



Deep-HAR: an ensemble deep learning model for recognizing the simple, complex, and heterogeneous human activities

Prabhat Kumar¹  · S Suresh¹

Received: 4 August 2021 / Revised: 28 April 2022 / Accepted: 31 January 2023 /
Published online: 23 February 2023

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023

Abstract

The recognition of human activities has become a dominant emerging research problem and widely covered application areas in surveillance, wellness management, healthcare, and many more. In real life, the activity recognition is a challenging issue because human beings are often performing the activities not only simple but also complex and heterogeneous in nature. Most of the existing approaches are addressing the problem of recognizing only simple straightforward activities (e.g. walking, running, standing, sitting, etc.). Recognizing the complex and heterogeneous human activities are a challenging research problem whereas only a limited number of existing works are addressing this issue. In this paper, we proposed a novel Deep-HAR model by ensembling the Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) for recognizing the simple, complex, and heterogeneous type activities. Here, the CNNs are used for extracting the features whereas RNNs are used for finding the useful patterns in time-series sequential data. The activities recognition performance of the proposed model was evaluated using three different publicly available datasets, namely WISDM, PAMAP2, and KU-HAR. Through extensive experiments, we have demonstrated that the proposed model performs well in recognizing all types of activities and has achieved an accuracy of 99.98%, 99.64%, and 99.98% for simple, complex, and heterogeneous activities respectively.

Keywords Human activity recognition · Class imbalance problem · CNNs · RNNs · Wireless sensor technology

✉ Prabhat Kumar
prabhat.kumar13@bhu.ac.in

¹ Department of Computer Science, Institute of Science, Banaras Hindu University, Varanasi 221 005, India

1 Introduction

Due to the advancements in wireless sensor technology, Human Activity Recognition (HAR) has been an emerging research area in recent years. Typical HAR application domains include activities analysis in the smart home [23], surveillance [44], wellness management [41], elders caring [40], gesture recognition [25], abnormal activities detection [7], healthcare [45], body temperature and indoor condition monitoring in quarantine due to COVID-19 [18], physical exercise recognition in the gym [9, 21], patients caring [10], and more. Currently, the research trends are varied among sensors, images, and video-based data for recognizing the activities of human beings. The sensor-based technique, however, has attracted the interest of researchers due to its low cost, ease of implementation, location independence, and non-harmful free radiation. The accelerometer and gyroscope sensors are widely utilized in digital devices such as smartphones and smartwatches for activity recognition [32]. Sensor data acquisition, segmentation, feature extraction, model training and validation, and classification are the five phases in which activity recognition tasks are typically accomplished, as illustrated in Fig. 1.

The first phase of the HAR system is to continuously acquire the sensor data while the subjects (e.g. humans) perform the activities using embedded sensors. Here, we need to apply data preprocessing for removing anomalies and outliers. The second phase is segmentation, responsible for slicing the time-series raw sensor data into equal sizes of window length. The third phase is feature extraction, extracting relevant useful features based on the time, frequency, and time-frequency domains. However, the segmentation and feature extraction needs to be done carefully because the classification performance is directly influenced by segment length and the quality of the features extracted from the sensor data. In the model training and validation phase, a suitable model (either machine or deep learning) is trained and validated by optimizing the parameters as per the application's needs. Finally, the classification phase recognizes the activity class labels on the input streaming of sensor data. Both machine learning and deep learning models have been widely used for HAR applications. Recently, the deep learning techniques gained momentum and outperform the traditional machine learning techniques which require essential sensor data preprocessing, lack of unique procedures for feature extraction, and domain knowledge experts.

In real life, human beings are not only performing simple activities (one after another activity) but also performs complex (a set of sequential temporal sub-activities), and heterogeneous activities (collection activity classes that differ from each other in terms of their associated

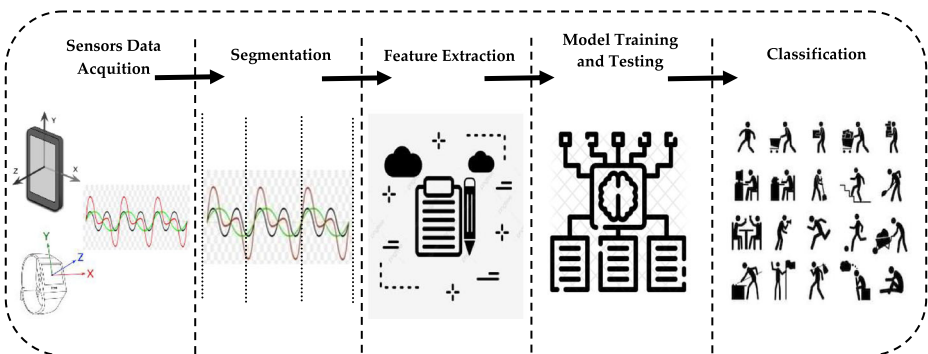


Fig. 1 Classical block diagram of HAR

actions). The problem of recognizing just simple activities (e.g. walking, running, standing, sitting, etc.) has been addressed in the majority of existing approaches. Recognizing complex and heterogeneous activities, on the other hand, is a difficult research challenge that necessitates very sophisticated and competent models. The problem of recognizing complex and heterogeneous activities does not receive much attention among the researchers and only a few existing works are addressing these activities [26]. Moreover, creating models with the capability of recognizing more complex and heterogeneous activities can further widen the application scope of the HAR.

In addition, there is a lack of research work that provides globally accepted solutions for the recognition of the simple, complex, and heterogeneous activities. This attracted our interest and motivated us to do this research work. In this study, we proposed a novel ensemble deep learning model for identifying simple, complex, and heterogeneous activities by recasting the HAR issue as a time-series based pattern classification challenge. To improve the performance, the ensemble learning technique combines various individual models. Deep ensemble learning models with multilayer processing architecture outperform shallow or traditional classification models in terms of recognition rates by combining the advantages of deep learning with ensemble learning [11]. The deep ensemble models have covered various range of application areas such as face recognition [29], cancer prediction [51], detection of COVID-19 on CT images [54], bioinformatics [3], sustainable business management [14, 36], edge computing [19], and more. The proposed novel approach is named as Deep-HAR model which is an ensemble deep learning model using Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) [27, 30]. This ensemble model have been used for two different purposes i.e. the convolutional layers are the fundamental building block of the CNNs, used for mining the effective features from raw dataset whereas recurrent layers in RNNs for activities classification whose embedded memory cell remembers the previous time series activities. As different types of activities (simple, complex, and heterogeneous) differ in their characteristics, classification models need to be tuned specifically for better recognition of each activity category. The deep ensemble model is the ideal choice for recognizing different activity types and the proposed model combines the beneficial features of CNNs (feature extraction) and RNNs (classification). In summary, we highlight the major research contributions addressed in this paper as follows.

1. We proposed a novel Deep-HAR model following the concept of the two-step recognition process. In the first step, the proposed model learns and extracts the efficient features from raw sensory data using current and temporal activity dependencies, accomplished through convolutional layers. The recurrent layers with the association of memory cells performst the activity recognition task in the second step.
2. The convolutional layers in CNNs can directly learn and extract the efficient spatial-temporal features from raw sensor data which need not require manual feature extraction and the feature engineering.
3. The proposed model needs a little bit of preprocessing of the experimental datasets, which makes undoubtedly accepted and suitable for the deployment of real-time activities recognition system.
4. The detailed comparative study on the recognition performance of our proposed model with recent publications on publicly available datasets are presented. The WISDM, PAMAP2, and KU-HAR datasets are used for simple, complex, and heterogeneous activities, respectively and experimentally demonstrated that the proposed model outperforms over existing models.

5. The experimental datasets used are suffering from the class imbalanced problem which usually affects the performance of classifiers. However, our proposed model has robustness against the class imbalanced problem.

The remaining part of the paper is unfolded as follows. Section 2 details the related literature review for HAR whereas the problem statement is given in section 3. Section 4 gives an overview of the description of experimental datasets. The experimental materials and methods are described in section 5. An in-depth discussion of the proposed Deep-HAR model is presented in section 6 and section 7 gives information regarding the experimental results and discussion. Finally, the summary of our research work and conclusions are presented in section 8.

2 Literature review

In recent research, various machine and deep learning techniques are predominantly used to accomplish the HAR. Earlier, researchers have widely used the classical machine learning techniques for activities recognition including the Random Forest (RF) [49], Support Vector Machine (SVM) [43], XG Boost classifier [53], Na ve Bayes [48], and more. The effectiveness of traditional machine learning classifiers is significantly reliant on manual feature extraction in most cases. Domain expertise limits this, as it is time-consuming and resource-intensive. To address these issues, the researchers have started to prefer deep learning techniques. Recent advancements in sensor-based HAR have revealed that deep learning algorithms, rather than relying on time-consuming manual feature learning on raw data, have produced remarkable performance on difficult activity detection problems with minimal feature engineering [46]. To combat manual feature engineering, the most common deep learning algorithms have been applied includes CNN [2, 35, 50], RNN [20], Generative Adversarial Networks (GAN) [34], LSTM [4], and their variant forms. Our objective behind this work is to propose a model to detect the user's activities ranging from simple to heterogeneous types. We have used three different publicly available datasets, WISDM, PAMAP2, and KU-HAR, as a representation of simple, complex, and heterogeneous activities, respectively. More details on these datasets are given in the subsequent section.

The recent works for HAR on the WISDM dataset include the Unsupervised Deep Learning Assisted Reconstructed Coder (UDR-RC) [22], 1D-CNN [13], Att-based Residual Network [12], lightweight RNN-LSTM [1], and Adaptive Feature Fusion Network (AFFNet) [47]. These models achieved the accuracy rate of 97.50%, 94.20%, 98.85%, 95.78%, and 94.60%, respectively. For the PAMAP2 dataset, the researchers have recently used the One-shot learning methods [24], Deep Learning Architecture for Physical Activity Recognition (DELAPAR) [15], Att-based Residual Network [12], Residual Network and Heterogeneous CNN (ResNet+HC) [17], and float CNN [5] with an accuracy rate of 84.41%, 96.62%, 93.16%, 92.97%, and 85.23%, respectively. The KU-HAR dataset was used by the RF classifier [33] and the transformer model [8], with accuracy rates of 90.00% and 99.20%, respectively.

