Neural Computing and Applications (2022) 34:5993–6010
https://doi.org/10.1007/s00521-021-06787-w

ORIGINAL ARTICLE

Utilizing deep learning models in CSI-based human activity recognition

Eman Shalaby[1] · Nada ElShennawy[1] · Amany Sarhan[1]

Received: 27 April 2021 / Accepted: 22 November 2021 / Published online: 7 January 2022

Abstract

In recent years, channel state information (CSI) in WiFi 802.11n has been increasingly used to collect data pertaining to human activity. Such raw data are then used to enhance human activity recognition. Activities such as lying down, falling, walking, running, sitting down, and standing up can now be detected with the use of information collected through CSI. Human activity recognition has a multitude of applications, such as home monitoring of patients. Four deep learning models are presented in this paper, namely: a convolution neural network (CNN) with a Gated Recurrent Unit (GRU); a CNN with a GRU and attention; a CNN with a GRU and a second CNN, and a CNN with Long Short-Term Memory (LSTM) and a second CNN. Those models were trained to perform Human Activity Recognition (HAR) using CSI amplitude data collected by a CSI tool. Experiments conducted to test the efficacy of these models showed superior results compared with other recent approaches. This enhanced performance of our models may be attributable the ability of our models to make full use of available data and to extract all data features, including high dimensionality and time sequence. The highest average recognition accuracy reached by the proposed models was achieved by the CNN-GRU, and the CNN-GRU with attention models, standing at 99.31% and 99.16%, respectively. In addition, the performance of the models was evaluated for unseen CSI data by training our models using a random split-of-dataset method (70% training and 30% testing). Our models achieved impressive results with accuracies reaching 100% for nearly all activities. For the lying down activity, accuracy obtained from the CNN-GRU model stood at 99.46%; slightly higher than the 99.05% achieved by the CNN-GRU with attention model. This confirmed the robustness of our models against environmental changes.

Keywords Channel state information · Human activity recognition · Convolution neural network · Gated recurrent unit

1 Introduction

Research into Human Activity Recognition (HAR) has recently focused on utilizing this technology through various real-time applications, including smart homes, context awareness, as well as various military and healthcare related applications. Remote healthcare has witnessed significant advances as the result of the need to adapt to restrictions imposed by the COVID-19. A trend of e-health has arisen, where patients are monitored through internet-connected facilities. A multitude of computerized applications have thus been developed, their performance being measured in terms of efficiency, safety, response time and cost. E-health applications allow remotely connected devices to exchange information with a controlling device, report the current status of patients, alert the doctors or nurses of the occurrence of an abrupt change in the health of patients, and even make intelligent decisions without human intervention. HAR has become an essential part of care for elderly patients who decide (or whose family decides) they stay home instead of at a hospital or nursing home despite diminishing cognitive abilities. HAR includes "fall detection" approaches [1], which function to alert care providers to the need for intervention if an elderly person falls. Other applications, including surveillance and security applications, can also benefit from advances in HAR systems [2].

Several techniques have been developed to advance HAR. To detect a fall or other movement, a person may be

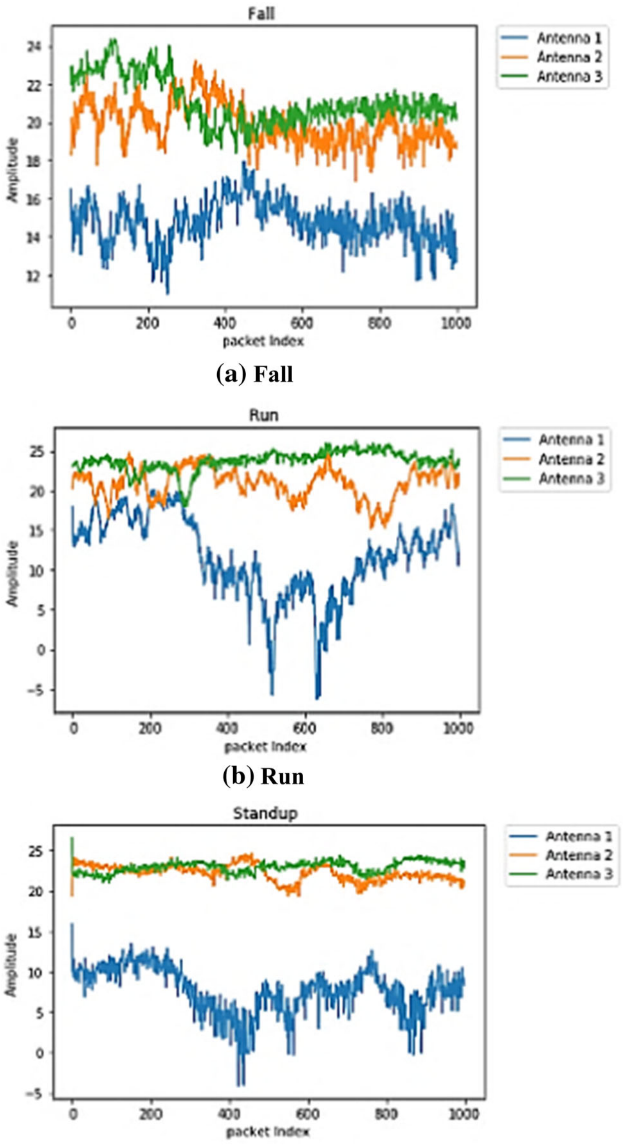
✉ Eman Shalaby
eman.shalaby@f-eng.tanta.edu.eg

