Logistic Regression, Linear Regression* suffer from it. On the other hand, tree-based learning algorithms like - *Decision Tree, Random Forest* have an almost negligible effect on data normalization and standardization.

4.1.4 Model building and evaluation

Once the data is pre-processed, the model building and evaluation are done, where one or multiple learning algorithms are trained and tested. At first, the entire dataset is divided into two parts, i.e., *training set* and *testing set*. The training set has a higher number of data instances as it is required to train the model for finding the rules and patterns from the dataset. For every ADL present in the dataset, the rules are generated and stored in the trained model. The testing set has a lower number of data instances since it is applied for model evaluation. Ideally, the dataset's input instances are split randomly for unbiased model classification and performance measurement. We analysed that the split ratio of $70 : 30$, $80 : 20$ and $90 : 10$ is the most optimal split for most of the dataset.

After the training and testing split, classification models are selected based on the requirements. Model training is an iterative process where new rules and patterns are determined from the training set w.r.t the associated class labels, and the best combination of weights is allotted over the classification range. The block-level view diagram of the model training phase for tree-based algorithms like - *Decision Tree* is shown in Fig. 7. Once the model is trained, it can be quickly evaluated using a test set for getting the performance score. One of the significant advantages of model training is that it can be deployed in multiple computing environments and can be used accordingly. The test set is supplied to the trained model and iteratively tested with different activity classes. Training and testing sets are often not enough for optimal performance evaluation (mostly with deep learning models); another set named - the validation set, which consists of a small number of data instances, is used to test and tune the model. *Hyper-parameter Tuning* is the process of tweaking the model in terms of various model-specific parameters like - weights (w), neighbour values (k), learning rate (l_r), threshold (α), etc. for getting the best performance from the models.

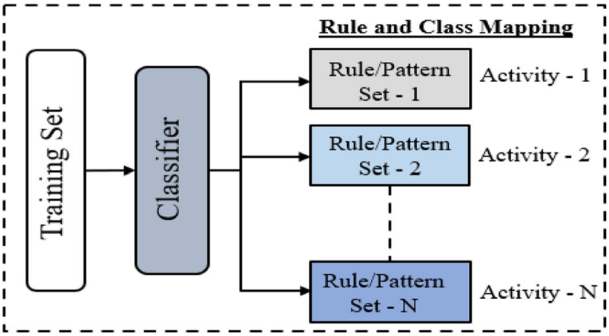


Fig. 7 Block view of the model training phase for tree-based algorithms

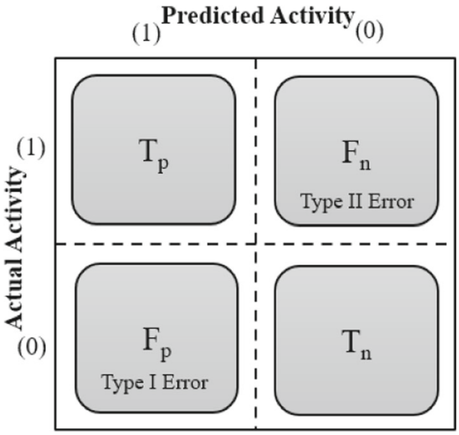
Model evaluation is done with the help of various performance metrics and is very important for different domain applications. HAR systems also need different performance metrics for evaluating the best classifier, architecture, dataset, sensor types, mounting locations, etc. Evaluation metrics that are extensively popular for measuring the classification performance with the help of confusion matrix, as shown in Fig. 8 for the HAR system, are as follows-

1. **Precision (P)** signifies the rate of correctly predicted positive classes when both the actual and notional positive classes are given to the model. Mathematically, it is defined as the number of true positives (T_p) over the number of true positives plus the number of false positives (F_p).

$$P = \frac{T_p}{T_p + F_p} \tag{1}$$

2. **Recall (R)** is the actual positives of the classification model that is defined as the total number of true positives (T_p) over the total number of true positives (T_p) and false negatives (F_n). This measure is also commonly known as *sensitivity or True Positive Rate* (TPR) and is incorporated when we need to see the performance of the actual class labels.

Fig. 8 Confusion Matrix for the classification model



$$R = \frac{T_p}{T_p + F_n} \tag{2}$$

3. **Specificity** calculates the performance of the classification model when we provide the negative or wrong class to the classifier, and the classifier manages to predict it as the negative class successfully or not. Specificity is also known as False Positive Rate (FPR). Mathematically, it is defined as the true negatives (T_n) over the total of true negatives (T_n) and false positives (F_p).

$$Specificity = \frac{T_n}{T_n + F_p} \tag{3}$$

4. **F1-Score (F1)** calculates the harmonic mean between the ***Precision*** (P) ***and Recall*** (R) and is one of the widely used performance metrics.

$$F1 - Score = 2 * \left(\frac{P * R}{P + R}\right) \tag{4}$$

5. **Accuracy (A)** is the most used performance metric in the HAR system and is defined as the ratio of correctly predicted class (*total true instances*) of the total number of observations. The accuracy (A) is defined as follows.

$$A = \frac{T_p + T_n}{T_p + T_n + F_p + F_n} \tag{5}$$

6. **Computational Time** (C_T) is defined as the total amount of time required for model training (T_r) and testing (T_s). Mathematically, it is defined as -

$$C_T = T_r + T_s \tag{6}$$

   where T_p, T_n, F_p and F_n is defined in the Table 6.

Finally, after completing the model evaluation, researchers often perform cross-validation of their HAR system to validate their incorporated parameters, and effective usage of the overall dataset [127]. Cross-validation helps in understanding overall system performance and identifies whether the proposed system is reliable or not. The popular methods which competent investigator [128, 129] inherits are as follows-

1. *K-Fold Cross-Validation:* K-Fold cross-validation is one of the most common validation methods used by researchers [130, 131]. It effectively utilises all portions of the dataset and gives possible room for hyperparameter tuning based on the different fold performance.
   In this method, the dataset is divided into k different folds, and the train and test sets are evaluated in multiple iterations. In each iteration, $(k - 1)$ folds are used for training, and

Table 6 Description of different measure used in the performance metrics

Measure	Description
T_p	When the positive class is classified as positive.
T_n	When the negative class is classified as negative.
F_p	When the negative class is classified as positive.
F_n	When the positive class is classified as negative.

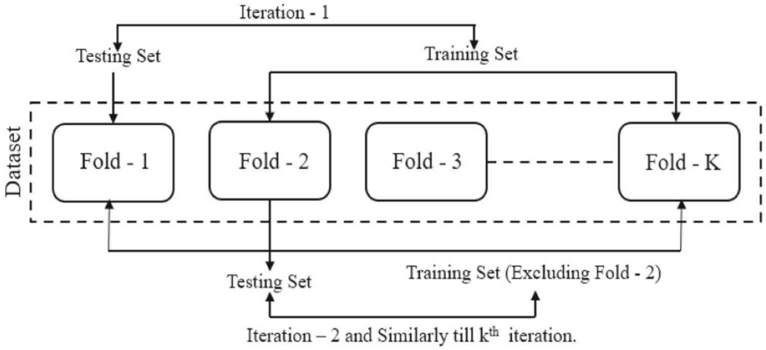
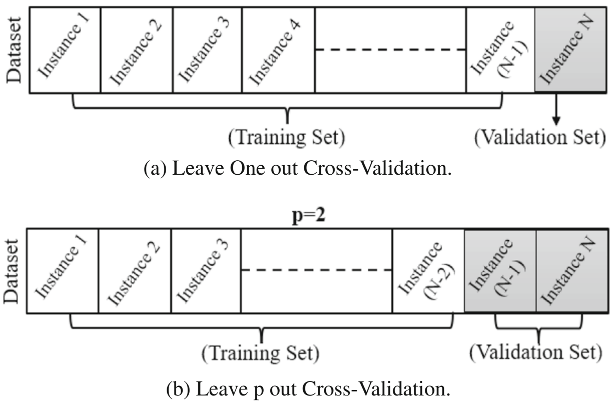


Fig. 9 Architecture of K-Fold Cross-Validation

the remaining 1 fold is used to test the model, as shown in Fig. 9. Finally, the average of all the k folds is considered as the overall performance measure.

2. *Leave One Out:* Leave one out is a form of *K-Fold Cross-Validation* where the dataset is divided into N number of parts, where N is equal to the number of data points or instances in the dataset [132, 133]. Therefore, one instance of the dataset will be used to validate the model and the remaining $(N - 1)$ instances will be used to train the classification model.

   Another form of leave one out cross-validation is *leave p out cross-validation* where the user defines the size of validating set in the form of p parameter. Accordingly, the data instances are chosen from the dataset for the validation set. The remaining $(N - p)$ instances are used as the training set. It can be easily understood with the help of Fig. 10. The above form of cross-validation is also iterative in nature, and like k-fold cross-validation, it is computationally expensive because almost all the instances need to be tested, which will require exponential time in case the dataset is huge.
3. *Monte Carlo Cross-Validation:* In above practices, user needs to explicitly pass a splitting parameter through which a dataset is partitioned into multiple subsets. Monte Carlo cross-validation [134] randomly picks a portion of data from the incorporated dataset and use the particular portion for model training. The remaining part is used for model validation in



(a) Leave One out Cross-Validation.

(b) Leave p out Cross-Validation.

Fig. 10 Architecture of Leave one and p out Cross-Validation

the end. This process is repeated multiple times iteratively by selecting different portions of the dataset without replacement as the training set and remaining data for validation. Finally, the average is calculated to measure the overall performance factor. But here, a few data instances could be left in the training and testing set as it is completely random in nature.