Recognizing complex and heterogeneous activities have not gained much attention unlike recognizing simple straightforward activities. In [42], the authors have proposed the hybrid method, a combination of bi-directional Long-Short Term Memory (BiLSTM) and Skip-Chain Conditional random field (SCCRF) as a BiLSTM-SCCRF approach for recognition of concurrent and interleaved activities using the Kasteren HouseB from Kasteren and Kyoto 3 from

CASAS benchmark datasets. Their proposed method has achieved an average accuracy rate of more than 93.00%. The authors in [52], have proposed a novel knowledge-driven approach for the recognition of concurrent activities (KCAR). This approach has been applied on a large scale of the real-world dataset and achieved an accuracy rate of 91.00%. In [31], the authors have proposed a shapelet-based approach (i.e. dictionary for time series patterns) for recognition of complex type activities using the opportunity experimental dataset. This approach has achieved an average accuracy rate of 96.00%. The authors in [16], have proposed a novel Emerging Patterns based approach for the detection of sequential, interleaved, and concurrent Activity Recognition (epSICAR). The experimental results with a segment length of 15-seconds achieved the accuracy rate of 90.96%, 87.98%, and 78.58% for sequential, interleaved, and complex type activities, respectively. The authors have conducted experimental studies in their own real smart home. The description of the recently published works related to our research work is summarized in Table 1.

Most of the existing works in the literature are competent, have novelty, and are innovative in model's architecture and performance but are designed for detection of simple straightforward activities. Moreover, some research works detect complex activities but lack in terms of performance and in detecting all types of activities. Some also work utilize external sensors to gather high-quality data to increase the recognition rates and focus only on single activity detection. In this regard, we have proposed a novel deep ensemble approach named as Deep-HAR model which is an ensemble of CNNs and RNNs model with the capability to detect the simple, complex, and heterogeneous types of activities.

3 Problem statement

The research problem statement for recognizing the simple, complex, and heterogeneous activities can be formulated using the following equations. Let's assume that given three different datasets $D = (d_1, d_2, d_3)$ for simple (S), complex (C), and heterogeneous (H) activities. Furthermore, the given datasets have split into training (X), validation (Y), and testing (Z), mentioned using the Eqs. (1), (2), and (3):

$$\text{Dataset } (d_1) = [\text{Training } (X_{d1}), \text{ Validating } (Y_{d1}), \text{ Testing } (Z_{d1})] \quad (1)$$

$$\text{Dataset } (d_2) = [\text{Training } (X_{d2}), \text{ Validating } (Y_{d2}), \text{ Testing } (Z_{d2})] \quad (2)$$

$$\text{Dataset } (d_3) = [\text{Training } (X_{d3}), \text{ Validating } (Y_{d3}), \text{ Testing } (Z_{d3})] \quad (3)$$

In Eqs. (4), (5), and (6), the split datasets X_{dn} , Y_{dn} , and Z_{dn} consists of (t) number of observations, where n = dataset number.

$$X_{dn} = \{x_1, x_2, x_3, \dots, x_t\} \quad (4)$$

$$Y_{dn} = \{y_1, y_2, y_3, \dots, y_t\} \quad (5)$$

$$Z_{dn} = \{z_1, z_2, z_3, \dots, z_t\} \quad (6)$$

Table 1 Summary of related research works

Reference	Methods	Datasets	Accuracy (%)	Activities Types				
				Sequential	Complex	Concurrent	Interleaved	Heterogeneous
[33]	Random Forest	KU-HAR	90.00	×	×	×	×	✓
[42]	BiLSTM-SCRF	Kasteren HouseB and Kyoto 3	93.00	×	×	✓	✓	×
[52]	KCAR	Own dataset	91.00	×	×	✓	×	×
[31]	Shapelet-based approach	Opportunity	96.00	×	✓	×	×	×
[16]	epSICAR	Own dataset	90.96, 87.98, and 78.58	✓	✓	×	✓	×

The simple activities label (S_m) = $\{s_1, s_2, s_3, \dots, s_{m1}\}$, complex activities label (C_m) = $\{c_1, c_2, c_3, \dots, c_{m2}\}$, and heterogeneous activities label (H_m) = $\{h_1, h_2, h_3, \dots, h_{m3}\}$ are unique, and the number of total activity labels equals to $m_1+m_2+m_3$. Now, the prediction model (M) uses the X_{dn} , Y_{dn} , and Z_{dn} samples for training, validating, and testing, respectively.

$$\text{Training } (M) \in \{X_{d1}, X_{d2}, X_{d3}, \dots, X_{dn}\} \quad (7)$$

$$\text{Validating } (M) \in \{Y_{d1}, Y_{d2}, Y_{d3}, \dots, Y_{dn}\} \quad (8)$$

$$\text{Testing } (M) \in \{Z_{d1}, Z_{d2}, Z_{d3}, \dots, Z_{dn}\} \quad (9)$$

In Eq. 7, the training samples X_{dn} are used for model training. However, for better performance, we need to optimize the parameter values, which called hyperparameter tuning. To accomplish the hyperparameter tuning task, Eq. 8 helps for validating the model using Y_{dn} . Finally, the designed model (M) use the testing Z_{dn} samples for assigning each row of the observation with activity class labels using (Eq. 9).

4 Experimental datasets description

The technical background details of experimental datasets are discussed in this section. For experiments, we have used the WISDM [28], PAMAP2 [37], and KU-HAR [33] datasets concerning the simple, complex, and heterogeneous activities, respectively. The summarized information regarding the experimental datasets has given in Table 2.

For simple-type activities, we have used WISDM as the experimental dataset. Simple activities are those activities that cannot be divided in to sub-activities. A single smartphone-based sensor (X , Y , and Z axis) has used to gather the data and mounted in the pocket of front leg pants. In the experimental context, 36 individuals are participated to perform the six activities. The annotated six activities in the WISDM dataset, are listed as Standing, Sitting, Downstairs, Upstairs, Jogging, and Walking. These activities were sampled at 20 Hz on a triaxial accelerometer sensor.

The complex activities contains a set of sequentially temporal sub-activities. We have used PAMAP2 as the experimental dataset for complex-type activities. This dataset was collected using Colibri wireless Inertial Measurement Units (IMU) that contains two accelerometers, a gyroscope, and a magnetometer sensors, mounted at the Chest, Wrist, and ankle, respectively. PAMAP2 dataset includes a total of 18 daily living activities. There was a collection constraint that a total of nine subjects can perform any twelve activities out of the listed 18 activities. The activities in the PAMAP2 dataset, are listed as Lying, Sitting, Standing, Walking, Running, Cycling, Nordic Walking, Watching TV, Computer Work, Car Driving, Ascending Stairs, Descending Stairs, Vacuum Cleaning, Ironing, Folding Laundry, House Cleaning, Playing Soccer, Rope Jumping, and Other (Transient Activities). These activities were sampled at 100 Hz.

Table 2 Description of experimental datasets

Dataset	WISDM [28]	PAMAP2 [37]	KU-HAR [33]
Sensors (#)	1	3	2
Sensors	Accelerometer	Colibri wireless Inertial Measurement Units (IMU)	Accelerometer and Gyroscope
Activities (#)	6	18	18
Activities	Standing, Sitting, Downstairs, Upstairs, Jogging, and Walking	Lying, Sitting, Standing, Walking, Running, Cycling, Nordic Walking, Watching TV, Computer Work, Car Driving, Ascending Stairs, Descending Stairs, Vacuum Cleaning, Ironing, Folding Laundry, House Cleaning, Playing Soccer, Rope Jumping, and Other (Transient Activities)	Walk-Circle, Walk-Backward, Table-Tennis, Push-Up, Run, Jump, Stair-Down, Stair-Up, Walk, Sit-Up, Pick, Lay-Stand, Talk-Sit, Lay, Talk-Stand, Stand, Sit, and Stand-Sit
Placement	Front pants leg pocket	Chest, Wrist, and ankle	Waist
Sampling Rate (Hz)	20	100	100
Years	2010	2012	2021
Subjects	36	9	90

For heterogeneous-type activities, we have used KU-HAR as the experimental dataset. The heterogeneous activities may have common subactivities but still contains unique patterns among the group of activities in particular dataset. In other words, the activity classes are different from each other in terms of the associated actions, although some of them are similar, such as walking forward, backward, and in circles. The process of dataset collection was accomplished using smartphone-based accelerometer and a gyroscope tri-axial sensors (*X*, *Y*, and *Z* axis), mounted at the waist. In the experimental context, a total of 90 individuals participated to perform the prescribed eighteen activities. These activities in the KU-HAR dataset, are listed as Walk-Circle, Walk-Backward, Table-Tennis, Push-Up, Run, Jump, Stair-Down, Stair-Up, Walk, Sit-Up, Pick, Lay-Stand, Talk-Sit, Lay, Talk-Stand, Stand, Sit, and Stand-Sit that have been sampled at 100 Hz. The activity sample distributions over the experimental datasets are shown in Table 3 and graphically demonstrated in Fig. 2.

Among the experimental datasets, the WISDM has been extremely influenced by the class imbalanced problem. This indicates that the distribution of data samples is highly skewed. In the WISDM dataset, the occurrence of sitting and standing activity classes are too less whereas walking and jogging have the highest number of samples. However, the remaining activities (upstairs and downstairs) have the average number of occurrences. The PAMAP2 dataset has also the occurrence of the class imbalanced problem, like the WISDM dataset. The rope jumping activity has the lowest number of samples, followed by running, descending stairs, and ascending stairs activity samples whereas ironing and walking consists of the highest number of activity samples. However, the cycling, vacuum cleaning, sitting, nordic walking, standing, and lying have equivalent number of samples in the PAMAP2 dataset. The KU-HAR dataset also suffers from class imbalance problem. The walk circle activity has the lowest samples whereas the stand-sit contained the highest number of samples. However, the remaining sixteen activities have the approximately equivalent number of samples distribution.