[1] Computers and Control Engineering Department, Faculty of Engineering, Tanta University, Tanta, Egypt

monitored using wearable sensors [3] or other surveillance devices, including radar devices [4], non-wearable sensors and/or cameras. Cameras have the disadvantage of needing good lighting conditions and angle sets. While smartphones are now equipped with sensors, monitoring affects battery life, and the phone may potentially be lost. Moreover, it is difficult for some people to use wearable sensors because they are uncomfortable to them. Wi-Fi devices have proven to be the most suitable detection devices. Wi-Fi is currently available at almost all universities, offices, malls, and private residences, making Wi-Fi devices the best choice for activity detection. Wi-Fi signals have the advantage of breaking through many barriers with different strengths, negating the need for the person being monitored to be in a line-of-sight position or to wear any particular type of device.

A device-free detection of human activity is based on the monitoring and analyzing of changes in RSSI patterns. Wi-Fi signals emitted between a transmitter and a receiver are altered by human action and movement. Wireless signals may reach the receiver or may be reflected due to the existence of an obstacle. Captured information can be measured in terms of amplitude, phase, and signal-to-noise ratio (SNR). Two metrics could be potentially used to describe a Wi-Fi signal; received signal strength (RSS) and channel state information (CSI). A large number of researchers have found RSS not to be ideal for the description of characteristics of Wi-Fi signals [5]. RSS is highly sensitive to minute changes in the environment, with a large amount of "noise" and, as a result, false "small events" detected.

In recent years, CSI has been increasingly used for the purposes of describing properties of a signal that propagates between receivers [6] and transmitters. Subcarriers provide additional information, including phase and amplitude. With CSI, we can recognize any changes causing signal reflection and multipath propagation, such as movements of the human body. In our work, a public dataset collected using the Linux CSI 802.11n tool [6] is used; which is ideal for the task of describing Wi-Fi signals. Actions that occur between a Wi-Fi transmitter and receiver affect the characteristics of the Wi-Fi signal, and in turn, the channel displays different amplitudes and phases with each action. Examples of change in CSI amplitude for three person's activities: fall, run, and stand up between three Wi-Fi transmitters and receivers are shown in Fig. 1. As the Figure indicates, CSI amplitude remains almost the same when a person is stationary. Then, when the person performs an action such as running or falling, the CSI amplitude changes; the magnitude of the change being different with each action. Wi-Fi devices covering specific areas could thus be used to recognize human activity occurring within that area [11]. Nowadays, Wi-Fi devices are widely available for indoor usage at affordable prices.



(a) Fall

(b) Run

(c) Stand up

Fig. 1 Examples of CSI Amplitude Variations due to Human Activities: **a** Fall, **b** Run, and **c** Stand up

Some researchers reported using traditional machine learning (ML) techniques such as the Random Forest (RF) [7], the Hidden Markov Model (HMM) [8], the Support Vector Machine (SVM) [9], and the K-Nearest Neighbor (KNN) [10] for the purposed of achieving HAR.

Those methods are used for classification and prediction after the feature extracting phase. However, to obtain good classification, valuable features would have been carefully selected. This is not an easy step, and, in many cases, optimization approaches were required in order to select the features.

Deep Learning (DL) is a fast-growing trend in the science of ML that combines feature extraction and classification within the same phase, making it more suitable to a large number applications. DL is simply a function that trains a computer to imitate how the human brain is believed to process data and solve complex problems. DL employs multiple layers of artificial neural networks that enable learning. It has the advantage of being able to overcome manual feature extraction, of being better suited for use when there are large numbers of samples in a dataset, and of running faster a GPU is being used. DL models such as the convolution neural network (CNN), Long Short-Term Memory (LSTM), and Recurrent Neural Network (RNN) are used in many fields. A CNN is a neural network that uses filters that are convolved to overcome high dimensionality and share weights; RNN is a class of neural networks that uses previous outputs as inputs in the current step while having hidden states. RNN is used to solve sequence problems and process time series.

Several DL approaches have recently been incorporated into CSI-signal based HAR. Examples of such approaches include the LSTM approach, in which learning depends only on the current state of CSI [11], and the Attention Based Bidirectional Long Short-Term Memory (ABLSTM) approach, which is a combination of an attention model designed to assign larger weights and time steps, increasing human activity recognition performance, and a Bidirectional Long Short-Term Memory (BLSTM) [12]. Accuracy of these approaches could be further improved by the use of more complicated DL approaches and models in real-time applications.

Consequently, in order to achieve higher levels of activity recognition accuracy, this paper proposes a framework for feature extraction of human activity that would employ Wi-Fi CSI automatic extraction rather than manual extraction (seen in traditional ML). The main contributions of this paper can be summarized as follows:

- Designing and testing four HAR models developed as a combination of a CNN with either a Gated Recurrent Unit (GRU) or LSTM, and which employ CSI signals. CNN has the advantage of overcoming data with high dimensionality, while GRU and LSTM are better at processing time sequence data. Output from such combinations are then fed to an activation function to produce the data to be classified as the actual class. In the proposed framework, the first model is a CNN-GRU, the second a CNN-GRU followed by an attention layer, the third a CNN-GRU-CNN that shuffles CNN and GRU layers, and the fourth, a CNN-LSTM-CNN, in which the GRU is replaced by an LSTM layer.
- The proposed models were trained using both the data split method and the k-fold method. Various performance metrics, including accuracy, precision, recall, and area under the curve (AUC) were measured in order to yield an actual response to the unseen data.