5 Experimental result and discussion

This section shows and describes a detailed process of experimenting with an efficient HAR system using in-build smartphone sensors. The results and their performance metrics are discussed in detail with the help of generated dataset and incorporated classifiers.

5.1 Experimental setup

For our experimental setup, we used two different devices for data collection from 24 different users. Three android smartphones named - “Oneplus 9 Pro”, “Poco X2” and “SAMSUNG Galaxy A30-S” are used for ADLs data collection by mounting it on the waist and front pocket. The mounting of the sensor module and various activities performed by different subjects are shown in Fig. 11. A publicly available android application named *SensorRecord* is incorporated to read the accelerometer and gyroscope readings from the user. The same data is stored in the form of a CSV (Comma-Separated Values) file in the smartphone itself for convenience. The features that are present in our dataset are directly collected from the android application and inbuilt smartphone sensors, and are constructed up of *Rotational*

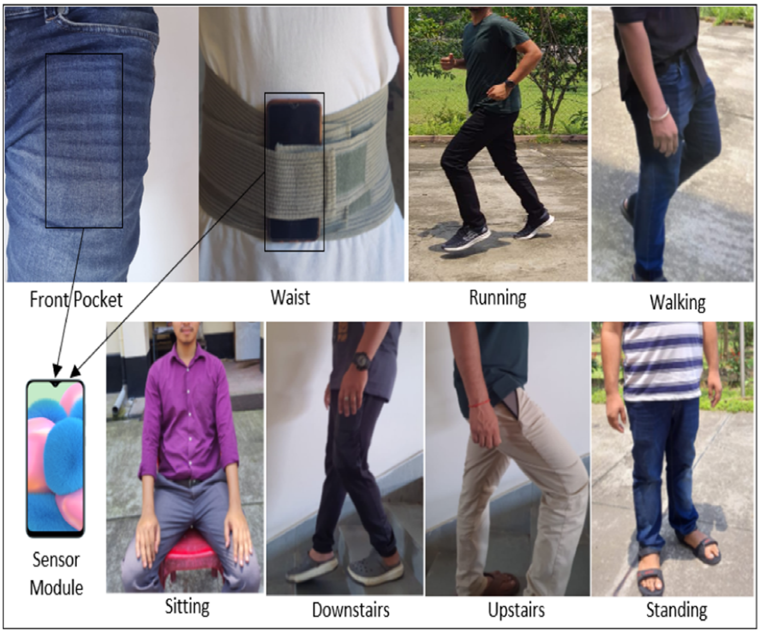


Fig. 11 Different ADLs performed by users and sensor module- mounting location

Rate (3D), *Gravity* (3D), *Linear Acceleration* (3D) and *Acceleration due to Gravity* (3D). Among the feature set, the gravity, linear acceleration and acceleration due to gravity are collected through the accelerometer sensor, and only the rotational rate is collected through the gyroscope sensor.

The data is collected at the rate of 100 Hz, and each individual activity of all the subjects is done for a period of 60 to 120 seconds each. The various activities include – *Standing, Walking, Running, Sitting, going Downstairs* and *going Upstairs*. The age of the subjects is in-between 25 ± 35; height is in-between 155 ± 185 cms, and weight is in-between 66 ± 85 kgs. The formulation of the dataset, along with its feature set with respect to its sensors, is shown in (10). Fs^1 and Fs^2 features are collected from the accelerometer sensor as defined in (7) and (8), and Fs^3 is collected from gyroscope sensor module as defined in (9), where, (AG) is *Acceleration due to Gravity*, (LA) is *Linear Acceleration*, (GV) is *Gravity* and (RR) is *Rotational Rate*.

$$Fs^1_{NX6} = \begin{Bmatrix} AG_{x_0} & AG_{y_0} & AG_{z_0} & LA_{x_0} & LA_{y_0} & LA_{z_0} \\ AG_{x_1} & AG_{y_1} & AG_{z_1} & LA_{x_1} & LA_{y_1} & LA_{z_1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \\ AG_{x_n} & AG_{y_n} & AG_{z_n} & LA_{x_n} & LA_{y_n} & LA_{z_n} \end{Bmatrix} \tag{7}$$

$$Fs^2_{NX3} = \begin{Bmatrix} GV_{x_0} & GV_{y_0} & GV_{z_0} \\ GV_{x_1} & GV_{y_1} & GV_{z_1} \\ \vdots & \vdots & \vdots \\ GV_{x_n} & GV_{y_n} & GV_{z_n} \end{Bmatrix} \tag{8}$$

$$Fs^3_{NX3} = \begin{Bmatrix} RR_{x_0} & RR_{y_0} & RR_{z_0} \\ RR_{x_1} & RR_{y_1} & RR_{z_1} \\ \vdots & \vdots & \vdots \\ RR_{x_n} & RR_{y_n} & RR_{z_n} \end{Bmatrix} \tag{9}$$

$$Dataset_{NX13} = \left\{Fs^1 \ Fs^2 \ Fs^3, Classes\right\} \tag{10}$$

The formulated dataset adds diversity and accounts for possible sensor performance, sensitivity, and quality variations across various devices as it considers multiple sensor modules for data acquisition. Also, we collected different ADLs data in uncontrolled environments, which allows the users to perform multiple activities freely, making the data instances real and minimally augmented, unlike the dataset formulated in a controlled scenario. Further, with data from 24 users, the dataset offers a relatively large and diverse user cohort. This diversity can help capture variations in human activities across individuals, leading to more robust and generalized models.

Along with our dataset, we incorporated a public dataset named - mHealth [75] for benchmark comparison and analytical discussion on different aspects. The mHealth dataset consists of sensor data from smartphone that were performed by 10 participants while they engaged in 12 physical pursuits. The dataset contains six different sensor readings, including ECG (electrocardiogram) data, accelerometer, gyroscope, and magnetometer readings. The dataset's objective is to make it easier to conduct mobile device study on human movement recognition. The dataset has been extensively used in machine learning research, especially in the creation and assessment of algorithms for fall detection and activity recognition.

After data collection and dataset building, we performed data pre-processing on the generated dataset and mHealth to remove all the inconsistencies and get insights into the acquired data. As an initial point, we checked for corrupted and missing values and found out that there were zero instances of them. Then we checked for redundant (duplicate) instances and found out that 37.46% of data were redundant. We removed them using the *drop* method from *pandas* [118] data analysis tool. For the rest of the experiment, we have used the remaining 62.54% of data instances as our primary focus. Removing redundant data (if it is in huge quantity) is imperative for saving computational time and model overfitting, which will lead to poor performance in real-time activity classification. Further, for mHealth there were no missing, duplicate and corrupted present in the dataset.

In the next stage, we checked for outliers in our dataset using data visualization tools and found out that 99% of the data are good and the remaining 1% are outliers. We did not remove the small fraction of them to make the model a bit more complex for both model training and evaluation. Then in the next step, we checked for class imbalance. We marked that our dataset has a significant class imbalance problem upon envisioning the different activity labels. As discussed above in Section 4.1.3, the class imbalance problem must be rectified, or otherwise, the majority class will dominate the minority classes, and it will affect the overall classification performance. To solve our datasets' imbalance trouble, we incorporated the *Random Over Sampler - ROS* [135] method that randomly picks considerable examples of a minority class with replacements and puts them into the training set. The result of before and after class imbalance can be seen in Figs. 12 and 13, respectively.The need of ROS for balancing the mHealth dataset was not needed as mHealth dataset was already balanced with almost 25243 ± 27648 instances except one activity with 9267 instances and the class distributions are shown in Fig. 14.

Once the class imbalance problem is solved, we constructed the features for passing the training and testing set to the classifiers. Here, the activity labels are encoded to an integer using *Label Encoder* as many of the classifiers only accept the numeric activity class labels for the classification task. The result of encoding is *0 - Running, 1 - Sitting, 2 - Standing, 3 - Walking, 4 - downstairs, 5 - upstairs*. The same has been done for the mHealth dataset using *Label Encoder*. Feature extraction and selection are not required for both datasets as they have enough data instances and features for model training. Then, we checked the performance of our incorporated models with and without data standardization - *Standard Scalar* and normalization - *Min Max Normalization* on both dataset. Data standardization and normalization helps in delivering the dataset patterns an equal weight for model training.

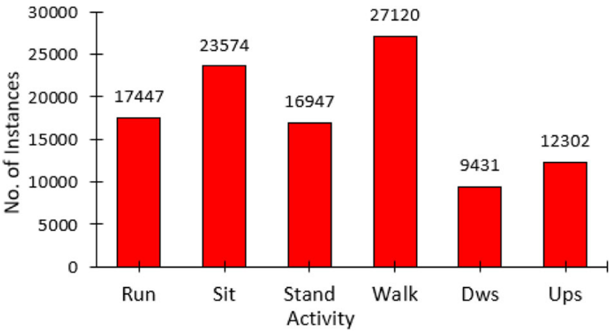


Fig. 12 Imbalanced data instances of our generated dataset

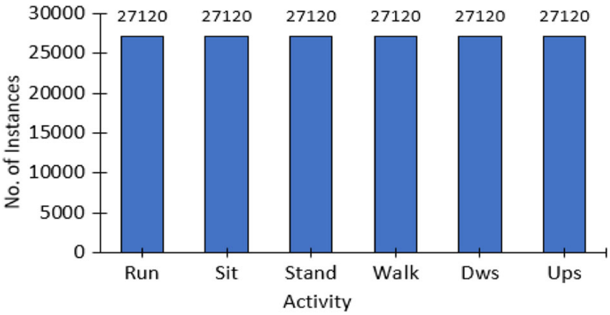


Fig. 13 Balanced data instances after ROS on own generated dataset

But because of this, a few models like - gradient-descent based classifiers suffer, which is further explained in the discussion section.