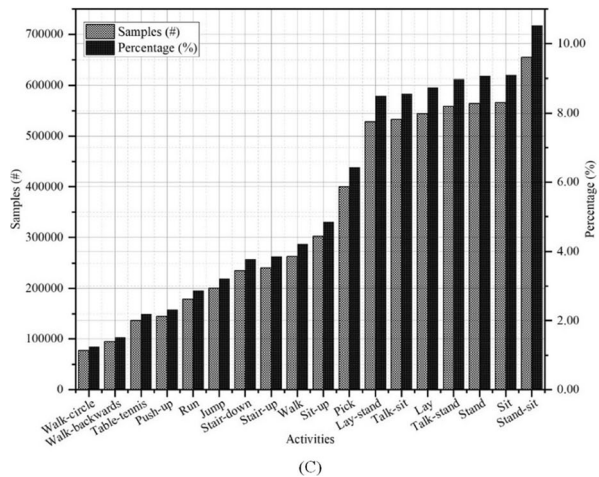
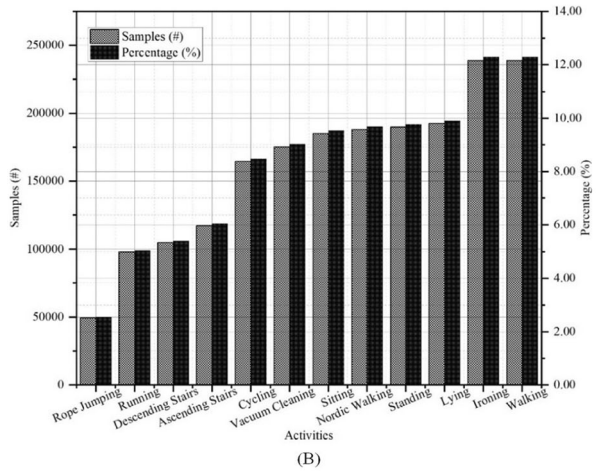
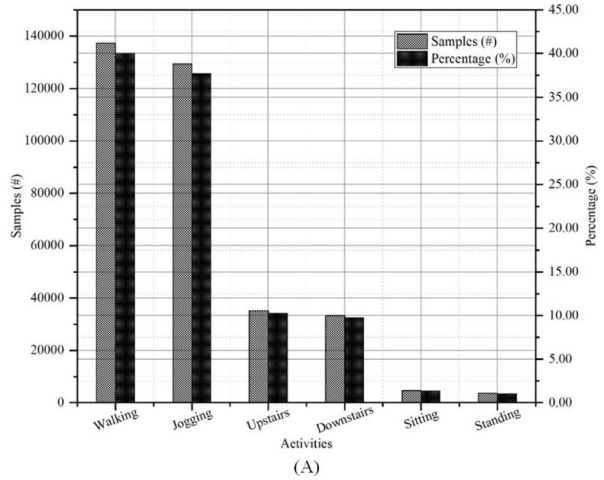
Table 3 Distribution of activity samples with contributed percentage over the experimental datasets

Dataset	Activity Name	Samples (#)	Percentage (%)
WISDM	Walking	1,37,375	40.00
	Jogging	1,29,392	37.67
	Upstairs	35,137	10.23
	Downstairs	33,358	09.71
	Sitting	4599	01.33
	Standing	3555	01.03
	Total	3,43,416	100.00
PAMAP2	Rope Jumping	49,360	02.54
	Running	98,199	05.05
	Descending Stairs	1,04,944	05.40
	Ascending Stairs	1,17,216	06.03
	Cycling	1,64,600	08.47
	Vacuum Cleaning	1,75,353	09.02
	Sitting	1,85,188	09.53
	Nordic Walking	1,88,107	09.68
	Standing	1,89,931	09.77
	Lying	1,92,523	09.90
	Ironing	2,38,690	12.28
	Walking	2,38,761	12.28
	Total	19,42,872	100.00
	KU-HAR	Walk-circle	77,664
Walk-backwards		94,537	01.51
Table-tennis		1,36,314	02.19
Push-up		1,44,895	02.32
Run		17,8880	02.87
Jump		2,00,300	03.21
Stair-down		2,34,678	03.77
Stair-up		2,40,350	03.86
Walk		2,62,887	04.22
Sit-up		3,02,085	04.85
Pick		4,00,353	06.43
Lay-stand		5,28,414	08.49
Talk-sit		5,32,901	08.56
Lay		5,44,237	08.74
Talk-stand		5,58,800	08.98
Stand		5,64,052	09.06
Sit		5,65,796	09.09
Stand-sit		6,54,696	10.52
Total		62,21,839	100.00

5 Experimental materials and methods

The goal behind the Deep-HAR is to architect a common model for recognizing the simple, complex, and heterogeneous activity patterns. The graphical representation of different activity patterns on a time series scale splitted into n – segments ($t_1, t_2, t_3, \dots, t_n$), is shown in Fig. 3. The n number of simple activities ($Sa_1, Sa_2, Sa_3, \dots, Sa_n$) performed in the sequential mode which means no embedding of sub-activities, are shown in Fig. 3(A). The n number of complex activities ($Ca_1, Ca_2, Ca_3, \dots, Ca_n$) containing the set of temporal activities ($Ctemp_1, \dots, Ctemp_n$), such as washing hands: opening the water tap, using a shop, washing hands, and closing the water tap, are shown in Fig. 3(B). Figure. 3(C) contains the n number of heterogeneous activities ($Ha_1, Ha_2, Ha_3, \dots, Ha_n$) hold unique properties ($Htemp_1, \dots, Htemp_n$) that make them differ from simple and complex activities. In the heterogeneous

Fig. 2 Graphical view of activity samples distribution for (A) WISDM, (B) PAMAP2, and (C) KU-HAR dataset



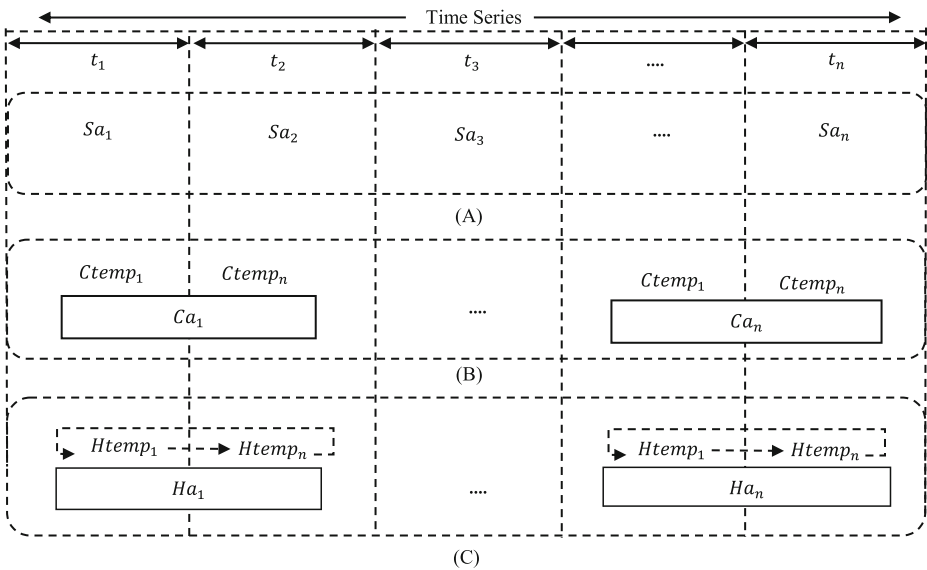


Fig. 3 Graphical view of activity patterns of (A) simple, (B) complex, and (C) heterogeneous activities

category, the subject performs the activity by repeatedly switching between two or more associated activities. For instance, in stand-sit and lay-stand activities, the subject repeatedly stands and sits and standing up and laying down repeatedly.

5.1 Convolutional Neural Networks (CNNs)

The CNN is a special class of neural networks that process the grid-like data. The commonly used CNN architectures consist of one dimension, two dimensions, and three dimensions. The two or three dimensional CNNs are mostly used for handling image and video data processing. However, the one dimensional CNN has used signal processing or vector data manipulation [38]. The layered architecture of CNN is built up with convolutional, pooling, and fully connected layers. The features from the experimental dataset are extracted by the first two convolutional and pooling layers. Finally, the fully connected layer is used for classification [39]. The graphical view of simple CNN architecture has shown in Fig. 4. The mathematical background details of CNN architecture are demonstrated as follows:

5.1.1 Convolutional layers

The convolutional layers are the primary building block of CNN, which extract the most suitable and efficient features from the raw dataset. The convolutional kernels are scanned over the complete dimension of raw data, then compute the dot product between input dimensions and filter values. With this event, CNN quickly learns the effective spatial and temporal domain features. Equation (10) is used for calculating the outcomes of the convolutional layer [6].

$$O_i^k = b_i^k + \sum_{j=1}^{N_{k-1}} I_i^{k-1} \otimes W_{ji}^{k-1} \tag{10}$$

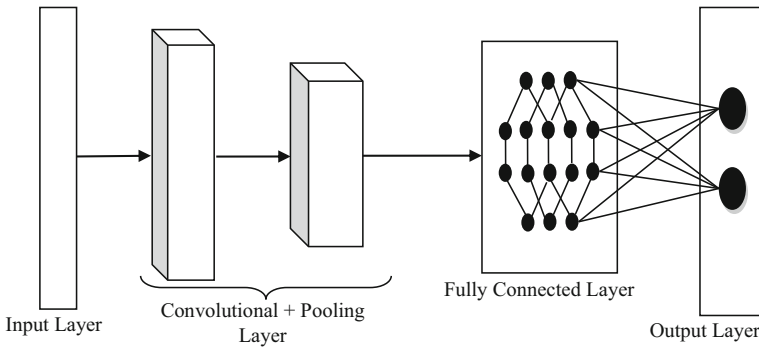


Fig. 4 The basic building block of the CNN architecture

where O_i^k defined as outcomes of convolutional layer, b_i^k is bias of i^{th} neuron at layer K , I_i^{k-1} is the output of i^{th} neuron at layer $k - 1$, W_{ji}^{k-1} is the kernel from j^{th} neuron of $k - 1$ layer to the i^{th} neuron of k layer.

5.1.2 Pooling layers

After the convolutional layer, we normally use the pooling layer. This layer is used to reduce the size of the feature map through downsampling. The pooling layer preserves only effective features and avoids redundant feature sets. The aggregate operations (maximum, average, and summation) are the most commonly employed pooling layers. Equation (11) is used for computing the outcomes of the max-pooling layer [6].

$$P^{l(n,m)} = \max_{(j-1)W+1 \leq m} \left\{ \sigma^l(m, n) \right\} \tag{11}$$

where $P^{l(n, m)}$ defines the pooling layer P at l^{th} layer with dimension n rows and m columns, W denotes the convolution kernel space, and the activation function denoted by σ finds the maximum value in $(n \times m)$ dimension at l^{th} layer.