The remainder of the paper is organized as follows: first, we discuss related work previously published in the literature in Sect. 2. Then, CNN and RNN models are discussed in Sect. 3. Next, an overview of the HAR system and a discussion of the proposed DL models are given in Sect. 4. Section 5 provides the results of testing our proposed models. Finally, we detail our conclusions in Sect. 6.

2 Related work

In this section, an overview of the problem is given, the technology used in HAR systems is clarified, and the basic ideas behind DL approaches are explained. Finally, a summary of previous studies comparable to the present work is given.

2.1 Human activity recognition

Currently, a large number of applications require the attainment of accurate information about human activities, which is a pervasive computing task. In recent years, the field of HAR has become an innovative field of study. HAR has found a multitude of applications, including healthcare, context awareness, national defense, and security. HAR aims to detect human motion and determine human actions, thus providing valuable information about the person being monitored. In indoor settings, movements being monitored may include running, jumping, and walking. First introduced in the 1990’s, HAR has since then gradually grew as a point of interest of both researchers and relevant vendors [13].

Generally speaking, HAR has been managed through two approaches: the use of external or wearable sensors. In the external-sensors approach [14], devices such as cameras and Wi-Fi devices are set up at specific sites within the area being monitored. In the wearable-sensors approach [3], the person being monitored is required to either constantly hold a device containing multiple sensors (such as smartphone) or be connected to the sensor itself. Invariably, each of the two approaches may be more suited for use in dissimilar applications and settings. Wearable devices require the approval of the person to be monitored (if they are able to make independent decisions) and are susceptible in many cases to damage during the person’s movements.

On the other hand, external approach devices are more stable, are usually hidden from those being monitored and thus pose less of a burden on them. Cameras, however, have the disadvantage of requiring the person being monitored to stay within their defined lines of sight. Cameras also require complex video processing techniques, and noise can affect accuracy [15]. Cellphones can run out of power, and then they become of no use. These concerns regarding cameras and cellphones led to the adaption of Wi-Fi approaches in HAR. RSS is the most commonly used signal in Wi-Fi indoor localization [16]. While it can be used for HAR, noise affects its performance and makes it unstable. Currently, CSI signals represent a stable alternative to RSS [6].

HAR starts by collecting informative data regarding the person being monitored. Based on the type of collected data, a preprocessing step may be required to put the data into a suitable form for the next step. Features of the various movements are extracted and fed to the classification step where the type of activity is recognized. Several methods have been used for classification such as: RF [7], HMM [8], SVM [9] and KNN [10] CNN [17], RNN [18] including LSTM [19] and GRU [20], Auto-Encoder (AE) [21], Boltzmann machine (BM) [22], Restricted Boltzmann Machine (RBM) [23], Deep Belief Network (DBN) [24], Generative Adversarial Network (GAN) [25], and Deep Reinforcement Learning (DRL) [26].

2.2 Previous work pertaining to HAR systems

HAR systems involve identifying specific predefined actions. Several systems have been proposed in the literature. To narrow the scope of this review to the most relevant previous studies, we review systems that incorporated the use of ML approaches. This section presents an overview of the various ML and DL based solutions that have been proposed for use in HAR. Recognizing activities need various devices to collect data and various algorithms to recognize activities based on this data. Algorithms used in activity recognition devices are mainly divided into vision-based (cameras) [15], sensor-based (wearable sensors, smartphones, dense sensing) [27], and Wi-Fi-based (RSS [16], CSI signal [11]). Thee algorithms are further subdivided into traditional and DL. Below, we list a number of studies that employed CSI signals and the relevant algorithms used.

In [11], an LSTM model was proposed for HAR. This model used one hidden layer with 200 hidden units. The model achieved an accuracy of over 75% compared with to two traditional algorithms: RF and HMM. Moreover, it had the advantage of being able to directly extract features without using any feature extraction technique (such as RF or HMM).

A CARM system was designed by [28] with the aim of defining human activity using CSI. This system rested upon two models: a CSI-speed model and a CSI-activity model. The relationship between the speed of human movements and CSI dynamics was computed using the CSI-speed model, and the relation between the speed of human movement and human activities was computed using the CSI-activity model. The model was shown to be highly sensitive to minute changes in the surrounding environment, with an accuracy rate of 96%.

Another system designed for HAR was the BLSTM [29]. The BLSTM was designed for feature learning using past and future data for CSI signals. A proposed ABLSTM [12] was also tested using the same dataset and its efficacy compared with other algorithms: (RF, HMM, SAE, and LSTM). Better recognition of all human activities occurred with the use of the ABLSTM compared with other algorithms, with an accuracy rate of $\geq$ 95%.

A deep RNN was proposed by [30]. The researchers used a CNN to extract features and used an LSTM to classify data. This model achieved an accuracy of 95%. In addition, the Principal Component Analysis (PCA) technique [31] was employed to eliminate noise resulting from objects such as doors and walls, which cause static multipath from CSI data.

In [32], both amplitude and phase were extracted. A low pass Butterworth filter and PCA were used for preprocessing. The authors proposed an empirical mode decomposition (EMD) for segmentation and feature extraction. Activities including pushing, dodging, striking, pulling, dragging, kicking, circling, punching (twice), and bowling were classified using the RF algorithm. The confusion matrix for both line-of-sight (LOS) and not in the line-of-sight (NLOS) was presented. LOS achieved a higher accuracy than NLOS, whose accuracy of 89.147%. Wi-Fi CSI was preprocessed with a Hampel filter to eliminate outliers and a low-pass filter to remove noise [33]. Features were then extracted and normalized so that data could be classified using an SVM [34]. Accuracy reached a rate of 98.4%.