5.2 Result analysis and discussion

Various classifiers with respect to its learning methods are incorporated for recognising human activities. Decision Tree (DT), k-Nearest neighbour (KNN) and Naive Bayes (NB) are considered under shallow learning algorithms, Random Forest (RF), eXtreme Gradient Boosting (xGB), Cat Boost (CB) and Lightweight Gradient Boosting Model (LGM) are considered under ensemble learning method, and Long Short Term Memory (LSTM) and Hybrid CNN-LSTM models are incorporated under deep learning methods.

To train and test the incorporated classifiers, we have used a Windows-based workstation built up of 32 GB RAM (31.6 GB usable), Intel Xeon W-2133 @ 3.6GHz Processor, 2TB Hard Disc Drive and NVIDIA Quadro P2000 GPU. All the models are tested on Python (Version 3.9) using Anaconda Jupyter Notebook. After the complete data pre-processing, we divided the dataset into two sets, i.e. training set and testing set, with a ratio of 80 : 20, respectively. The generalized view of the training and testing set is summarized in (11) and (12).

$$Train_{NX13}^{Set} = \left\{(Fs_{1-n}^{1}\ Fs_{1-n}^{2}\ Fs_{1-n}^{3}, Activity)\right\} \tag{11}$$

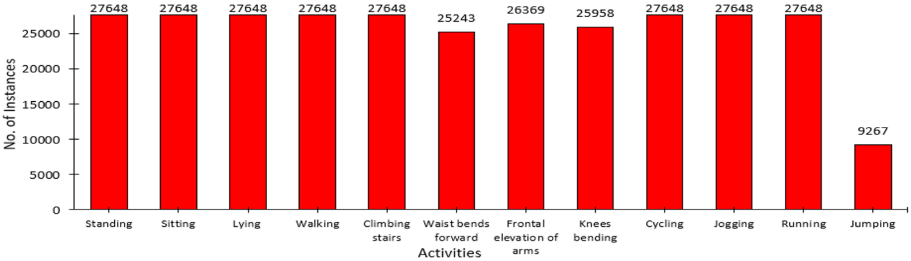


Fig. 14 mHealth dataset class label distributions

$$Test_{NX12}^{Set} = \left\{((Fs_{k-m}^{1}\ Fs_{k-m}^{2}\ Fs_{k-m}^{3})\right\} \tag{12}$$

5.2.1 Analysis of deep learning algorithms

We choose two deep learning models, namely- LSTM and CNN-LSTM, for evaluating our dataset. LSTM and CNN-LSTM are considered as they are very efficient with automatic feature analysis on time series data and have low chances of training and validating loss. The incorporated hyperparameters for both models are summarized in Table 7. On testing the models, we got an average accuracy of 96.24% and 96.53% with LSTM and CNN-LSTM models on our dataset, respectively. The detailed comparison between the two model accuracies after 10- fold cross validation is shown in Fig. 15. The P, R is also promising as compared to the shallow and ensemble learning models, as the capability of model generalization is better with deep learning algorithms.

LSTM shows the volatility in model training as the loss on validation data is not consistent, and because of this, its overall training accuracy is also changing with successive iterations. It is happening because it lacks efficient hierarchical feature extraction on time series data. It successfully handles the long-term dependencies but fails to analyse the prime features. The model is overfitting after a certain amount of training and validation, and the same is shown in Fig. 16. Unlike LSTM, the hybrid CNN-LSTM model manages to extract features from the time series data using its Conv1D method and its again memorized with the help of LSTM memory mechanism. Long-term dependencies and high-level features are efficiently extracted here, producing very low training loss. The training-validation accuracy and loss of

Table 7 Summary of hyper-parameters used for deep learning model

Model	Metric	Used parameter
CNN-LSTM	Convolution	Conv1D
	Filter	64
	Kernal_size	3
	drop_out	0.5
	MaxPooling	MaxPooling1D(2)
	CNN_activation	relu
	LSTM_activation	relu
	output_activation	softmax
	neuron_dense	200
	batch_size	512
	epoch	150
LSTM	model	Sequential
	drop_out	0.5
	LSTM_activation	relu
	output_activation	softmax
	LSTM_neurons	200
	neuron_dense	200
	epoch	150
	batch_size	512

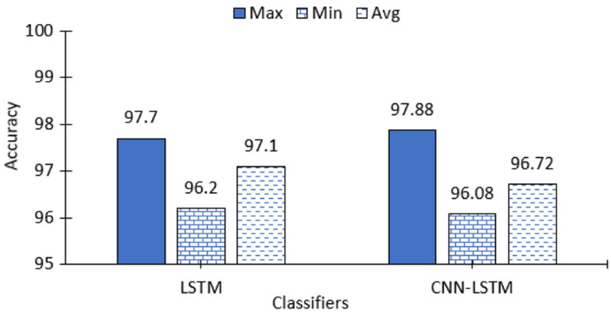


Fig. 15 10-Fold accuracies of deep learning models (own dataset)

this model can be seen in Fig. 17. As we have comparatively fewer samples after redundant data removal, the above model can yield better results when a huge dataset is provided to them. Due to the lack of data, deep learning models are overfitting and hampering the overall accuracy, as shown in Fig. 18.

Both model also achieves optimal accuracies on the mHealth dataset as well, and the 10-Fold accuracies are shown Fig. 19. CNN-LSTM achieves better results than LSTM due to the advantage of superior feature extraction from Conv1D layers. Also, huge training instances help the models to generalize optimally and exert higher training and testing accuracies as compared to our dataset. The training and testing accuracies of LSTM and CNN-LSTM are compared in Fig. 20.

Due to the cardinal benefit of balanced and vast data instances, both models experience minimal training and validation data losses. Both models were trained for 150 iterations, as referred to in the Table 7 and the training accuracies and loss achieved by the models were optimal, as shown in Figs. 21 and 22, respectively. Also, as both models are not overfitting in the successive iteration after 60 ± 80, we can stop the training process at 80 epochs to save computational cost.

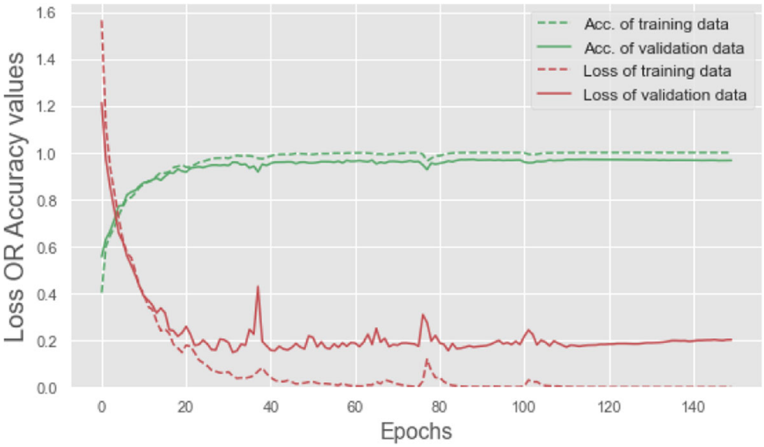


Fig. 16 Model loss and accuracy of LSTM classifier (own dataset)

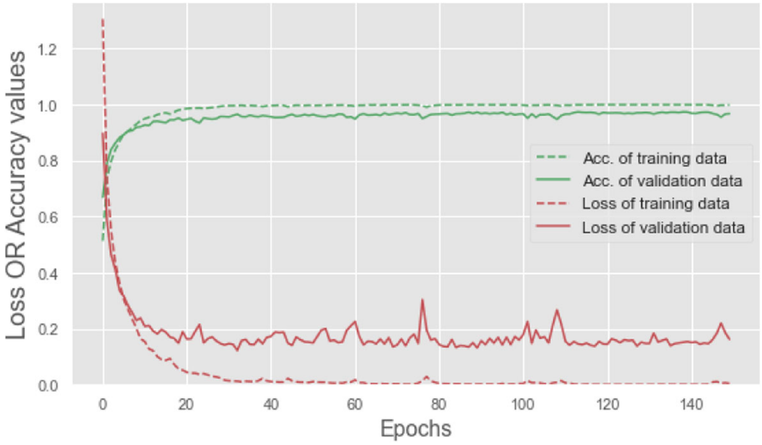


Fig. 17 Model loss and accuracy of CNN-LSTM classifier(own dataset)

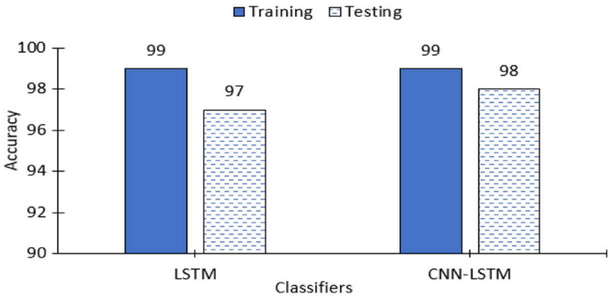


Fig. 18 Training and Testing accuracies of LSTM and CNN-LSTM models showing model overfitting (own dataset)