5.1.3 Activation functions

The activation function controls whether or not a neuron is activated. The most commonly utilized activation functions are ReLU (Rectified Linear Unit) and softmax. The ReLU activation function replaces the negative values with zeros and passes the non-negative values the same as the inputs. This activation function is followed by every convolutional layer. The ReLU activation function carries the non-linearity nature and has no back-propagation error. Equation (12) is used for computing the ReLU function.

$$\pi(x) = \max(0, x) \tag{12}$$

where π indicates the ReLU activation functions and (x) is network parameters.

In most neural networks, the softmax activation function is utilized at the classification layer. This calculates the probability distribution at the classification layer and then maps the

output values to the $[0, 1]$ range, and their total sum of probability values is equal to one. Equation (13) is used for computing the softmax activation function.

$$\sigma(x)_i = \frac{e^{x_i}}{\sum_{j=1}^k e^{x_j}}, j = 1, 2, 3, \dots, k \quad (13)$$

where σ indicates the softmax activation function, x is non-normalized parameters, and indexed value j for output unit for $1, 2, \dots, k$.

5.1.4 Fully connected layers

This is a feed-forward neural network in its most basic form. The final convolutional or pooling layer output is flattened and then given as input to the fully connected layer. The formula used by the fully connected layer is given in Equation (14) [6].

$$O_{output}^l = f(x^{l-1} \times w^l \times b^l) \quad (14)$$

where f denotes the activation function, x^{l-1} is input from the previous layer, w^l is the neural weight at l^{th} layer, and b bias at l^{th} layer.

5.1.5 Dropout layers

The dropout layers are mostly used for handling the overfitting problem. While training the neural networks, some neurons are trained in dependency mode and work similarly. This is beneficial to remove those neurons otherwise will generate overfitting issues. The dropout layer eliminates certain neurons from the network at random without affecting the classification performance.

5.1.6 Regression layers

The regression layer computes the Mean Squared Error (MSE) loss from the predicted to the actual class response. Equation (15) is used for the regression process [6].

$$MSE(L_r) = \frac{1}{N} \sum_{i=1}^{N_i} (O_i^L - t_i^p)^2 \quad (15)$$

where N_i is the number of class labels, t_i denotes the target vector, actual output vector is denoted by O_i , and input vector is denoted by p .

5.2 Recurrent Neural Networks (RNNs)

The RNN is the best neural network for dealing with continuous sequential data and embedded internal memory cells. This memory helps to carry the past information from layer $(l - 1)$ to layer (l) for making the process to the layer $(l + 1)$. The recurrence weights (W) in RNNs are changed using a feedback loop between the output and hidden layers. The recurrent edges exist connecting the output and hidden layer at t time step. Here, we assume that input x , hidden layer h , output layer o , target layer y , loss l , softmax activation function, discrete output form,

and negative log-likelihood loss. The node connected with the recurrent edge gets the value from the current data point $x^{(t)}$ and values of the previously hidden node $h^{(t-1)}$ at network state. The value of output $y^{(t)}$ is calculated at each time step t . Equations (16) and (17) are used for calculating the forward pass in RNNs at each time [30].

$$h^{(t)} = \sigma \left(W^{hx}x^{(t)} + W^{hh}h^{(t-1)} + b_h \right) \tag{16}$$

$$y^{(t)} = \text{softmax} \left(W^{yh}.h^{(t)} + b_y \right) \tag{17}$$

where, W^{hx} , W^{hh} , and W^{yh} are denoted as a convolutional weighted matrix between input & hidden layer, hidden layer & itself, and hidden & output layer at the adjacent time steps, respectively with bias parameter b_h and b_y . Figure 5 depicts the simple layered architecture of the recurrent network.

6 Proposed Deep-HAR model

The proposed Deep-HAR model is an ensemble DL method (1D-CNN + RNN) for recognizing the simple, complex, and heterogeneous activities. The basic idea behind the Deep-HAR is to encapsulate the beneficial characteristics of DL models into a single model. For instance, the convolutional layers are powerful to extract the appropriate features from raw sensor data. However, the CNN model has no concept of the memory cell and backward propagation. On the other hand, the RNNs are well suited for handling the time series data sample and memory cell that remembers the temporal dependencies.

In terms of different architecting layers, the ensemble Deep-HAR model is made up of many convolutional and recurrent networks. There are three ensembled convolutional (Conv Model 1, Conv Model 2, and Conv Model 3) and recurrent (Recur Model 1, Recur Model 2,

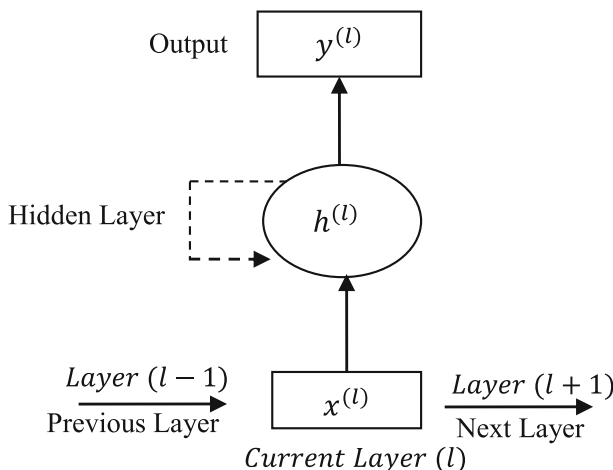


Fig. 5 Simple recurrent network with one input, hidden, and output layer

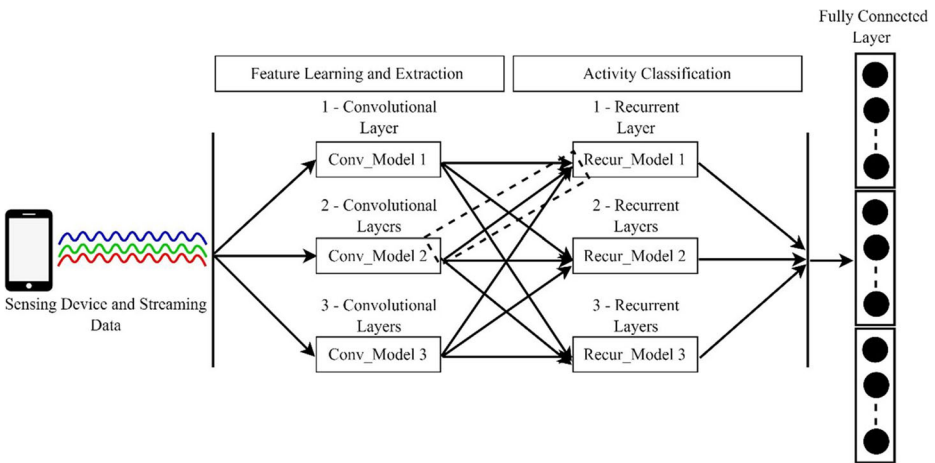


Fig. 6 Ensemble Deep-HAR model

and Recur Model 3) networks, as shown in Fig. 6. The diversity exists in the residency of architecting layers in ensembled methods. Hence, there are a total of nine combinations of possibilities ($3 Conv_{model} \times 3 Recur_{model}$) exists before finalizing the ensemble Deep-HAR model. By analyzing the architectural behavior as a possible combination, we found that the proposed model should be optimistic and lightweight. While ensembling the Deep-HAR model, the convolutional networks with two layers have been used for feature learning and extraction whereas for activity recognition, the recurrent networks with a single layer were used.

The architectural view of the proposed Deep-HAR model has depicted in Fig. 7. Two convolutional layers, one max-pooling layer, and dropout layers have been set up in the 1D-CNN model. The first Convo_Layer 1 learns the effective features and followed the

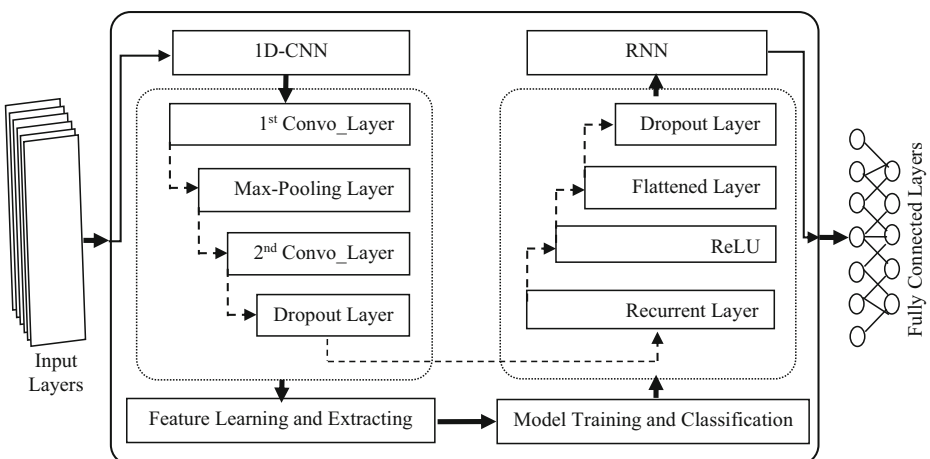


Fig. 7 Architectural diagram of proposed Deep-HAR model

Max_Pooling layer helps in feature dimension reduction. The Convo Layer 2 is then added to extract the effective feature sets from the decreased dimensions. Finally, a dropout layer is added to prevent overfitting. Furthermore, the outcomes of the 1D-CNN model were given as input into the next RNN model. A single recurrent, ReLU activation function, flattened, and dropout layer has been used in the RNN model. When the model is trained, the recurrent layer memorize the dependencies among the successive activities. Subsequently, the ReLU activation function ($\max(0, x)$ where x – input parameters) decides the activation of neurons. Next, the values are flattened into the 1D vector and given as input for the dropout layer. The dropout layer randomly removes the neurons proportional to the overfitting issue. Finally, the Deep-HAR model uses fully connected layer for classification.

6.1 Algorithms for data preprocessing and proposed model designing

The data preprocessing and splitting procedure are described in algorithm 1. This algorithm works by taking the time-series raw sensor triaxial dataset $D [(x_1, y_1, z_1), \dots, (x_n, y_n, z_n)]$ as input and returns the split samples corresponding to the specified train, validate, and test ratio. Our proposed model has an addition to quantitative data instead of string values. First of all, we need to check whether the activity label is already encoded in numerical form or not. Furthermore, we need to split the dataset D with the respect to the given train, validate, and test ratio. The train samples are used for model training whereas the validation dataset used for hyperparameter tuning which controls the behavior of our model. The performance of the proposed model is evaluated on the testing dataset to derive the experimental findings. The working procedure of our proposed Deep-HAR model is mentioned in algorithm 2. This algorithm receives the data samples: training (X_s), validating (Y_s), and testing (Z_s) and returns the gathered experimental result.