Data from the SignFi public dataset and FallDeFi were used in the (CsiGAN) HAR model [35], which consisted of two generators and one discriminator added for recognition. This was needed to increase the performance of the left-out user. The researchers used multiple models, with the CsiGAN achieving the highest accuracy. Accuracy rates stood at 84.17% with the SignFi dataset and 86.27% with the FallDeFi dataset.

A Wi-Motion model [36] was proposed as an enhanced HAR model. It was designed to use an amplitude and phase extracted from CSI. A Weighted Moving Average (WMA) was used to remove noise and a PCA to remove redundant information. Phase calibration was used for phase preprocessing and a discrete wavelet transform (DWT) was used for feature extraction for both amplitude and phase. Lastly, a SVM with dynamic time warping (DTW) was used for activity recognition. Average accuracy rates stood at 96.6% in the LOS and 92% activity in the NLOS.

The work of [37], involved a WiAct HAR system and an Adaptive Activity Cutting Algorithm (AACA). Recognition depended on the difference in signal variance between the action and non-action, were. The researchers used an HMM, an SAE and an LSTM in addition to their proposed Extreme Learning Machine (ELM). The ELM achieved the highest accuracy, which stood at 94.20%.

In [38], the authors were able to use a device-free method (CSI) to recognize human activity. They introduced two classification algorithms: a Support Vector Machine (SVM) and a long short-term memory (LSTM) algorithms. Wavelet analysis was used to the purposes of preprocessing and feature extraction. As a result, they were able to recognize walking, sitting, standing, and running activities. In addition, they experimented with detecting “falling.” Lastly, they counted the number of people in a room using the same algorithms. As a result, most activities were recognized with accuracies of up 95%. The fall accuracy stood at 100%.

In the [39] study, a CNN and BLSTM model was used to detect spatial–temporal data as handcrafted features had failed. Various elements in the indoor setting that could have affected accuracy were studied in order to create an enhanced CSI dataset. Those included sizes of sliding windows and the number of hidden units. Data were increased by concatenating collected data from three regions. The accuracy obtained changed for all studied impacts and reached 96.96% for four activities. Public data were also used by [11], and a recognition accuracy of over 90% was achieved with all activities except standing up, for which the recognition accuracy stood at only 86%.

Synthetic data were used to reduce the need for real CSI in a study by [40]. A PCA was used to remove noise with high bandwidth. A Short-time Fourier transform (STFT) was used to transform the signal into the frequency domain, and a GAN was used to generate new data. A LSTM algorithm was used for classification. Accuracy stood at 87.2% with 50% ‘real’ plus 50% synthetic data, and at 92.8% with a set of all ‘real’ data. In the aforementioned works, a tenfold cross-validation evaluation method was used. Table 1 summarizes these works and highlights the algorithms used in recognition and accuracies reached achieved.

3 Convolution Neural Network and Recurrent Neural Network

Increasing recognition of the importance of DL is reflected by the plethora of recently published studies found in the literature. This may be attributable to its property of automatic feature extraction without handcraft features. Feature extraction and classification tasks are performed together. This is why DL is used in fields where it is hard to determine essential features mathematically. It is an advanced branch of ML, in which the network consists of multiple layers, each containing a group of neurons that build up the deep neural network (DNN). DNN technology is currently used in various applications such as image classification, object detection, indoor localization, and intrusion detection [41].

DL-based HAR is more robust than other frameworks, as features are extracted automatically without the need for features design. The same model can be used in a multitude of applications, with each application having its own set of data. DL-based technology is more scalable as it is able to handle any amount of data. Various DL approaches have been proposed, varying in their ability to perform the various required functions of ML such as feature learning and classification, clustering or finding correlations between a given set of data. The DL approaches are generally classified into three main categories: generative, discriminative, and hybrid. DBN and RNN are examples of generative approaches with LSTM and GRU being variations of RNN. CNN is an example of discriminative approaches, and GAN is an example of hybrid approaches.

Currently, the two most widely researched DL approaches are CNN and RNN. They have been incorporated into almost all human detection systems. In this section, differences between CNN and RNN are detailed. For a better understanding of the specifications of each of our proposed models, we also discuss the types of DL frameworks we used in this study.

3.1 Convolution neural network (CNN)

CNN models were the first DL models to be developed and have been used in many applications since. In order to reduce the high dimensionality of data and extract features automatically,

CNN was designed to have several neural layers. These layers are:

3.1.1. Convolution layer: This layer makes the analysis of input data easier. It is composed of multiple filters, which consist of a set of weights. Input data, which is fed into this layer, is used to produce feature maps.

3.1.2. Batch normalization: This layer calculates means and standard deviations of each input variable. It uses a mini-batch to speed the learning process and reduce overfitting.

3.1.3. Pooling (Max/Average): This layer summarizes data features by calculating their maximum or average. This layer helps in reducing translation invariance.