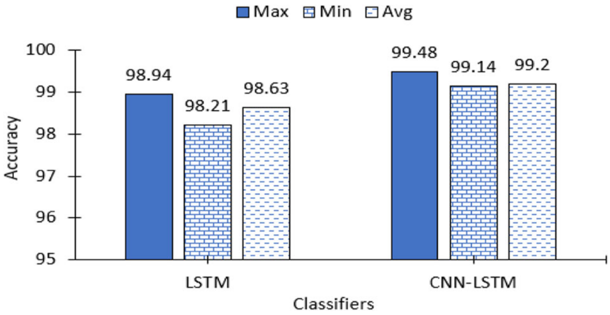


Fig. 19 10-Fold accuracies of deep learning models (mHealth dataset)

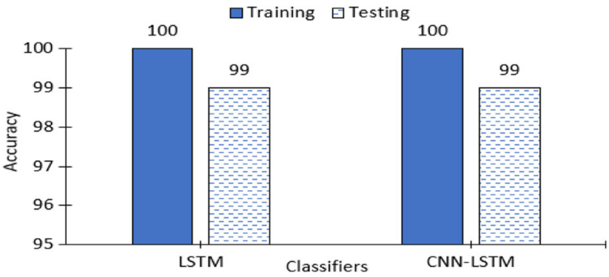


Fig. 20 Training and Testing accuracies of LSTM and CNN-LSTM models showing model overfitting (mHealth dataset)

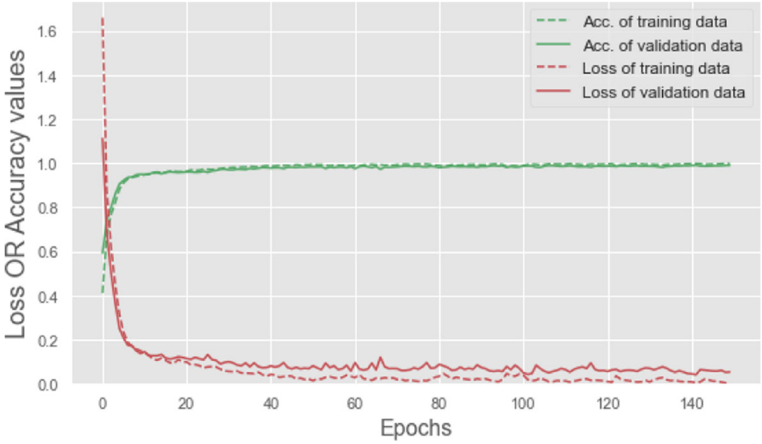


Fig. 21 Model loss and accuracy of LSTM classifier (mHealth dataset)

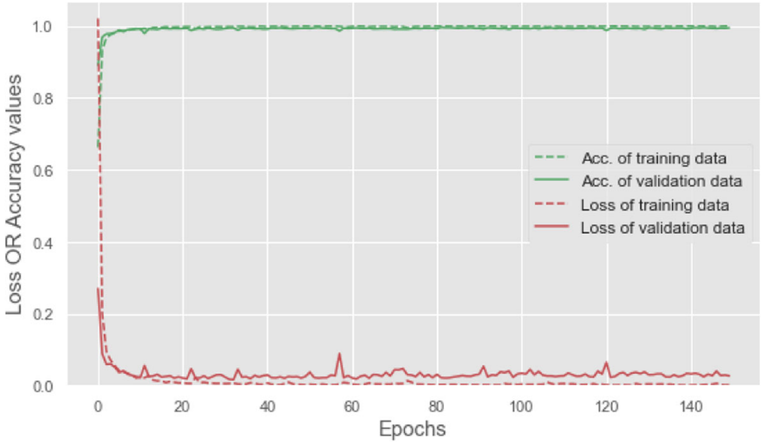


Fig. 22 Model loss and accuracy of CNN-LSTM classifier (mHealth dataset)

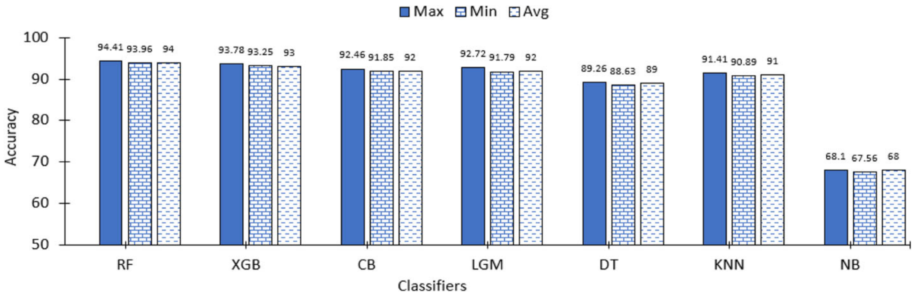


Fig. 23 Accuracy comparison of different shallow and ensemble models of own dataset

5.2.2 Analysis of shallow and ensemble learning algorithms

Evaluating the incorporated shallow learning models on the processed datasets, we managed to achieve average accuracies of 89% and 91% with DT and KNN, respectively on our own dataset. At the same time, both DT and KNN managed to achieve an average accuracy of 97% and 91%, respectively, on the mHealth dataset. NB suffers from performance loss as the classification criteria were conditional probability which is optimal for binary classification but significantly not efficient for multi-class classification. DT classifier is tested with *Gina_Index* criteria with the minimum sample split of 2. On the other hand, KNN is tested with 5 neighbours and Euclidean distance for distance calculation criteria. The Precision, Recall and F1-Score values are also remarkably high and consistent for all the 10 - fold cross-validation. The average, maximum and minimum accuracies of both datasets achieved after 10-fold cross-validation are compared in Figs. 23 and 24. Shallow learning models are not able to match the performance of deep and ensemble learning models as it lacks automatic feature engineering, which is automatically done by the hidden layers of deep learning algorithms, and they also lacks the support of multiple base learners for effective model generalization. The need for experts to perform feature extraction and selection is a primary drawback here. But in contrast, because of its simple mathematical operations and model training, the computational time is much more efficient than the ensemble and deep models. The detailed performance comparison with other learning models are described in Table 8.

On the other hand, ensemble learning models performed much more efficiently than shallow learning models. The accuracies of all four classifiers for both datasets are shown in Figs. 23 and 24, respectively. The involvement of multiple base learners and random dataset

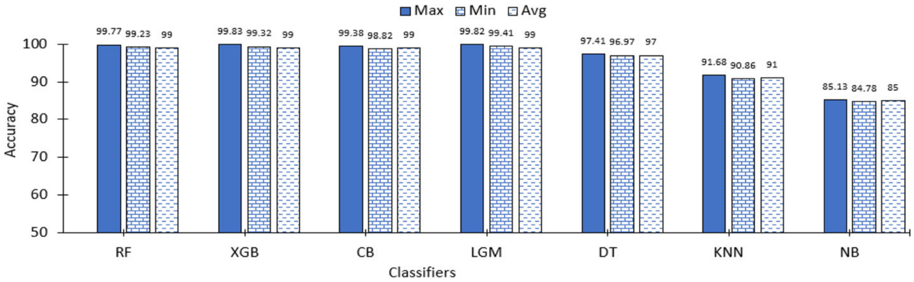


Fig. 24 Accuracy comparison of different shallow and ensemble models of our mHealth dataset

Table 8 Detailed comparison of different learning models with computational times of our own dataset

Classifier	Hold-Out	10-Fold Acc.	Avg.	Precision	Recall	F1-Score	Avg. C_T (Sec.)
RF	94.41	94	M. Avg.	0.94	0.94	0.94	25.1
			W. Avg.	0.94	0.94	0.94	
XGB	93.78	93	M. Avg.	0.93	0.93	0.93	295.8
			W. Avg.	0.93	0.93	0.93	
CB	92.46	92	M. Avg.	0.92	0.92	0.92	3.6
			W. Avg.	0.92	0.92	0.92	
LGM	92.72	92	M. Avg.	0.93	0.93	0.93	29
			W. Avg.	0.93	0.93	0.93	
DT	89.26	89	M. Avg.	0.9	0.9	0.9	1.7
			W. Avg.	0.9	0.9	0.9	
KNN	91.41	91	M. Avg.	0.91	0.91	0.91	3.53
			W. Avg.	0.91	0.91	0.91	
NB	68.1	68	M. Avg.	0.68	0.68	0.68	3.53
			W. Avg.	0.68	0.68	0.68	
LSTM	97.7	97	M. Avg.	0.99	0.99	0.99	892
			W. Avg.	0.99	0.99	0.99	
CNN-LSTM	97.88	97	M. Avg.	0.99	0.99	0.99	627
			W. Avg.	0.99	0.99	0.99	

* Acc.=Accuracy, Avg.=Average, M. Avg=Micro Average, W. Avg=Weighted Average

splits yields better results. Also, the support of multi-feature handling and transferring to the base learners produced better output in all the trained models, and the result aggregation got a better performance result. It takes more time to train and test the model as compared to shallow learning models, but it is much more computationally efficient than deep learning models. The tendency of handling bias or variance on all the base learners multiple times helps overcome model overfitting and under-fitting, resulting in better accuracy performance. XGB yielded highest accuracy of 99.43% with the training parameters of 100 base estimators and gini index splitting criteria. Along with the XGB classifier, RF, CB and LGM got an average accuracy of 99% each on the mHealth dataset. Our dataset suffers from performance loss as the data collected has fewer training samples than mHealth, and the models fail to train the parameters optimally. The detailed comparison of the incorporated ensemble approaches are described in Table 8. In contrast to the deep and shallow learning algorithms, the RF ensemble model performs better in computational time and yields comparable accuracy to the deep learning models for the mHealth dataset. The detailed performance comparison of all the incorporated classifiers of our own and mHealth datasets are summarized in Tables 8 and 9, respectively.