Algorithm 1 Data Preprocessing

Input: A time-series raw sensor S_n dataset $D [(x_1, y_1, z_1), \dots, (x_n, y_n, z_n)]$

Output: Data samples: *Training* (X_s), *validating* (Y_s), and *testing* (Z_s)

1. **Initialize:** $X_s = [empty]$, $Y_s = [empty]$, $Z_s = [empty]$, and $Start = 0$
 2. **While** $Start < \text{Length}(D)$
 3. **If** $\text{is_String}(D[\text{activity}]) == 1$, where 1: True, 0: False
 4. **then**
 5. $N = \text{Count}(\text{unique}(D[\text{activity}])),$ where N: number of activities
 6. Add column
 $D[\text{activity}_{id}] = \forall D[\text{activity}] \in \text{encode}\{D[\text{activity}] \leftarrow id(1 \dots N),$
 where $\text{encode}()$ function replaces each activity name with corresponding each
 unique Id.
 7. **End if**
 8. **End while**
 9. **Initialize Ratio:** train = 0.80, validate = 0.10, and test = 0.10
 10. $X_s, Z_s = \text{Train}_{\text{test_split}}(\text{test_size} = 1 - \text{train})$
 11. $Y_s, Z_s = \text{Train}_{\text{test_split}}(\text{test_size} = (\text{test} / (\text{test} + \text{validate})) \times 100)$
 12. **Return:** *Training* (X_s), *validating* (Y_s), and *testing* (Z_s)
-

Algorithm 2 Deep-HAR Model

Input: Data samples: *training* (X_s), *validating* (Y_s), and *testing* (Z_s)
Output: Experimental Result of Deep-HAR Model.

1. **Initialize the CNN and RNN networks, Start = 0**
2. **Input Samples: X_s, Y_s , and Z_s**
3. **While Start < Length (Train X_s)**
4. **Forward Networks Propagation (1D-CNN)**
 5. Calculate the 1D-CNN layered values:
 6. $\text{Convo_Layer} (L_1) \leftarrow O_i^k = b_i^k + \sum_{j=1}^{N_k-1} I_i^{k-1} \otimes W_{ji}^{k-1}$
 7. $\text{Max-Pooling} (P_1) \leftarrow P^{l(m,n)} = \max_{(j-1)W+1 \leq m} \{\sigma^{l(m,n)}\}$
 8. $\text{Convo_Layer} (L_2) \leftarrow O_i^k = b_i^k + \sum_{j=1}^{N_k-1} I_i^{k-1} \otimes W_{ji}^{k-1}$
 9. Layer. Dropout (0.25)
10. **Backward Networks Propagation (RNN)**
 11. Calculate the RNN gates value:
 12. $\text{Hidden State} (H_t) \leftarrow \sigma (W^{hh}x^{(t)} + W^{hh}h^{(t-1)} + b_h)$
 13. $\text{ReLU activation function } \pi (x) \leftarrow \max (0, x)$, where x : *networks parameters*
 14. Perform Layer. BatchNormalization() for smoothing and optimization the hidden layered performance.
 15. Then, the outcomes of Layer. Flattened () converted into single 1D vector data.
 16. Layer. Dropout (0.20)
 17. Final output: $\tilde{y}^{(t)} \leftarrow \text{softmax} (W^{yh} \cdot h^t + b_y)$
18. **End While**
19. **Model** \leftarrow Deep-HAR.Compile:
where LearningRate \leftarrow 0.0001 and *Loss* \leftarrow *Sparse Categorical Crossentropy*
20. **Model** = Deep-HAR.Train: (X_s, Y_s):
where Epochs \leftarrow 10 and *batch size* \leftarrow 128
21. **Model** = Deep-HAR.Predict (Z_s)
22. **Experimental Result:**
 - 22.1. Model.confusion_matrix ()
 - 22.2. Model.classificaiotn_report ()
23. **Return:** Experimental result.

We have used the forwarding propagation algorithm (1D-CNN) for feature extraction and the backward propagation algorithm (RNN) for model training and classification. In the forward propagation algorithm, the outcomes of the first convolutional layer, calculated using the given equations, are fetched to the max-pooling layer to learn the effective features and reproduce the convolutional dimension. Subsequently, the reproduced convolutional dimension is fetched to the second convolutional layer for extracting the suitable feature. For resolving the overfitting issue, the dropout layer is used. Then, the outcome of forwarding propagation networks is fetched to the backward propagation network (RNN). The RNN calculates the value for hidden states using the given expression and is followed by the ReLU activation function. Furthermore, batch normalization is used for optimizing the performance of hidden states and generates the 1D-vector using a flattened layer. Again, the dropout layer is used for the above-mentioned purpose. Finally, the output of the proposed model is determined to report the experimental outcomes.

6.2 Configuration of the Deep-HAR model

The importance of hyperparameters in influencing the behavior of the Deep-HAR model is highlighted in this section. First and foremost, we must comprehend the link between model performance and hyperparameters. When there is a performance disparity between training and testing error, hyperparameter optimization becomes crucial. The

primary goal behind the hyperparameters tuning is to enhance the model capacity for handling the complexity of tasks. After parameter tuning, the optimized values for the Deep-HAR model have shown in Table 4.

7 Experimental results and discussion

The activities recognition performance of the Deep-HAR model have been evaluated on simple (WISDM), complex (PAMAP2), and heterogeneous (KU-HAR) datasets. For assessment purpose, we employed the accuracy, recall, precision, and F1 score. The experimental dataset has been divided into three parts: 80% for training, 10% for validating, and 10% for evaluating the performance of the proposed model. The number of samples available in the experimental datasets are shown in Table 5.

The setup of the experimental environment has explained as follows. We have used the Google Colab for the implementation of the proposed model using the python code. The TensorFlow, Scikit-Learn, Keras, pandas, NumPy, and matplotlib packages have been imported to accomplish the task of data preprocessing, splitting, model architecting, training, validating, testing, and plotting the experimental outcomes. We shall examine the model authenticity, classification quality using the confusion matrix, experimental results, comparative study with recent papers, and architecting behavior in this section.

7.1 The authenticity of the Deep-HAR model

Overfitting and underfitting conditions were used to assess the legitimacy of the proposed model. Whenever the proposed model has suffered either an overfitting or underfitting circumstance, the classification performance is always skewed.

Table 4 Optimized parameters values of Deep-HAR model

Deep HAR Model	Tuning Parameters		Values
CNN Architecture	Convo_Layer 1	Input Shape	#
		Kernel Size	3
		Filters	512
		Padding	Same
	Max Pooling Layer		2
	Convo_Layer 2	Kernel Size	1
		Filters	256
RNN Architecture	Dropout Layer		00.25
	Recurrent Layer	Units	128
	Activation Function		ReLU
	Dropout Layer		00.20
	Dense		#
Model Training	Activation Function		Softmax
	Loss		Sparse Categorical Crossentropy
	Learning Rate		0.0001
	Optimizer		Adam
	Batch Size		128
	Epochs		10

Note: The Hash (#) shows that the initialized value varies according to given experimental datasets

Table 5 Data samples distribution of the experimental datasets

Experimental Dataset	Training (#)	Validating (#)	Testing (#)	Total Samples
WISDM	02,74,732	34,342	34,342	03,43,416
PAMAP2	15,54,297	1,94,288	1,94,288	19,42,873
KU-HAR	49,77,471	6,22,184	6,22,184	62,21,839

First of all, we need to authenticate the Deep-HAR model. With an epoch rate of 10, the proposed model was trained on the full training dataset, yielding results for training accuracy, validation accuracy, train loss, and validation loss, respectively. The training accuracy, validation accuracy, training loss and validation loss are shown in Fig. 8. The primary objective behind plotting these values against the epoch rates is to observe whether our proposed model has fallen in overfitting/underfitting conditions or not. Here, the accuracy and loss lines are much close to each other which implies that the proposed model has been designed and hyperparameters are tuned perfectly to meet the predefined goal. With the increasing rate of the epoch, the train and validation accuracy is covering the maximum defined score (1.0). Similarly, the train and validation loss turned out to be 0.0, which should be as low as much possible. From this, we conclude that our proposed model has been designed perfectly without any overfitting and underfitting conditions.

7.2 Confusion matrix of the Deep-HAR model

The classification results of the simple, complex, and heterogeneous activities using the confusion matrix has shown in Fig. 9 including the number of activities samples.

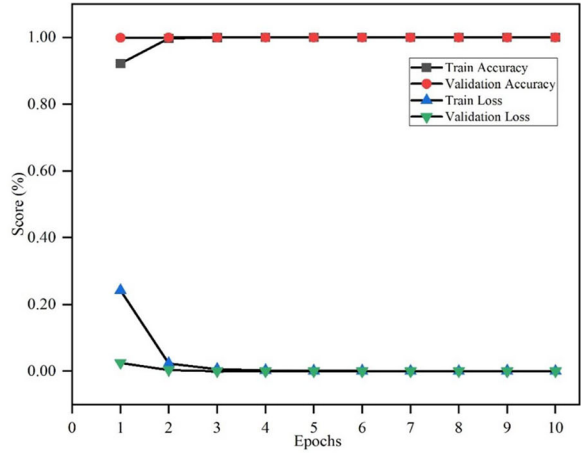
In the confusion matrix, each row corresponds to the actual class label and each column corresponds to the predicted class label. From the confusion matrix, it is clear that the proposed model performs well in activity recognition. Furthermore, the proposed Deep-HAR model may be accepted as a universal model for the recognition of simple, complex, and heterogeneous activities.

7.3 Experimental results of the Deep-HAR model

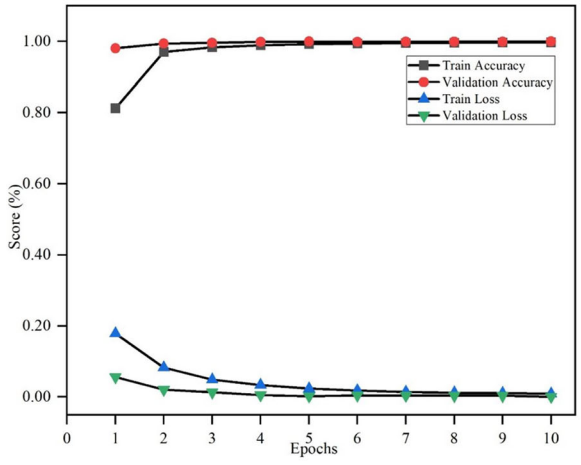
The total recognition rates of the Deep-HAR model for each type of activities are shown in Table 6. The accuracy, precision, recall, and F1-score of the Deep HAR model are high enough to recognize the simple, complex, and heterogeneous activities.