Table 1 Summary of related work

Reference no	Activities	Classification method	Accuracy
[11]	Lie down, Fall, Walk, Run, Sit down and Stand up	LSTM	75%
[28]	Run, Walk, Sit down, Open refrigerator, Fall, Box, Push one hand, and Brush teeth	CARM	96%
[12]	Lie down, Fall, Walk, Run, Sit down and Stand up	ABLSTM	95%
[30]	Lie down, Fall, Walk, Run, Si down and Stand up	CNN + LSTM	95%
[32]	Push, Dodge, Strike, Pull, Drag, Kick, Circle, Punch (twice), and Bowl	RF	89.147% for NLOS
[34]	Sit down and Stand up	SVM	98.4%
[35]	Sign language gestures and FallDeFi Data: Fall, Walk, Jump, pick up, Sit down, and Stand up	CSIGAN	84.17% for sign language 86.27% for FallDeFi
[36]	Bend, Halve squat, Step, Stretch leg, and Jump	SVM with DTW	96.6% for LOS 92% for NLOS
[37]	Push, Wave, Kick, Run, Fall, Box, Sit, Pick, Walk, and Empty	ELM	94.2%
[38]	Walk, Sit, Stand, Run and Fall	SVM, LSTM	95% and more for the rest of activities, 100% for fall
[39]	Bend, Box, Clap and Wave, Fall, pick up, Run, Sit down, Stand up and Walk	CNN + BLSTM	96.96% for four activities for three regions 90% for six activities except stand up is 86%
[40]	Lay down, Fall, Walk, Run, Sit down and Stand up	GAN + LSTM	87.2% With 50% accuracy and 50% Synthetic 92.8 for all data accurate

3.1.4. Fully connected layer: It also called Dense layer; its input is a vector obtained by flattening the three-dimensional output of the previous layer. The output of this dense layer is then fed to softmax activation function for classification.

3.2 Recurrent neural networks

This type of DL model is usually used with time sequence data and covers a wide range of applications that depend on the real-time response. Three important categories of RNN models have been developed: LSTM, BLSTM, and GRU. Initially, RNN was a class of neural networking that used previous outputs as inputs in the current step while having hidden states, but that suffered from short-term memory (ignoring important information at the beginning of the sequence if it is long). LSTM, BLSTM, and GRU were thus developed as RNNs with gates that regulate flow of information and a resultant long-term memory.

3.2.1 Long short-term memory

RNNs have solved the problem of vanishing gradients by featuring a memory block that stores weight values. Each memory block consists of three gates that decide the block's state and output. These gates are the *forget gate*, the *input gate*, and the *output gate*, as shown in Fig. 2.

- Forget gate: It is responsible for what information must be discarded from the unit.
- Input gate: It is responsible for which input values will update the memory state.
- Output Gate: It decides the block's output based on the input and the unit memory.

LSTM is used to overcome the long-term problem by using only the previous state as its input, and is defined as a unidirectional LSTM. Its structural complexity and

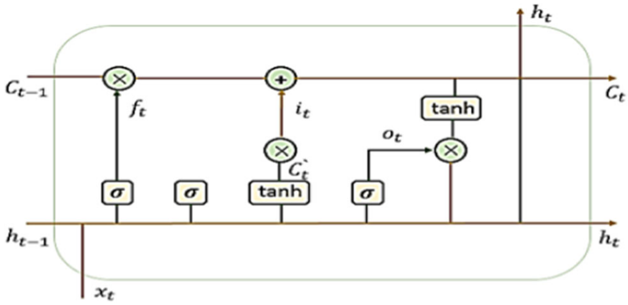


Fig. 2 LSTM architecture

multiple parameters make it fairly expensive. An LSTM's input gate is used to select user information and store it in the LSTM cell. In LSTMs, the forget gate is used to determine the information to be neglected, and the output gate is used to select information to be presented. It has memory cells that store information for a long time [19].

3.2.2 Bidirectional-long-short-term-memory

It is an extension of LSTM networking. It has the ability to train all data input in the past and future of a specific time frame. BLSTM [29] has the advantages of Bi-RNN and LSTM, being a combination of both. It overcomes the problem of long-term dependency and uses past and future states. It gives more data and better results. It, however, requires a long time to operate as the information is extracted from the two directions through the forward and backward layers, as shown in Fig. 3.

3.2.3 Gated recurrent unit

It is a lighter version of the LSTM in terms of the number of gates inside the unit and the computational cost. In GRUs, the forget and input layers are merged into a single cell. GRU architecture is shown in Fig. 4. It is similar to LSTM, but GRUs have fewer parameters, and combines the input and forget gates of LSTM into a single update gate [20]. GRUs have an update gate and a reset gate, as represented by Eqs. 1 and 2 [42].

The main advantage of GRUs is they are able to maintain the information relevant to the prediction for a long time without removing it. GRUs are straightforward and remarkably easy to modify and take less time to train. GRUs are thus more efficient than other models and were deemed suitable for the purposes of this work.

$$z_t = \sigma(W(z)x_t + U(z)h_{t-1}) \tag{1}$$

$$r_t = \sigma(W(r)x_t + U(r)h_{t-1}) \tag{2}$$

where z_t Update gate, r_t Reset gate, x_t The input, W Weight matrices, σ Sigmoid.