5.3 Effect of model evaluation schemes

We incorporated holdout and k-fold cross-validation (k= 10) method for performance comparison and checking the performance of different model evaluation schemes. The Holdout method performs better in terms of accuracy than the 10-fold cross-validation as it uses a

Table 9 Detailed comparison of different learning models with computational times of mHealth dataset

Classifier	Hold-Out	10-Fold Acc.	Avg.	Precision	Recall	F1-Score	Avg. C_T (Sec.)
RF	99.37	99	M. Avg.	1	1	1	105.1
			W. Avg.	1	1	1	
XGB	99.43	99	M. Avg.	1	1	1	1998.23
			W. Avg.	1	1	1	
CB	99.38	99	M. Avg.	0.99	0.99	0.99	169.85
			W. Avg.	0.99	0.99	0.99	
LGM	99.82	99	M. Avg.	1	1	0.93	11.02
			W. Avg.	1	1	1	
DT	97.41	97	M. Avg.	0.97	0.97	0.97	10.4
			W. Avg.	0.97	0.97	0.97	
KNN	91.68	91	M. Avg.	0.92	0.92	0.92	651.28
			W. Avg.	0.92	0.92	0.92	
NB	85.13	85	M. Avg.	0.85	0.85	0.68	0.32
			W. Avg.	0.85	0.85	0.85	
LSTM	98.94	98	M. Avg.	0.99	0.99	0.99	1256
			W. Avg.	0.99	0.99	0.99	
CNN-LSTM	99.48	99	M. Avg.	1	1	1	856
			W. Avg.	1	1	1	

* Acc.=Accuracy, Avg.=Average, M. Avg=Micro Average, W. Avg=Weighted Average

more significant proportion of the dataset for training and a smaller proportion for model validation, which results in an accurate evaluation of the testing set. However, this also means that the holdout method may suffer from higher variance and may not generalize as well to new data. On the other hand, k-fold cross-validation provides better estimate of model performance by averaging the results of multiple evaluations (in our case, 10-folds). This is particularly useful when the dataset is large or when the model is complex, as it ensures that all data is used for both training and validation. However, k-fold cross-validation requires more computational resources and is much slower than the holdout scheme. The comparison of both evaluation scheme with respect to accuracy is summarized in the Tables 8 and 9 for own and mHealth dataset, respectively.

5.4 Effect of data standardization and normalization on different classifiers

As discussed in section 4.1.3, we reviewed that the gradient-descent-based models suffer from data normalization or standardization. We applied Min-Max normalization and Standard-Scalar data standardization to see this degrading performance phenomenon and compared the results of all the incorporated classifiers. Upon testing the incorporated classifiers, distance-based classifiers managed to achieve better results with the performance increase of $(91 \pm 94)\%$ with our own dataset and $(91 \pm 99)\%$ with the mHealth dataset. Also, the computational times were optimised because of common scale data. Other shallow and ensemble learning models did not show any changes to the accuracies, but the training times were also optimized for them.

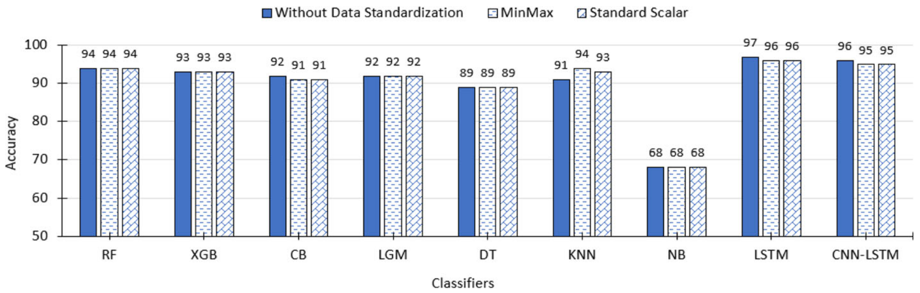


Fig. 25 Accuracy comparison of different classifiers with and without data standardization and normalization on own dataset

On the other hand, both gradient-descent-based models - LSTM and CNN-LSTM suffered from accuracy loss with an overall accuracy loss of 1% on both datasets. It is happening as the gradient descent algorithm is sensitive towards the scale of input features and the relationships between the input features and output variables are not preserved after data normalization. But, one positive effect was the efficient computational times of both models as the computations become easy for non-linear activation functions. The achieved accuracies by different classifiers with and without data standardization and normalization on our dataset and mHealth are shown in Figs. 25 and 26, respectively.

5.5 Effect of different sensors and learning models

Finally, we analysed the performance of different features acquired by the incorporated sensors for data collection. First, we checked the individual feature accuracies and achieved good performance with Accelerometer (A) features on our own dataset. On the contrary, Gyroscope-based (G) feature - Rotational Rate, achieved poor performance result as it fails in optimally categorise the activity patterns. But, combining both A and G features, the performance improves and achieves better accuracies than the individual feature set. The detailed comparison of all the incorporated classifiers on our dataset is described in the Table 10.

On the other hand, G features yield better performance for the mHealth dataset, and A features suffer from model generalisation. mHealth has other sensor features, like - Magnetometer (M) and Electrocardiogram(ECG), which returns poor accuracies than A and G

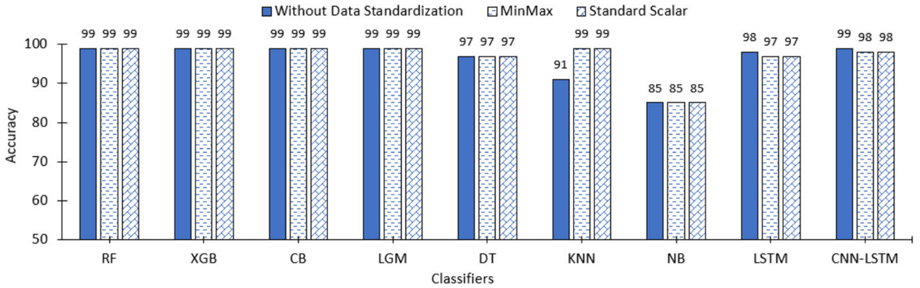


Fig. 26 Accuracy comparison of different classifiers with and without data standardization and normalization on mHealth dataset

Table 10 Accuracy comparison (in %) of various learning models with respect to different sensors of our own dataset

Dataset	Models	A	G	A+G
Own Dataset	RF	92	79	94
	XGB	90	76	93
	CB	88	73	92
	LGM	90	75	92
	DT	88	75	89
	KNN	89	74	91
	NB	63	53	68
	LSTM	90	59	97
	CNN-LSTM	91	63	97

* A = Accelerometer, G = Gyroscope

feature set. The average accuracies for M are between $(50 \pm 71\%)$ and $(15 \pm 21\%)$ for ECG features, respectively. Analysing different feature sets, we can conclude that G and A features yield better accuracies and can easily generalise patterns than the other two. So, we combined both A and G features and trained and tested our model to compare the performance. We managed to achieve comparable performance concerning the original dataset achieved performance. Other combinations of feature sets are also included in our experimental study, and we found out that $(A + G + ECG)$ feature set provides optimal performance results and is better than the original feature set, as well. The detailed comparison of all the sensor-based feature accuracies are briefly described in Table 11.

However, it was observed that in our dataset, A-based features yields better accuracies, but for the mHealth dataset, G-based features yielded better performance results. Based on these findings, we cannot conclude whether A-based features or G-based features are better for activity recognition, but we could certainly confirm that the combination of both A and G features are efficient towards human activity recognition. Further, appropriate mounting location and incorporation of physiological sensors like - ECG are optimal for activity recognition, but it certainly demands more computational power for model training and validation.

Table 11 Accuracy comparison (in %) of various learning models with respect to different sensors of mHealth dataset

Dataset	Models	A	G	M	ECG	A+G	A+G+M	A+G+ECG	A+G+M+ECG
mHealth	RF	95	98	71	17	99	99	99	99
	XGB	94	98	69	21	99	99	99	99
	CB	92	96	67	21	99	99	99	99
	LGM	94	98	69	21	99	99	99	99
	DT	90	98	63	17	97	97	97	97
	KNN	95	97	67	16	96	91	97	91
	NB	75	51	50	13	80	85	88	85
	LSTM	93	97	68	16	98	98	98	98
	CNN-LSTM	93	98	70	15	98	99	99	99

* A = Accelerometer, G = Gyroscope, M = Magnetometer, ECG = Electrocardiogram

Note - All the classifiers incorporated for the experimental analysis are the basic state-of-the-art learning classifiers, and the achieved results can be further improved by analyzing optimal hyperparameters. Models were incorporated to show the effect of different learning approaches towards sensor-based HAR systems.

6 Conclusion

This paper addressed the recent state-of-the-art HAR works in detail, focusing on sensor-based HAR system development. Classification based on sensor technology, learning algorithms and significant scope are discussed, along with its pros and cons. Design issues for efficient HAR system development, data collection metrics with do's and don'ts for good quality dataset generation, and a generic architecture of the Activity Recognition Chain are conferred to exhibit the step-by-step procedure for developing an efficient sensor-based HAR system. Popular state-of-the-art works and public datasets for developing a HAR system are summarized in tables with their methodologies and incorporated evaluation metrics.