For simple activities, the proposed model achieved the recognition rate with an accuracy of 99.98%, precision of 90.57%, recall of 100.00%, and F1-score of 95.13%, respectively. A collection of temporal activities has often been found in each form of complex activities. The classification performance of the Deep-HAR model was 99.64% accurate, the precision of 91.86%, recall of 100.00%, and F1 score of 96.61%, respectively for the complex activities. Recognizing the heterogeneous activities was considerably more challenging. But, our proposed model, on the other hand, fearlessly acknowledged this forms of activities. The accuracy rate was 99.98%, the precision was 97.38%, the recall was 100.00%, and the F1 score was 98.96% for heterogeneous activities. Figure 10 illustrates the graphical depiction of the experimental findings.

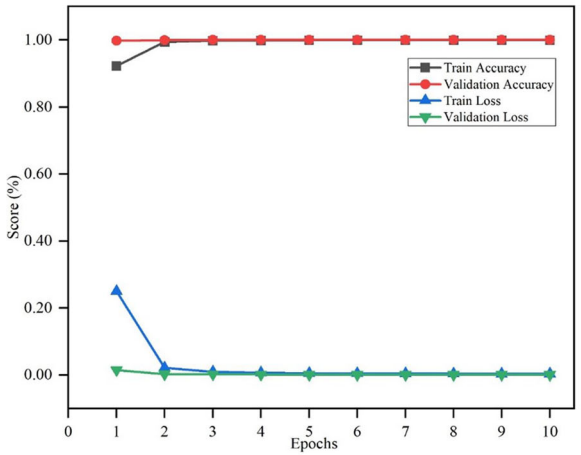
Fig. 8 Accuracy and loss score during the training and validating of the proposed model on (A) simple, (B) complex, and (C) heterogeneous activities



(A)



(B)



(C)

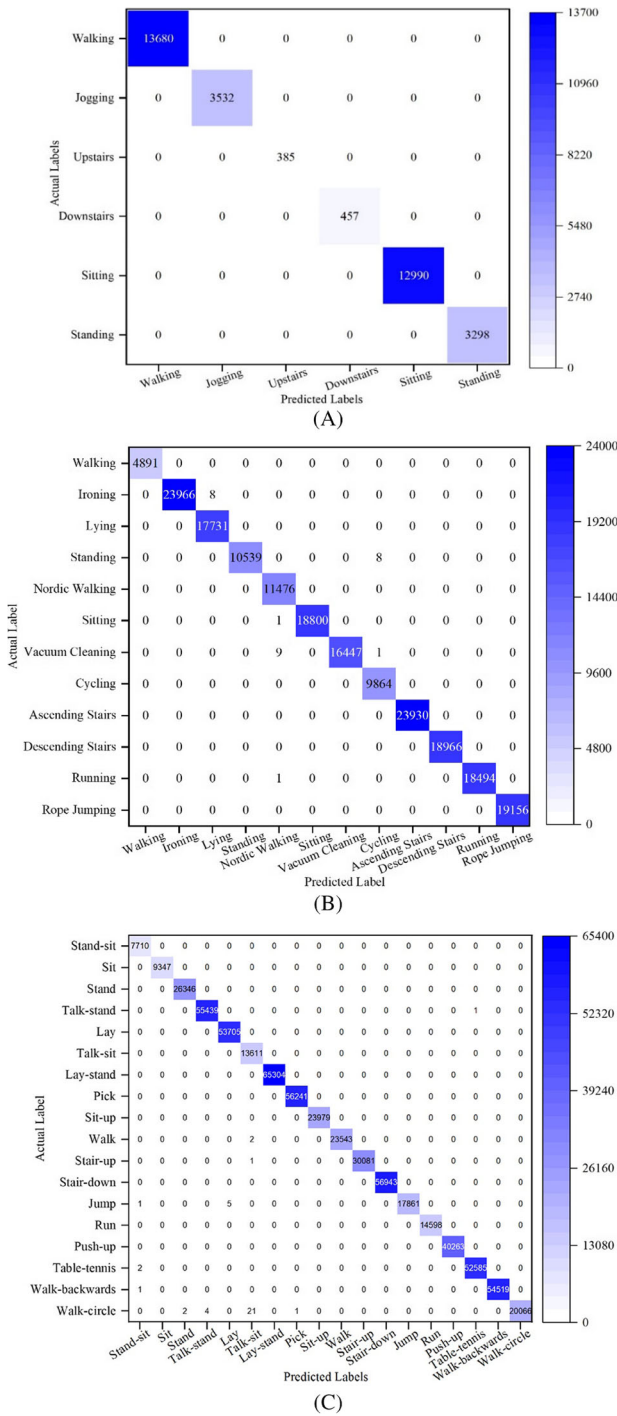


Fig. 9 Confusion matrix of (A) Simple, (B) Complex, and (C) Heterogeneous activities

Table 6 Recognition rates of the Deep-HAR model for each activity type

Deep-HAR model					
Activities Type	Dataset	Evaluation Matrix			
		Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Simple	WISDM	99.98	90.57	100.00	95.13
Complex	PAMAP2	99.64	91.86	100.00	96.61
Heterogeneous	KU-HAR	99.98	97.38	100.00	98.96

7.4 Comparison with recently published research works

The experimental result of recently published research works and our proposed model has comparatively shown in Table 7. The authors in [1, 5, 8, 12, 13, 15, 17, 22, 24, 33, 47] have applied their proposed model on the WISDM, PAMAP2, and KU-HAR datasets, respectively. Moreover, we have also used the same experimental datasets but in the context of different activity types.

In the study of [1, 12, 13, 22, 47], the authors have used various models namely UDR-RC, 1D-CNN, Att-based Residual Network, RNN-LSTM, and AFFNet on the WISDM dataset. The best recognition performance was obtained by the Att-based Residual Network with an accuracy of 98.85% and followed by UDR-RC with 97.50% accuracy. The authors in [5, 12, 15, 17, 24] have used the one-shot learning methods, DELAPAR, Att-based Residual Network, ResNet+HC, and float CNN on PAMAP2 experimental dataset. In [15], the DELAPAR

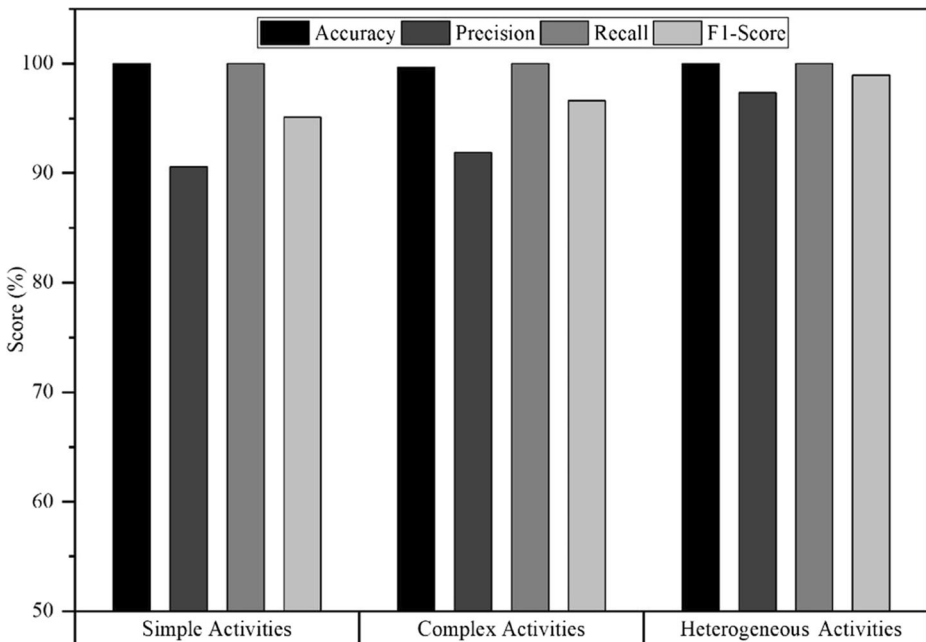
**Fig. 10** Graphical view of experimental results of the Deep-HAR model

Table 7 Comparative study on experimental results of our model with recently published research works

Activities Type	Dataset	Methods	Accuracy (%)	Year	Reference
Simple	WISDM	UDR-RC	97.50	2020	[22]
		1D-CNN	94.20	2018	[13]
		Att-based Residual Network	98.85	2021	[12]
		RNN-LSTM	95.78	2020	[1]
		AFFNet	94.60	2022	[47]
		Deep-HAR (Our Model)	99.98	–	–
Complex	PAMAP2	One-shot learning methods	84.41	2021	[24]
		DELAPAR	96.62	2020	[15]
		Att-based Residual Network	93.16	2021	[12]
		ResNet+HC	92.97	2022	[17]
		float CNN	85.23	2021	[5]
		Deep-HAR (Our Model)	99.64	–	–
Heterogeneous	KU-HAR	Random Forest	90.00	2020	[33]
		Adapted transformer model	99.20	2022	[8]
		Deep-HAR (Our Model)	99.98	–	–

achieved the highest accuracy rate of 96.62%, followed by the Att-based Residual Network [12] with an accuracy rate of 96.62%. On the KU-HAR dataset, the authors have applied the RF classifier [33] and Adapted transformer model [8] that achieved the accuracy rate of 90.00% and 99.20%, respectively. Finally, our proposed Deep-HAR model achieved the best prediction performance on WISDM, PAMAP2, and KU-HAR datasets with an accuracy rate of 99.98%, 99.64%, and 99.98%, respectively. Moreover, we can state that our proposed model can be globally acceptable and most recommended for recognizing the simple, complex, and heterogeneous type activities.