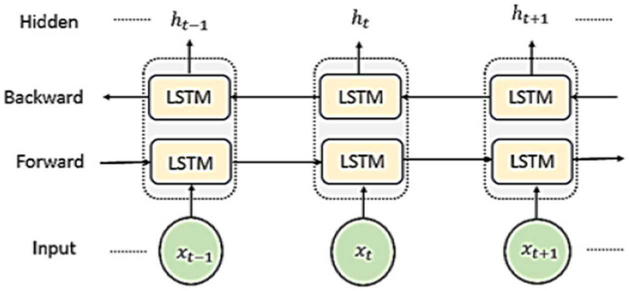


Fig. 3 BLSTM Architecture

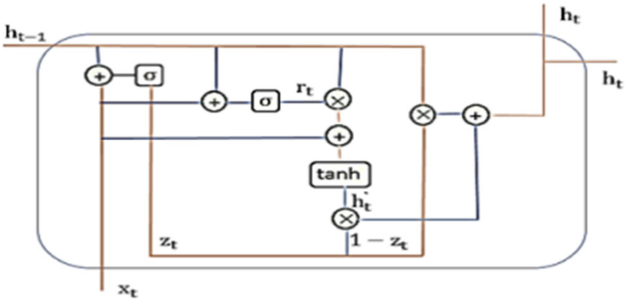


Fig. 4 GRU circuit

3.2.4 The attention mechanism

Attention is a recently introduced mechanism of DL that leads to more robust ML. It supports sequence-to-sequence models designed to map sequence input to sequence output. It helps such models focus on specific elements and pay less attention to the rest, allowing sequential reasoning to be performed, a feat that has been quite difficult for traditional deep learning models to perform. Starting with natural language processing, attention has now had a large number of applications in serval fields, including machine translation, healthcare, self-driving cars, speech recognition, and sentiment classification. The use of attention in HAR has led to better performance, giving models the ability to assign new weights and a larger number of features [43].

4 CSI-based HAR system design

By displaying channel disturbances caused by human activity, CSIs allow for the more precise recognition of human activity. Patterns of channel distortion are correlated with specific human movements/activities. The current action the person being monitored is identified using CSI signals received from the area where the person is performing the action. In our work, to achieve activity recognition based on CSI streams, we developed a DL model that is a combination of CNN and GRU networks.

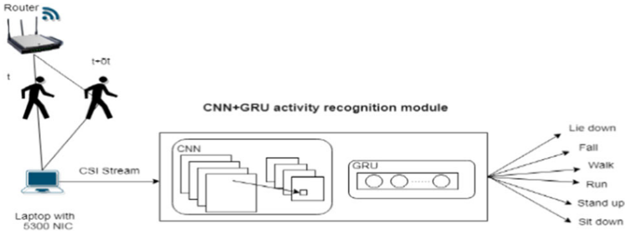


Fig. 5 Architecture CSI-based deep learning activity recognition system

The general architecture of our CSI-based DL activity recognition system is shown in Fig. 5. Public CSI data were collected in the indoor office using a router with three antennas to transmit Wi-Fi signals and a laptop with 5300 NIC to capture those signals. We extracted CSI amplitude streams from those captured signals and used them as inputs to the system. The system's role was to automatically extract features from the CSI stream and detect each activity. The output of our system was the recognized activity corresponding to the input.

In our models, we used CNNs due to their efficiency in deriving good features from raw CSI measurements. The models were subdivided into two main steps: feature extraction and classification. Our models worked by first extracting the required features from the CSI signals using CNN, GRU, LSTM and attention, then sending them for classification. The subsequent step involved classifying those extracted features and generating the correct activity. Those four proposed models are discussed in detail in the next subsections.

4.1 The CNN-GRU model

The CNN-GRU model is presented in Fig. 6. It consists of three parts: input, feature extraction, and classification. In the input layer, CSI data have several samples each equal to 1000, where each CSI sample has an initial matrix of 1 × 3 × 30. After reshaping the CSI samples to be suitable as inputs to CNNs, their matrices grow to a size of 1000 × 30 × 3. The feature extraction part consists of two convolutional layers and a GRU layer. The first layer filters data with a size 5 × 5 × 128 kernel and size 1 × 1 stride followed by batch normalization, ReLU activation, average pooling to reduce the number of the parameters with size 2 × 2 pool, size 2 × 2 stride, and dropout to prevent overfitting during training with a value equal to 0.6. The second convolutional layer is of the same structure except that it has no dropout. Finally, the output passes through the flattened layer with time distributed input to convert data into a vector suitable for the GRU layer, consisting of 256 units. It is used to obtain time sequence data, and its output connected to the fully connected layer for classification.

4.2 The CNN-GRU-attention model

The CNN-GRU-Attention model is shown in Fig. 7. This model uses an attention layer that assigns larger weights and time steps, with CNNs and GRUs being used for feature extraction. Size 1000 × 30 × 3 CSI data are used as the input. The feature extraction part includes two convolution layers and a GRU layer. The first layer has a size 5 × 5 × 128 kernel and size 1 × 1 stride, followed by batch normalization, ReLU activation, average pooling with size 2 × 2 pool, size 2 × 2 stride, and dropout to prevent overfitting during training with a value equal to 0.6. The second convolution layer is of the same structure but without a dropout. After that, data exit through a flatten layer which reshapes it into a form then enters the GRU layer (which is made of several size 256 units). Next, data move through the attention, which focuses on specific