Further, a complete step-by-step HAR system design and development has been experimented for comprehensive understanding of empirical human activity recognition chain. A multi-user dataset is generated in an uncontrolled environment with the help of inbuilt smartphone sensors, and with different learning algorithms, and data pre-processing routine we have classified various human activities. Methods and frameworks for overcoming the common problem of class imbalance, effect of sensors types, data standardization, noise removal and many more are also discussed in detail. Upon classification, we achieved the highest accuracy of 99% with deep learning models - LSTM and CNN-LSTM and the lowest accuracy of 85% with shallow learning model - Naive Bayes on the mHealth dataset. Similarly, our dataset achieved the highest accuracy of 97% with the LSTM classifier and the lowest of 68% with the Naive Bayes classifier. The influence of different pre-processing methods and the use of learning methods with key findings are discussed in detail with a verifiable performance result. Finally, the prospective future opportunities are highlighted, and the research direction is laid out for future scope and applications.

Data Availability Data will be made available on reasonable request.

Declarations

Conflicts of interest The authors have no conflict of interest.

Competing Interests No competing interest applicable.

References

1. Romaissa B, Nini B, Sabokrou M, Hadid A (2020) Vision-based human activity recognition: a survey. Multimed Tools Appl 79(11)
2. Mani N, Haridoss P, George B (2021) A wearable ultrasonic-based ankle angle and toe clearance sensing system for gait analysis. IEEE Sensors J 21(6):8593–8603
3. Yuan G, Liu X, Yan Q, Qiao S, Wang Z, Yuan L (2021) Hand gesture recognition using deep feature fusion network based on wearable sensors. IEEE Sensors J 21(1):539–547

4. Amin Choudhury N, Moulik S, Choudhury S (2020) Cloud-based real-time and remote human activity recognition system using wearable sensors. IEEE Int Conf Consum Electron - Taiwan (ICCE-Taiwan), pp 1–2
5. Moulik S, Majumdar S (2019) Fallsense: An automatic fall detection and alarm generation system in iot-enabled environment. IEEE Sensors J 19(19):8452–8459
6. Jobanputra C, Bavishi J, Doshi N (2019) Human activity recognition: A survey. Procedia Comput Sci 155:698–703
7. WHO, Physical activity. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/physical-activity
8. Wang A, Zhao S, Zheng C, Yang J, Chen G, Chang C-Y (2021) Activities of daily living recognition with binary environment sensors using deep learning: a comparative study. IEEE Sensors J 21(4):5423–5433
9. Demrozi F, Pravadelli G, Bihorac A, Rashidi P (2020) Human activimty recognition using inertial, physiological and environmental sensors: a comprehensive survey. IEEE Access 8:210816–210836
10. Lara OD, Labrador MA (2013) A survey on human activity recognition using wearable sensors. IEEE Commun Surv Tutors 15(3):1192–1209
11. Slim SO, Atia A, Elfattah MM, Mostafa M-SM (2019) Survey on human activity recognition based on acceleration data. Int J Adv Comput Sci Appl 10(3)
12. Tian Y, Wang X, Chen L, Liu Z (2019) Wearable sensor-based human activity recognition via two-layer diversity-enhanced multiclassifier recognition method. Sensors, 19(9)
13. Zhang B, Zheng R, Liu J (2021) A multi-source unsupervised domain adaptation method for wearable sensor based human activity recognition: Poster abstract. In: Proceedings of the 20th international conference on information processing in sensor networks (co-located with CPS-IoT Week 2021), ser IPSN '21. Association for Computing Machinery, New York, NY, USA, pp 410–411
14. Wang H, Zhao J, Li J, Tian L, Tu P, Cao T, An Y, Wang K, Li S (2020) Wearable sensor-based human activity recognition using hybrid deep learning techniques. Secur Commun Netw 2020:2132138
15. Mekruksavanich S, Jitpattanakul A (2021) Deep convolutional neural network with rnns for complex activity recognition using wrist-worn wearable sensor data. Electron 10(14)
16. Attal F, Mohammed S, Dedabrishvili M, Chamroukhi F, Oukhellou L, Amirat Y (2015) Physical human activity recognition using wearable sensors. Sensors 15(12):31314–31338
17. Du Y, Lim Y, Tan Y (2019) A novel human activity recognition and prediction in smart home based on interaction. Sensors 19(20)
18. Irvine N, Nugent C, Zhang S, Wang H, NG WWY (2020) Neural network ensembles for sensor-based human activity recognition within smart environments. Sensors 20(1)
19. Tax N (2018) Human activity prediction in smart home environments with lstm neural networks. In: 2018 14th international conference on intelligent environments (IE), pp 40–47
20. Hassan MM, Uddin MZ, Mohamed A, Almogren A (2018) A robust human activity recognition system using smartphone sensors and deep learning. Futur Gener Comput Syst 81:307–313
21. Andrade-Ambriz YA, Ledesma S, Ibarra-Manzano M-A, Oros-Flores MI, Almanza-Ojeda D-L (2022) Human activity recognition using temporal convolutional neural network architecture. Expert Syst Appl 191:116287
22. Jindal S, Sachdeva M, Kushwaha AKS (2022) Deep learning for video based human activity recognition: Review and recent developments. In: Bansal RC, Zemmari A, Sharma KG, Gajrani J (eds) Proceedings of international conference on computational intelligence and emerging power system. Springer, Singapore, pp 71–83
23. Ehatisham-Ul-Haq M, Javed A, Azam MA, Malik HMA, Irtaza A, Lee IH, Mahmood MT (2019) Robust human activity recognition using multimodal feature-level fusion. IEEE Access 7:60736–60751
24. Mliki H, Bouhlel F, Hammami M (2020) Human activity recognition from uav-captured video sequences. Pattern Recognit 100:107140
25. Ke S-R, Thuc HLU, Lee Y-J, Hwang J-N, Yoo J-H, Choi K-H (2013) A review on video-based human activity recognition. Comput 2(2):88–131
26. Kang J, Shin J, Shin J, Lee D, Choi A (2022) Robust human activity recognition by integrating image and accelerometer sensor data using deep fusion network. Sensors 22(1)
27. Ni J, Sarbajna R, Liu Y, Ngu AHH, Yan Y (2021) Cross-modal knowledge distillation for vision-to-sensor action recognition
28. Banjarey K, Sahu SP, Dewangan DK (2022) Human activity recognition using 1d convolutional neural network. In: Shakya S, Balas VE, Kamolphiwong S, Du K-L (eds) Sentimental analysis and deep learning. Springer Singapore, Singapore, pp 691–702
29. Vyas R, Doddabasappla K (2022) Fft spectrum spread with machine learning (ml) analysis of triaxial acceleration from shirt pocket and torso for sensing coughs while walking. IEEE Sensors Lett 6(1):1–4

30. Choudhury NA, Moulik S, Roy DS (2021) Physique-based human activity recognition using ensemble learning and smartphone sensors. IEEE Sensors J 21(15):16852–16860
31. Nandy A, Saha J, Chowdhury C, Singh KPD (2019) Detailed human activity recognition using wearable sensor and smartphones, In: International conference on opto-electronics and applied optics (Optronix), pp 1–6
32. Asim Y, Azam MA, Ehatisham-ul Haq M, Naeem U, Khalid A (2020) Context-aware human activity recognition (cahar) in-the-wild using smartphone accelerometer. IEEE Sensors J 20(8):4361–4371
33. Ignatov A (2018) Real-time human activity recognition from accelerometer data using convolutional neural networks. Appl Soft Comput 62:915–922
34. Chen D, Yongchareon S, Lai EM-K, Sheng, QZ Liesaputra V (2021) Locally-weighted ensemble detection-based adaptive random forest classifier for sensor-based online activity recognition for multiple residents. IEEE Internet Things J 1–1
35. Yu H, Chen Z, Zhang X, Chen X, Zhuang F, Xiong H, Cheng X (2021) Fedhar: Semi-supervised online learning for personalized federated human activity recognition. IEEE Trans Mob Comput 1–1
36. Vakili M, Rezaei M (2021) Incremental learning techniques for online human activity recognition
37. Abdul Haroon PS, Premachand DR (2021) Human activity recognition using machine learning approach. J Robot Control (JRC) 2(5):395–399
38. Biswal A, Nanda S, Panigrahi CR, Cowlessur SK, Pati B (2021) Human activity recognition using machine learning: a review. In: Panigrahi CR, Pati B, Pattanayak BK, Amic S, Li K-C (eds) Progress in advanced computing and intelligent engineering. Springer Singapore, Singapore, pp 323–333
39. Papaleonidas A, Psathas AP, Iliadis L (2021) High accuracy human activity recognition using machine learning and wearable devices' raw signals. J Inf Telecommun 0(0):1–17
40. Subasi A, Khateeb K, Brahimi T, Sarirete A (2020) Chapter 5 - human activity recognition using machine learning methods in a smart healthcare environment. In: Lytras MD, Sarirete A (eds) Innovation in Health Informatics. ser. Next Gen Tech Driven Personalized Med&Smart Healthcare. Academic Press, pp 123–144
41. Thakur D, Guzzo A, Fortino G (2021) t-sne and pca in ensemble learning based human activity recognition with smartwatch*. In: 2021 IEEE 2nd International conference on human-machine systems (ICHMS), pp 1–6
42. Sekiguchi R, Abe K, Yokoyama T, Kumano M, Kawakatsu M (2020) Ensemble learning for human activity recognition. In: Adjunct Proceedings of the 2020 ACM international joint conference on pervasive and ubiquitous computing and proceedings of the 2020 ACM international symposium on wearable computers. ser. UbiComp-ISWC '20. Association for Computing Machinery, New York, NY, USA, p 335–339
43. Kasubi JW, Huchaiah MD (2021) Human activity recognition for multi-label classification in smart homes using ensemble methods. In: Solanki A, Sharma SK, Tarar S, Tomar P, Sharma S, Nayyar A (eds) Artificial intelligence and sustainable computing for Smart City. Springer International Publishing, Cham, pp 282–294
44. Wan S, Qi L, Xu X, Tong C, Gu Z (2020) Deep learning models for real-time human activity recognition with smartphones. Mob Netw Appl 25(2):743–755
45. Xia K, Huang J, Wang H (2020) Lstm-cnn architecture for human activity recognition. IEEE Access 8:56855–56866
46. Mekruksavanich S, Jitpattanakul A (2021) LSTM networks using smartphone data for sensor-based human activity recognition in smart homes. Sensors 21(5)
47. Chinimilli PT, Redkar S, Zhang W (2017) Human activity recognition using inertial measurement units and smart shoes. In: 2017 American control conference (ACC), pp 1462–1467
48. Vu CC, Kim J (2018) Human motion recognition using swcnt textile sensor and fuzzy inference system based smart wearable. Sensors and Actuators A: Physical 283:263–272
49. Lara OD, Labrador MA (2012) A mobile platform for real-time human activity recognition. In: 2012 IEEE consumer communications and networking conference (CCNC), pp 667–671
50. Jain Y, Tang CI, Min C, Kawsar F, Mathur A (2022) Collossl: Collaborative self-supervised learning for human activity recognition. Proc ACM Interact Mob Wearable Ubiquitous Technol 6(1)
51. Lattanzi E, Donati M, Freschi V (2022) Exploring artificial neural networks efficiency in tiny wearable devices for human activity recognition. Sensors 22(7)
52. Xu G, Wan Q, Deng W, Guo T, Cheng J (2022) Smart-sleeve: a wearable textile pressure sensor array for human activity recognition. Sensors 22(5)
53. Ghosal S, Sarkar M, Sarkar R (2022) NoFED-Net: Non-Linear Fuzzy Ensemble of Deep Neural Networks for Human Activity Recognition. IEEE Internet Things J 1-1
54. Rashid N, Demirel BU, Faruque MAA (2022) Ahar: Adaptive cnn for energy-efficient human activity recognition in low-power edge devices. IEEE Internet Things J 1–1