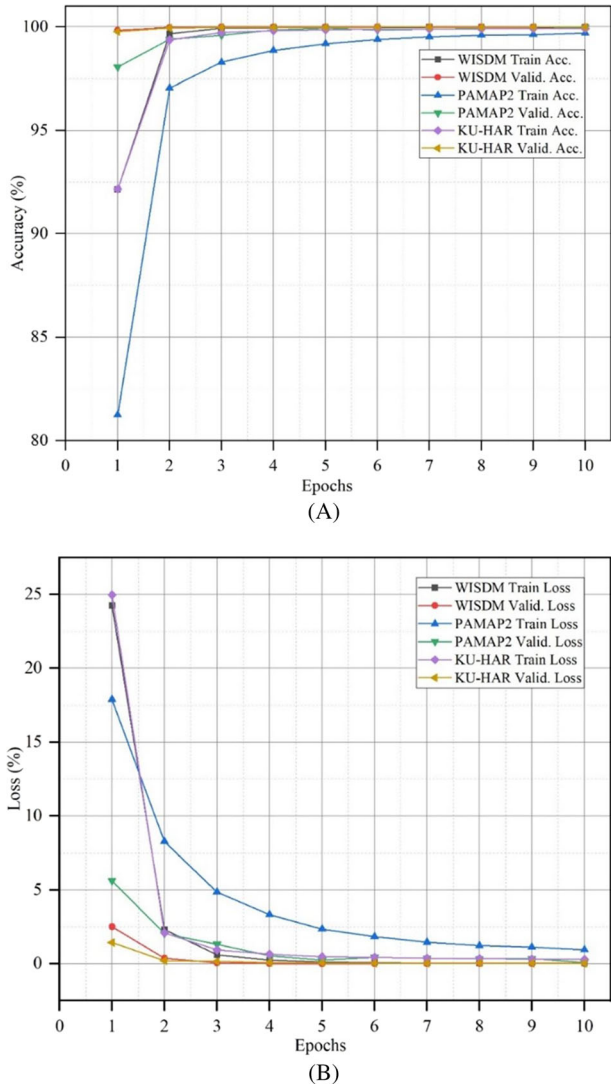
7.5 Comparison with architectural behavior of different past models

This section presents the comparative study on the architectural behavior of previous models to our model.

The UDR-RC [22] model has mainly focused on optimizing the data during pre-processing, minimizing the computational time, and improving the recognition rates. This approach has followed the fixed-size window strategy. The Reconstructed Coder (RC) has used the concept of encoder-decoder for minimizing the reconstruction errors. Hence, our proposed model has achieved higher robustness against the class imbalancing issue (solved using F1-score) and conflicting behavior attributes in HAR datasets such as the number of sensors, subjects, activities, sampling rates, sensing devices, and more. So, we need not require to make more emphasis during the pre-processing phase.

The activity recognition model has been elaborated using distance matrices of recurrence plot with CNN in [13]. Recurrent plotting is an visualizing method to represent the recurrent state of the dynamic system. Furthermore, this approach converted the raw acceleration data into an image formation of a recurrent state, which was then used to train the CNN model. The CNN model began with the input layer, which was followed by two convolutional layers, single max-pooling, and dropout layer. Before, data sending for flattening and fully connected layer, again passed to the two successive convolutional layers, single max-pooling, and dropout layer. Hence, as compared to this model, our proposed model has been designed more lightweight and interactive as we have directly used the sensory data in the model.

Fig. 11 Overall (A) Accuracy (%) and (B) Loss (%) of the proposed model



In [24], the authors have used the one-shot learning technique. This technique needs a strong and high-level feature extraction technique for better recognition rates. The one-shot learning technique has achieved considerable performance in recognition of the similar type of activities where few instances of activity classes are available. This approach faced difficulties while dealing with complex activities rather than similar type activity. However, our proposed model has used the convolutional layers for extracting the features and need not require a strong feature extraction technique. The proposed DeepHAR model can perform more reliably in recognizing simple, complex, and heterogeneous activities.

In [15], the authors have proposed the three window-based modules for activity recognition and post-processing technique. The first module used the overlapped window of data segments and extract the feature in the frequency domain. In the second module, the deep learning model detects the activity in each window. The third module expands the window-level choice over

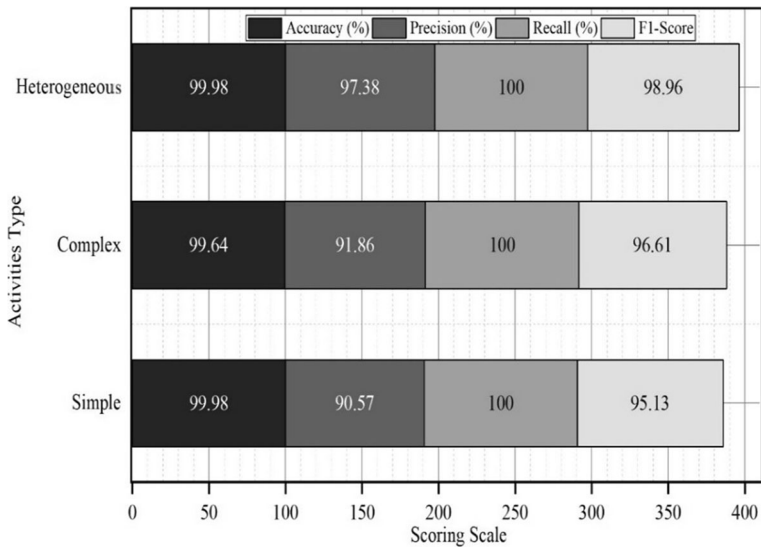


Fig. 12 Overall accuracy, precision, recall, and F1-score of the proposed model

longer periods, resulting in considerable performance gains. Further, the post-processing techniques, median filter and HMMs, are used for improving the activity recognition rates. However, the proposed DeepHAR model has used the complete sensory data to learn and extract the efficient features. Moreover, the proposed model declined the need for a post-processing technique for improving the performance of activity recognition rates.

In [33], the authors have used the classical supervised learning algorithm i.e. RF for recognizing the heterogeneous type activities. The RF is a widely used ensemble learning algorithm that produces the outcomes of the best decision tree, selected from various decision trees. The activity recognition performance of classical learning algorithms is completely dependent on the quality of feature engineering. This limitation can be overcome by using our proposed model that used automatic feature extraction techniques.

8 Conclusion and future works

In this paper, we have proposed the Deep-HAR model as a one-stop solution for recognizing the simple, complex, and heterogeneous human activities. The proposed model extracts the effective features set from raw sensor data and then learns the activity patterns using convolutional and recurrent layers for recognizing the simple, complex, and heterogeneous activities. The classification performance of the proposed model has been evaluated experimentally on three different representative datasets i.e. WISDM, PAMAP2, and KU-HAR. Figure 11 shows the accuracy and loss value of the model while training and testing with a 10-epoch value. From the figure, it is clear that the train and test accuracy are close to each other and the train and test loss are also to each other. This signifies that the proposed model has good generalization and neither fell in overfitting nor underfitting condition.

For evaluating the performance of the Deep-HAR model, we have used the accuracy, precision, recall, and F1-Score. The summarized experimental results of the proposed model is given in Fig. 12 for different activity types. The proposed model holds an average accuracy of

99.86%, a precision of 93.27%, a recall of 100.00%, and an F1 score of 96.90%, respectively for all activity types. The experimental results prove that the Deep-HAR model works well as a single platform for recognizing the simple, complex, and heterogeneous type activities instead of different models for recognizing each activity type, separately.

Furthermore, the proposed solution can easily be extended to recognize activities specific to a particular domain such as gym, yoga, and sports. To train deep ensemble models for recognizing more specialized domain-specific activities, large amounts of training data must be collected. The issue of scarcity of labeled quality datasets can be addressed by exploiting the concept of a transfer learning approach to cope with more complex, specialized tasks.

Acknowledgments The authors would like to express their gratitude to the anonymous reviewers for their insightful comments, which greatly improved the quality of the paper. The authors are grateful for the support provided by the UGC, New Delhi through the JRF and Banaras Hindu University through the Institute of Eminence (IoE) Seed Grant.

Code availability The source code used for implementing this research is available on a request basis to the corresponding author, Prabhat Kumar (prabhat.kumar13@bhu.ac.in).

Author contributions Prabhat Kumar designed the architecture, and experiments analyzed the data and wrote the paper whereas S. Suresh guided the paper writing and reviewed the paper. All authors read and approved the final manuscript.

Declarations

Conflict of interest The authors state that they have no known conflicting financial interests or personal connections that could have appeared to affect the work presented in this publication.

References

1. Agarwal P, Alam M (2020) A lightweight deep learning model for human activity recognition on edge devices. *Procedia Comput Sci* 167:2364–2373. <https://doi.org/10.1016/j.procs.2020.03.289>
2. Bojan Kolosnjaji CE (2015) Neural network-based user-independent physical activity recognition for Mobile devices. 378–386 <https://doi.org/10.1007/978-3-319-24834-9>
3. Cao Y, Geddes TA, Yang JYH, Yang P (2020) Ensemble deep learning in bioinformatics. *Nat Mach Intell* 2:500–508. <https://doi.org/10.1038/s42256-020-0217-y>
4. Chen WH, Betancourt Baca CA, Tou CH (2017) LSTM-RNNs combined with scene information for human activity recognition. 2017 IEEE 19th International Conference on e-Health Networking, Applications and Services, Healthcom 2017 2017-Decem:1–6. <https://doi.org/10.1109/HealthCom.2017.8210846>
5. de Vita A, Pau D, di Benedetto L, Licciardo GD (2021) Highly-accurate binary tiny neural network for low-power human activity recognition. *Microprocess Microsyst* 87:104371. <https://doi.org/10.1016/j.micpro.2021.104371>
6. Dhillon A, Verma GK (2020) Convolutional neural network: a review of models, methodologies and applications to object detection. *Progress Artif Intell* 9:85–112. <https://doi.org/10.1007/s13748-019-00203-0>
7. Dhiman C, Vishwakarma DK (2019) A review of state-of-the-art techniques for abnormal human activity recognition. *Eng Appl Artif Intell* 77:21–45. <https://doi.org/10.1016/j.engappai.2018.08.014>
8. Dirgová Luptáková I, Kubovčík M, Pospíchal J (2022) Wearable sensor-based human activity recognition with transformer model. *Sensors* 22:1911. <https://doi.org/10.3390/s22051911>
9. Elshafei M, Shihab E (2021) Towards detecting biceps muscle fatigue in gym activity using wearables. *Sensors (Switzerland)* 21:1–18. <https://doi.org/10.3390/s21030759>
10. Fridriksdottir E, Bonomi AG (2020) Accelerometer-based human activity recognition for patient monitoring using a deep neural network. *Sensors (Switzerland)* 20:1–13. <https://doi.org/10.3390/s20226424>