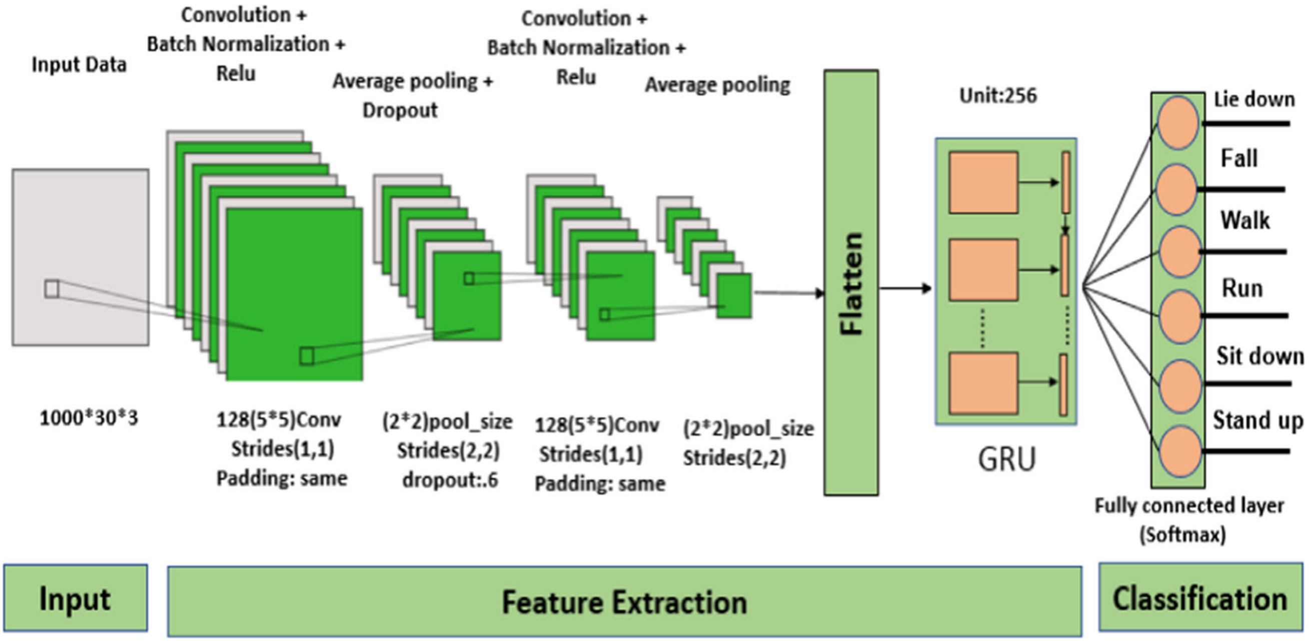

Input Data
Convolution + Batch Normalization + Relu
Average pooling + Dropout
Convolution + Batch Normalization + Relu
Average pooling
Flatten
Unit:256
GRU
Lie down
Fall
Walk
Run
Sit down
Stand up
Fully connected layer (Softmax)
1000*30*3
128(5*5)Conv Strides(1,1) Padding: same
(2*2)pool_size Strides(2,2) dropout:.6
128(5*5)Conv Strides(1,1) Padding: same
(2*2)pool_size Strides(2,2)
Input
Feature Extraction
Classification

Fig. 6 The CNN-GRU human activity recognition model

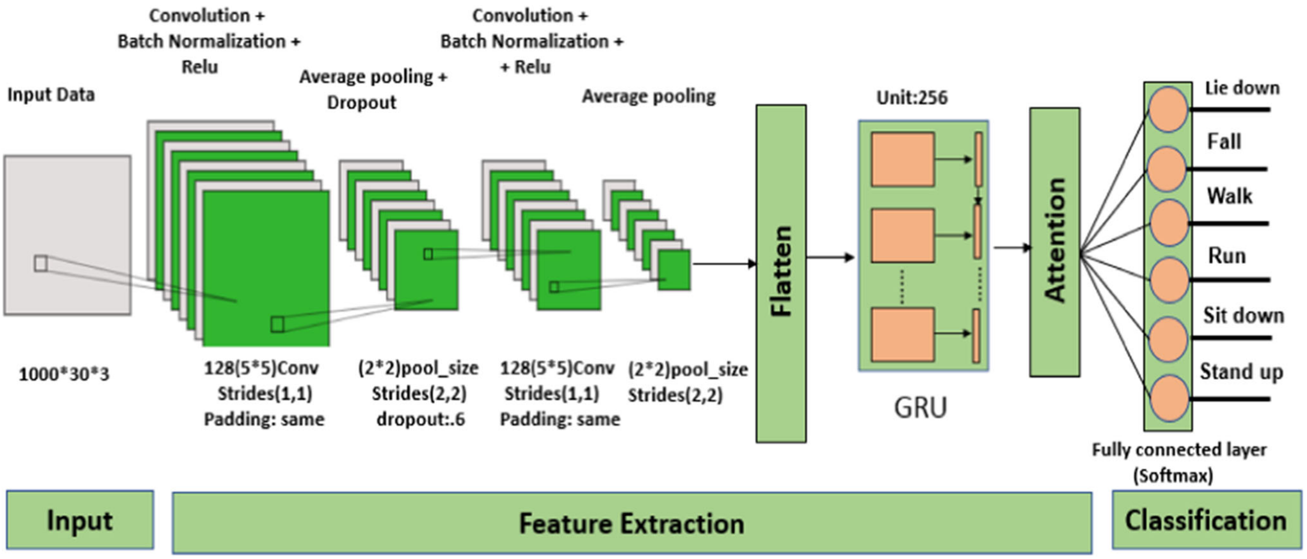


Fig. 7 The CNN-GRU-attention human activity recognition model

features of the input. Finally, data are fed into the classification layer, which classifies it into one of the activities.