55. Han C, Zhang L, Tang Y, Huang W, Min F, He J (2022) Human activity recognition using wearable sensors by heterogeneous convolutional neural networks. Expert Syst Appl 198:116764
56. Alsaify BA, Almazari MM, Alazrai R, Alouneh S, Daoud MI (2022) A csi-based multi-environment human activity recognition framework. Appl Sci 12(2)
57. Kwon E, Park H, Byon S, Jung E, Lee Y (2018) HaaS(human activity analytics as a service) using sensor data of smart devices. In: 2018 International conference on information and communication technology convergence (ICTC), pp 1500–1502
58. Dehzangi O, Sahu V (2018) Imu-based robust human activity recognition using feature analysis, extraction, and reduction, In: 2018 24th International conference on pattern recognition (ICPR), pp 1402–1407
59. Patel AD, Shah JH (2019) Performance analysis of supervised machine learning algorithms to recognize human activity in ambient assisted living environment. In: 2019 IEEE 16th India council international conference (INDICON), pp 1–4
60. Khokhlov I, Reznik L, Cappos J, Bhaskar R (2018) Design of activity recognition systems with wearable sensors. In: 2018 IEEE sensors applications symposium (SAS), pp 1–6
61. Hong Y, Kim I, Ahn SC, Kim H (2008) Activity recognition using wearable sensors for elder care. In: 2008 Second international conference on future generation communication and networking, vol 2, pp 302–305
62. Choudhury NA, Soni B (2023) An adaptive batch size based-cnn-lstm framework for human activity recognition in uncontrolled environment. IEEE Trans Ind Inform 1–9
63. Fu Z, He X, Wang E, Huo J, Huang J, Wu D (2021) Personalized human activity recognition based on integrated wearable sensor and transfer learning. Sensors 21(3)
64. Mekruksavanich S, Jitpattanakul A (2021) Biometric user identification based on human activity recognition using wearable sensors: An experiment using deep learning models. Electron 10(3)
65. Anguita D, Ghio A, Oneto L, Parra X, Reyes-Ortiz JL (2013) A public domain dataset for human activity recognition using smartphones. In: ESANN
66. Zhang M, Sawchuk A (2012) Usc-had: a daily activity dataset for ubiquitous activity recognition using wearable sensors. In: ACM international conference on ubiquitous computing (Ubicomp) workshop on situation, activity and goal awareness (SAGAware), vol 09, pp 1036–1043
67. Damaševičius R, Maskeliūnas R, Venčkauskas A, Woźniak M (2016) Smartphone user identity verification using gait characteristics. Symmetry 8(10)
68. Yang P, Yang C, Lanfranchi V, Ciravegna F (2022) Activity graph based convolutional neural network for physical activity recognition using acceleration and gyroscope data. IEEE Trans Ind Inform 1–1
69. Ehatisham-ul-Haq M, Murtaza F, Azam MA, Amin Y (2022) Daily living activity recognition in-the-wild: modeling and inferring activity-aware human contexts. Electronics 11(2)
70. Siirtola P, Röning J (2021) Context-aware incremental learning-based method for personalized human activity recognition. J Ambient Intell Humanized Comput 12(12):10499–10513
71. Choudhury NA, Moulik S, Roy DS (2021) Harsense: statistical human activity recognition dataset
72. Malekzadeh M, Clegg RG, Cavallaro A, Haddadi H (2019) Mobile sensor data anonymization. In: Proceedings of the international conference on internet of things design and implementation, ser. IoTDI '19. Assoc Comput Mach, New York, NY, USA , pp 49–58
73. Vavoulas G, Chatzaki C, Malliotakis T, Pediaditis M, Tsiknakis M (2016) The mobiact dataset: recognition of activities of daily living using smartphones. In: 2nd international conference on information and communication technologies for ageing well and e-health, vol 01, pp 143–151
74. Chavarriaga R, Sagha H, Calatroni A, Digumarti ST, Tröster G, J. Millán del R, Roggen D (2013) The opportunity challenge: a benchmark database for on-body sensor-based activity recognition. Pattern Recognition Letters, 34(15):2033–2042, smart Approaches for Human Action Recognition
75. Banos O, Villalonga C, Garcia R, Saez A, Damas M, Holgado-Terriza JA, Lee S, Pomares H, Rojas I (2015) Design, implementation and validation of a novel open framework for agile development of mobile health applications. BioMedical Engineering OnLine 14(2):S6
76. Sztyler T, Stuckenschmidt H (2016) On-body localization of wearable devices: an investigation of position-aware activity recognition. In: 2016 IEEE international conference on pervasive computing and communications (PerCom), pp 1–9
77. Hayashi T, Nishida M, Kitaoka N, Takeda K (2015) Daily activity recognition based on dnn using environmental sound and acceleration signals. In: 2015 23rd European Signal Processing Conference (EUSIPCO), pp 2306–2310
78. Kwapisz JR, Weiss GM, Moore SA (2011) Activity recognition using cell phone accelerometers. SIGKDD Explor Newsl 12(2):74–82
79. Reiss A, Stricker D (2012) Introducing a new benchmarked dataset for activity monitoring. In: 2012 16th international symposium on wearable computers, vol 06