11. Ganaie MA, Hu M, Malik AK et al (2021) Ensemble deep learning: A review. *Eng Appl Artif Intell* 115. <https://doi.org/10.1016/j.engappai.2022.105151>
12. Gao W, Zhang L, Teng Q et al (2021) DanHAR: Dual Attention Network for multimodal human activity recognition using wearable sensors. *Appl Soft Comput*:111. <https://doi.org/10.1016/j.asoc.2021.107728>
13. Garcia-Ceja E, Uddin MZ, Torresen J (2018) Classification of recurrence plots' distance matrices with a convolutional neural network for activity recognition. *Procedia Comput Sci* 130:157–163. <https://doi.org/10.1016/j.procs.2018.04.025>
14. Gardas BB, Mangla SK, Raut RD, Luthra S (2019) Green talent management to unlock sustainability in the oil and gas sector. *J Clean Prod* 229:850–862. <https://doi.org/10.1016/j.jclepro.2019.05.018>
15. Gil-Martin M, San-Segundo R, Fernández-Martínez F, Ferreiros-López J (2020) Improving physical activity recognition using a new deep learning architecture and post-processing techniques. *Eng Appl Artif Intell* 92: 103679. <https://doi.org/10.1016/j.engappai.2020.103679>
16. Gu T, Wu Z, Tao X et al (2009) ePSICAR: an emerging patterns based approach to sequential, interleaved and concurrent activity recognition. 7th annual IEEE international conference on pervasive computing and communications. *PerCom*. <https://doi.org/10.1109/PERCOM.2009.4912776>
17. Han C, Zhang L, Tang Y, He J (2022) Human activity recognition using wearable sensors by heterogeneous convolutional neural networks. *Expert Syst Appl* 198:116764. <https://doi.org/10.1016/j.eswa.2022.116764>
18. Hoang ML, Carratù M, Paciello V, Pietrosanto A (2021) Body temperature—indoor condition monitor and activity recognition by mems accelerometer based on IoT-alert system for people in quarantine due to COVID-19. *Sensors* 21:2313. <https://doi.org/10.3390/s21072313>
19. Huh JH, Seo YS (2019) Understanding edge computing: engineering evolution with artificial intelligence. *IEEE Access* 7:164229–164245. <https://doi.org/10.1109/ACCESS.2019.2945338>
20. Inoue M, Inoue S, Nishida T (2018) Deep recurrent neural network for mobile human activity recognition with high throughput. *Artif Life Robotics* 23:173–185. <https://doi.org/10.1007/s10015-017-0422-x>
21. Ishii S, Yokokubo A, Luimula M, Lopez G (2021) Exersense: physical exercise recognition and counting algorithm from wearables robust to positioning. *Sensors (Switzerland)* 21:1–16. <https://doi.org/10.3390/s21010091>
22. Janarthanan R, Doss S, Baskar S (2020) Optimized unsupervised deep learning assisted reconstructed coder in the on-nodule wearable sensor for human activity recognition. *Measurement: J Int Measurement Confederation* 164:108050. <https://doi.org/10.1016/j.measurement.2020.108050>
23. Jethanandani M, Sharma A, Perumal T, Chang J-R (2020) Multi-label classification based ensemble learning for human activity recognition in smart home. *Int Things* 12:100324. <https://doi.org/10.1016/j.iot.2020.100324>
24. Kasnesis P, Chatzigeorgiou C, Patrikakis CZ, Rangoussi M (2021) Modality-wise relational reasoning for one-shot sensor-based activity recognition. *Pattern Recogn Lett* 146:90–99. <https://doi.org/10.1016/j.patrec.2021.03.003>
25. Kim Y, Toomajian B (2016) Hand gesture recognition using micro-Doppler signatures with convolutional neural network. *IEEE Access* 4:7125–7130. <https://doi.org/10.1109/ACCESS.2016.2617282>
26. Kim E, Helal S, Cook D (2010) Human activity recognition and pattern discovery. *IEEE Pervasive Comput* 9:48–53. <https://doi.org/10.1109/MPRV.2010.7>
27. Kiranyaz S, Avci O, Abdeljaber O, Inman DJ (2021) 1D convolutional neural networks and applications: a survey. *Mech Syst Signal Process* 151:107398. <https://doi.org/10.1016/j.ymssp.2020.107398>
28. Kwapisz JR, Weiss GM, Moore SA (2011) Activity recognition using cell phone accelerometers. *ACM SIGKDD Explorations Newsletter* 12:74–82. <https://doi.org/10.1145/1964897.1964918>
29. Lee H, Park SH, Yoo JH, Huh JH (2020) Face recognition at a distance for a stand-alone access control system. *Sensors (Switzerland)* 20. <https://doi.org/10.3390/s20030785>
30. Lipton ZC, Berkowitz J, Elkan C (2015) A critical review of recurrent neural networks for sequence learning. <https://doi.org/10.48550/arxiv.1506.00019>
31. Liu L, Peng Y, Wang S, Huang Z (2016) Complex activity recognition using time series pattern dictionary learned from ubiquitous sensors. *Inf Sci* 340–341:41–57. <https://doi.org/10.1016/j.ins.2016.01.020>
32. Morales J, Akopian D (2017) Physical activity recognition by smartphones, a survey. *BiocyberneticsBiomed Eng* 37:388–400. <https://doi.org/10.1016/j.bbe.2017.04.004>
33. Nahid A-A, Sikder N, Rafi I (2020) KU-HAR: an open dataset for human activity recognition. *Pattern Recogn Lett* 3:46–54. <https://doi.org/10.1016/j.patrec.2021.02.024>
34. Park SW, Ko JS, Huh JH, Kim JC (2021) Review on generative adversarial networks: focusing on computer vision and its applications. *Electronics* 10(10):1216. <https://doi.org/10.3390/electronics10101216>
35. Połap D, Woźniak M, Wei W, Damaševičius R (2018) Multi-threaded learning control mechanism for neural networks. *Futur Gener Comput Syst* 87:16–34. <https://doi.org/10.1016/j.future.2018.04.050>

36. Raut RD, Mangla SK, Narwane VS, Narkhede BE (2019) Linking big data analytics and operational sustainability practices for sustainable business management. *J Clean Prod* 224:10–24. <https://doi.org/10.1016/j.jclepro.2019.03.181>
37. Reiss A, Stricker D (2012) Introducing a new benchmarked dataset for activity monitoring. Proceedings - international symposium on wearable computers, ISWC 108–109. <https://doi.org/10.1109/ISWC.2012.13>
38. Shrestha A, Mahmood A (2019) Review of deep learning algorithms and architectures. *IEEE Access* 7: 53040–53065. <https://doi.org/10.1109/ACCESS.2019.2912200>
39. Sindi H, Nour M, Rawa M, Polat K (2021) Random fully connected layered 1D CNN for solving the Z-bus loss allocation problem. *Measurement: J Int Measurement Conf* 171:1–8. <https://doi.org/10.1016/j.measurement.2020.108794>
40. Tao M, Li X, Wei W, Yuan H (2021) Jointly optimization for activity recognition in secure IoT-enabled elderly care applications. *Appl Soft Comput* 99:106788. <https://doi.org/10.1016/j.asoc.2020.106788>
41. Tarafdar P, Bose I (2021) Recognition of human activities for wellness management using a smartphone and a smartwatch: a boosting approach. *Decis Support Syst* 140:113426. <https://doi.org/10.1016/j.dss.2020.113426>
42. Thapa K, Abdullah Al ZM, Lamichhane B, Yang SH (2020) A deep machine learning method for concurrent and interleaved human activity recognition. *Sensors (Switzerland)* 20:1–20. <https://doi.org/10.3390/s20205770>
43. Tran DN, Phan DD (2016) Human activities recognition in android smartphone using support vector machine. In: Proceedings - international conference on intelligent systems, Modelling and Simulation, ISMS. IEEE Computer Society: 64–68. <https://doi.org/10.1109/ISMS.2016.51>
44. Ullah A, Muhammad K, Ding W, Baik SW (2021) Efficient activity recognition using lightweight CNN and DS-GRU network for surveillance applications. *Appl Soft Comput* 103:107102. <https://doi.org/10.1016/j.asoc.2021.107102>
45. Wang Y, Cang S, Yu H (2019) A survey on wearable sensor modality centred human activity recognition in health care. *Expert Syst Appl* 137:167–190. <https://doi.org/10.1016/j.eswa.2019.04.057>
46. Wang J, Chen Y, Hao S, Hu L (2019) Deep learning for sensor-based activity recognition: a survey. *Pattern Recogn Lett* 119:3–11. <https://doi.org/10.1016/j.patrec.2018.02.010>
47. Wang T, Liu Z, Zhang T et al (2022) Adaptive feature fusion for time series classification Knowledge-Based Syst:243. <https://doi.org/10.1016/j.knosys.2022.108459>
48. Weiss GM, Timko JL, Gallagher CM et al (2016) Smartwatch-based activity recognition: a machine learning approach. 3rd IEEE EMBS international conference on biomedical and health informatics. BHI 2016:426–429. <https://doi.org/10.1109/BHI.2016.7455925>
49. Weiss GM, Yoneda K, Hayajneh T (2019) Smartphone and smartwatch-based biometrics using activities of daily living. *IEEE Access* 7:133190–133202. <https://doi.org/10.1109/ACCESS.2019.2940729>
50. Woźniak M, Połap D, Capizzi G, Frankiewicz K (2018) Small lung nodules detection based on local variance analysis and probabilistic neural network. *Comput Methods Prog Biomed* 161:173–180. <https://doi.org/10.1016/j.cmpb.2018.04.025>
51. Xiao Y, Wu J, Lin Z, Zhao X (2018) A deep learning-based multi-model ensemble method for cancer prediction. *Comput Methods Prog Biomed* 153:1–9. <https://doi.org/10.1016/j.cmpb.2017.09.005>
52. Ye J, Stevenson G, Dobson S (2015) KCAR: a knowledge-driven approach for concurrent activity recognition. *Pervasive Mobile Comput* 19:47–70. <https://doi.org/10.1016/j.pmcj.2014.02.003>
53. Zhang W, Zhao X, Li Z (2019) A comprehensive study of smartphone-based indoor activity recognition via Xgboost. *IEEE Access* 7:80027–80042. <https://doi.org/10.1109/ACCESS.2019.2922974>
54. Zhou T, Lu H, Yang Z, Dong Y (2021) The ensemble deep learning model for novel COVID-19 on CT images. *Appl Soft Comput* 98. <https://doi.org/10.1016/j.asoc.2020.106885>

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

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.