80. Reyes-Ortiz J-L, Oneto L, Ghio A, Samá Monsonís A, Anguita D, Parra F (2014) Human activity recognition on smartphones with awareness of basic activities and postural transitions. In: Artificial neural networks and machine learning - ICANN 2014. Springer International Publishing, Cham, pp 177–184
81. Bhat G, Tran N, Shill H, Ogras UY (2020) w-HAR: an activity recognition dataset and framework using low-power wearable devices. Sensors 20(18)
82. Stisen A, Blunck H, Bhattacharya S, Prentow T, Kjærgaard M, Dey A, Sonne T, Jensen M (2015) Smart devices are different: assessing and mitigating mobile sensing heterogeneities for activity recognition. In: Proceedings of the 13th ACM conference on embedded networked sensor systems, ser. SenSys '15. Assoc Comput Mach, New York, NY, USA, pp 127–140
83. Khan IU, Afzal S, Lee JW (2022) Human activity recognition via hybrid deep learning based model. Sensors 22(1)
84. Mutegeki R, Han DS (2020) A cnn-lstm approach to human activity recognition, In: 2020 international conference on artificial intelligence in information and communication (ICAIIC), pp 362–366
85. Abdel-Basset M, Hawash H, Chakrabortty RK, Ryan M, Elhoseny M, Song H (2021) St-deephar: Deep learning model for human activity recognition in ioht applications. IEEE Internet of Things J 8(6):4969–4979
86. Cruciani F, Vafeiadis A, Nugent C, Cleland I, McCullagh P, Votis K, Giakoumis D, Tzovaras D, Chen L, Hamzaoui R (2020) Feature learning for human activity recognition using convolutional neural networks. CCF Trans Pervasive Comput Interact 2(1):18–32
87. Dua N, Singh S, Semwal V (2021) Multi-input cnn-gru based human activity recognition using wearable sensors. Comput 103:1–18
88. Kwon H, Tong C, Haresamudram H, Gao Y, Abowd GD, Lane ND, Plötz T (2020) Imutube: Automatic extraction of virtual on-body accelerometry from video for human activity recognition. Proc ACM Interact Mob Wearable Ubiquitous Technol 4(3)
89. Garcia-Gonzalez D, Rivero D, Fernandez-Blanco E, Luaces MR (2023) New machine learning approaches for real-life human activity recognition using smartphone sensor-based data. Knowledge-Based Syst 262:110260
90. Panja AK, Rayala A, Agarwala A, Neogy S, Chowdhury C (2023) A hybrid tuple selection pipeline for smartphone based human activity recognition. Expert Syst Appl 217:119536
91. Yadav SK, Sai S, Gundewar A, Rathore H, Tiwari K, Pandey HM, Mathur M (2022) CSITime: privacy-preserving human activity recognition using WiFi channel state information. Neural Networks 146:11–21
92. Ismail Fawaz H, Lucas B, Forestier G, Pelletier C, Schmidt DF, Weber J, Webb GI, Idoumghar L, Muller P-A, Petitjean F (2020) Inceptiontime: finding alexnet for time series classification. Data Min Knowl Discov 34(6):1936–1962
93. Rashid N, Demirel BU, Faruque MAA (2022) Ahar: adaptive cnn for energy-efficient human activity recognition in low-power edge devices. IEEE Internet Things J 1–1
94. Luo F, Khan S, Huang Y, Wu K (2021) Binarized neural network for edge intelligence of sensor-based human activity recognition. IEEE Trans Mob Comput 1–1
95. Taylor W, Shah SA, Dashtipour K, Zahid A, Abbasi QH, Imran MA (2020) An intelligent non-invasive real-time human activity recognition system for next-generation healthcare. Sensors 20(9)
96. Guo Y, Chu Y, Jiao B, Cheng J, Yu Z, Cui N, Ma L (2021) Evolutionary dual-ensemble class imbalance learning for human activity recognition. IEEE Trans Emerg Top Comput Intell 1–12
97. Khaled H, Abu-Elnasr O, Elmougy S, Tolba AS (2021) Intelligent system for human activity recognition in iot environment. Complex Intell Syst
98. Rahman A, Hassan I, Ahad MAR (2021) Nurse care activity recognition: a cost-sensitive ensemble approach to handle imbalanced class problem in the wild. Assoc Comput Mach, New York, NY, USA, pp 440–445
99. Hamad RA, Yang L, Woo WL, Wei B (2020) Joint learning of temporal models to handle imbalanced data for human activity recognition. Appl Sci 10(15)
100. Choudhury NA, Soni B (2022) Effect of shallow and ensemble learning models for human activity recognition in uncontrolled environment. In: 2022 IEEE 19th India council international conference (INDICON), pp 1–6
101. Dua N, Singh SN, Semwal VB, Challa SK (2023) Inception inspired CNN-GRU hybrid network for human activity recognition. Multimed Tools Appl 82(4):5369–5403
102. Liu K, Liu W, Ma H, Tan M, Gan C (2021) A real-time action representation with temporal encoding and deep compression. IEEE Trans Circ Syst Vid Technol 31(2):647–660
103. Choudhury NA, Soni B (2022) An efficient cnn-lstm approach for smartphone sensor-based human activity recognition system. In: 2022 5th International conference on computational intelligence and networks (CINE), pp 01–06

104. Li X, He Y, Fioranelli F, Jing X (2021) Semisupervised human activity recognition with radar micro-doppler signatures. IEEE Trans Geosci Remote Sens 1–12
105. Tang CI, Perez-Pozuelo I, Spathis D, Brage S, Wareham NJ, Mascolo C (2021) SelfHAR: improving human activity recognition through self-training with unlabeled data. CoRR arXiv:2102.06073
106. Hassan I, Mursalin A, Salam, RB, Sakib N, Haque HMZ (2021) AutoAct: an auto labeling approach based on activities of daily living in the wild domain. In: 2021 joint 10th international conference on informatics, electronics vision (ICIEV) and 2021 5th international conference on imaging, vision pattern recognition (icIVPR), pp 1–8
107. Chen X, Liu W, Liu X, Zhang Y, Han J, Mei T (2022) Maple: Masked pseudo-labeling autoencoder for semi-supervised point cloud action recognition, In: Proceedings of the 30th ACM international conference on multimedia, ser. MM '22. Assoc Comput Mach, New York, NY, USA, pp 708–718
108. Liu K, Liu W, Gan C, Tan M, Ma H (2018) T-C3D: Temporal convolutional 3d network for real-time action recognition. Proceedings of the AAAI Conference on Artificial Intelligence 32(1)
109. Chen L, Hoey J, Nugent CD, Cook DJ, Yu Z (2012) Sensor-based activity recognition. IEEE Trans Syst Man Cybern Part C (Appl Revi) 42(6):790–808
110. Wang J, Chen Y, Hao S, Peng X, Hu L (2019) Deep learning for sensor-based activity recognition: a survey. Pattern recognition letters, deep learning for. Pattern Recognit 119:3–11
111. Sun Z, Ke Q, Rahmani H, Bennamoun M, Wang G, Liu J (2023) Human action recognition from various data modalities: A review. IEEE Trans Pattern Anal Mach Intell 45(3):3200–3225
112. Ramanujam E, Perumal T, Padmavathi S (2021) Human activity recognition with smartphone and wearable sensors using deep learning techniques: a review. IEEE Sensors J 21(12):13029–13040
113. Awan MA, Guangbin Z, Kim H-C, Kim S-D (2015) Subject-independent human activity recognition using smartphone accelerometer with cloud support. Int J Ad Hoc Ubiquit Comput 20(3):172–185
114. Hoang ML, Pietrosanto A (2021) A new technique on vibration optimization of industrial inclinometer for mems accelerometer without sensor fusion. IEEE Access 9:20295–20304
115. Park S, Gil M-S, Im H, Moon Y-S (2019) Measurement noise recommendation for efficient kalman filtering over a large amount of sensor data. Sensors 19(5)
116. Chen Y, Li D, Li Y, Ma X, Wei J (2017) Use moving average filter to reduce noises in wearable ppg during continuous monitoring. In: Giokas K, Bokor L, Hopfgartner F (eds) eHealth 360°. Springer International Publishing, Cham, pp 193–203
117. Jadhav A, Pramod D, Ramanathan K (2019) Comparison of performance of data imputation methods for numeric dataset. Appl Artif Intell 33(10):913–933
118. McKinney W (2010) Data structures for statistical computing in python. In: van der Walt S, Millman J (eds) Proceedings of the 9th python in science conference, pp 56–61
119. Harris CR, Millman KJ, van der Walt SJ, Gommers R, Virtanen P, Cournapeau D, Wieser E, Taylor J, Berg S, Smith NJ, Kern R, Picus M, Hoyer S, van Kerkwijk MH, Brett M, Haldane A, del Río JF, Wiebe M, Peterson P, Gérard-Marchant P, Sheppard K, Reddy T, Weckesser W, Abbasi H, Gohlke C, Oliphant TE (2020) Array programming with NumPy. Nature 585(7825):357–362
120. Bagozi A, Bianchini D, De Antonellis V, Garda M, Marini A (2019) A relevance-based approach for big data exploration. Futur Gene Comput Syst 101:51–69
121. Midway SR (2020) Principles of effective data visualization. Patterns 1(9):100141
122. Wang H, Bah MJ, Hammad M (2019) Progress in outlier detection techniques: a survey. IEEE Access 7:107964–108000
123. Suto J, Oniga S, Sitar PP (2016) Feature analysis to human activity recognition. Int J Comput Commun Control 12(1):116–130
124. Yala N, Fergani B, Fleury A (2015) Feature extraction for human activity recognition on streaming data. In: 2015 International symposium on innovations in intelligent systems and applications (INISTA), pp 1–6
125. Al Machot F, Mayr HC (2016) Improving human activity recognition by smart windowing and spatio-temporal feature analysis. In: Proceedings of the 9th ACM international conference on pervasive technologies related to assistive environments. ser. PETRA '16. Association for Computing Machinery, New York, NY, USA
126. Chen Z, Zhang L, Cao Z, Guo J (2018) Distilling the knowledge from handcrafted features for human activity recognition. IEEE Trans Ind Inform 14(10):4334–4342
127. Berrar D (2019) Cross-validation. In: Ranganathan S, Gribskov M, Nakai K, Schönbach C (eds) Encyclopedia of bioinformatics and computational biology. Academic Press, Oxford, pp 542–545
128. Ramezan CA, Warner TA, Maxwell AE (2019) Evaluation of sampling and cross-validation tuning strategies for regional-scale machine learning classification. Remote Sens 11(2)
129. Schaffer C (1994) Selecting a classification method by cross-validation. Machine Learning 13

130. Wong T-T, Yeh P-Y (2020) Reliable accuracy estimates from k-fold cross validation. IEEE Trans Knowl Data Eng 32(8):1586–1594
131. Yadav S, Shukla S (2016) Analysis of k-fold cross-validation over hold-out validation on colossal datasets for quality classification. In: IEEE 6th international conference on advanced computing (IACC), pp 78–83
132. Wong T-T (2015) Performance evaluation of classification algorithms by k-fold and leave-one-out cross validation. Pattern Recognit 48(9):2839–2846
133. Kearns M, Ron D (1999) Algorithmic stability and sanity-check bounds for leave-one-out cross-validation. Neural Comput 11(6):1427–1453
134. Xu Q-S, Liang Y-Z (2001) Monte carlo cross validation. Chemometr Intell Lab Syst 56(1):1–11
135. Douzas G, Bacao F, Last F (2018) Improving imbalanced learning through a heuristic oversampling method based on k-means and smote. Inf Sci 465:1–20

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.