4.3 CNN-GRU-CNN model

In our third model, CNN-GRU-CNN, the first CNN is used for automatic feature extraction, GRU is used to extract time series features, and the second CNN is used to expand the abstract features. As shown in Fig. 8, after the input layer, with an input of 1000 × 30 × 3, there is a convolution layer with 128 filters with a size 5 kernel, size 1 stride, batch normalization, and a ReLU activation function. These are followed by an average pooling layer with size 2 pool, size 2 stride, dropout of 0.6, and ending with a GRU layer with 256 neurons. This is followed by another convolution layer with a 128 filter with a size 5 kernel, size 1 stride, batch normalization, a ReLU activation function and another average pooling layer with size 2 pool, and size 2 stride. Finally, the output of the second CNN layer is fed to the flatten layer to deliver data to a vector, making the data available to a dense layer. Finally, data are fed into the classification layer.

4.4 The CNN-LSTM-CNN model

The structure of the CNN-LSTM-CNN model is comparable to that of the CNN-GRU-CNN model, with an LSTM replacing the GRU. The first CNN is used for automatic

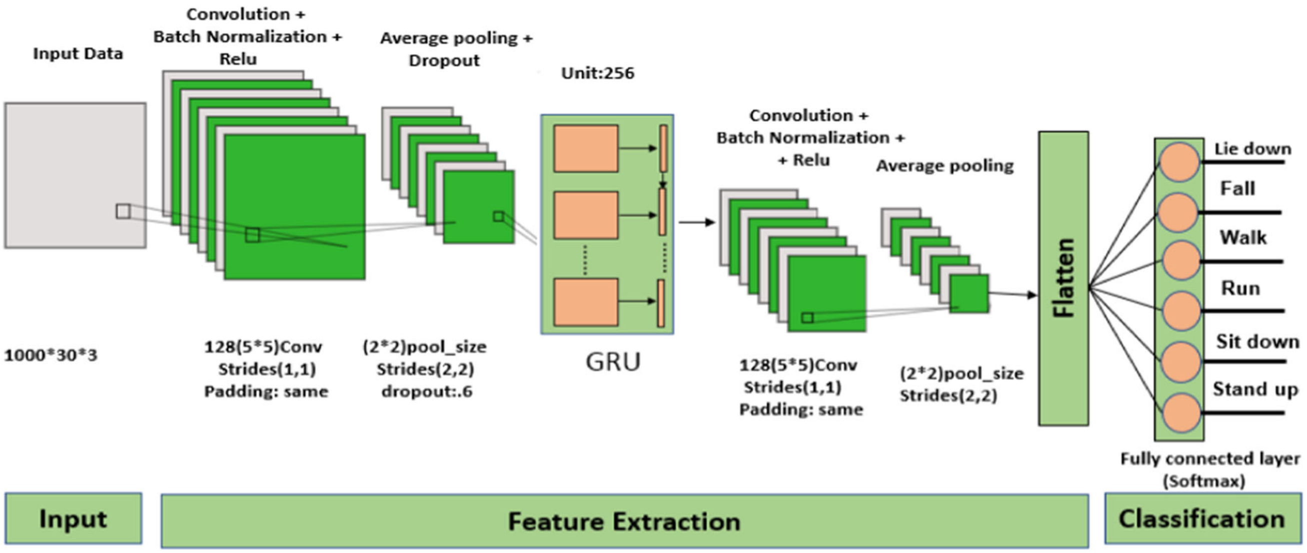


Fig. 8 The CNN-GRU-CNN human activity recognition model

feature extraction, the LSTM for the extraction of time series features, and the second CNN is used to expand abstract features.

As shown in Fig. 9, after the input layer with the input of 1000 × 30 × 3, there is a convolution layer with 128 filters with size 5 kernel, size 1 stride, batch normalization, and a ReLU activation function. The average pooling layer has pool of size 2, stride 2, a dropout of 0.6, an LSTM layer with 256 neurons, another convolution layer with the 128 filters with size 5 kernel, size 1 stride, batch normalization, and a ReLU activation function, and another average pooling layer with size 2 pools and size 2 stride. The output of the second CNN layer is fed to the flatten layer to represent the data into a vector to be available for a dense layer. Lastly, the flattened data are fed into the classification layer. The classification part is the same for all four models. A fully connected layer with dense equal to class number 6 and softmax activation is used to determine one activity for each input. All parameters are initialized randomly. Errors are computed using the categorical-cross entropy loss type. An Adaptive Moment Estimation (Adam) optimizer is used with a learning rate of 0.0007. A size 64 batch value is used.

5 Experimental work and results

This section describes the results of evaluations we conducted to test the efficacies of our proposed CSI-based human activity models and the training schemes we tested as well. We also report on the comparisons we conducted between our models and existing state-of-the-art approaches [11, 12], using a public benchmark dataset.

5.1 Dataset used

We used a public dataset collected by [11] to train our models. The dataset is comprised of CSI data collected using an Intel 5300 NIC run laptop. This dataset is compatible with all devices and suitable only for a family of IEEE 802.11a/b/g/n devices and only in the presence of a CSI tool. The CSI tool was installed using a Linux operating system containing a modified firmware and wireless driver. The laptop and the CSI tool were used as receivers and the MIMO router as a transmitter with 1000 packets. The setup had three antennas and 30 subcarriers, thus generating 1000 × 30 × 3 shaped data. A camera was also set up in order to capture and label each activity. A log file was used to record packets with Matlab scripts. Each file contained several samples: some samples with no activities and others with activities.

The six activities tested were labeled: *Lie down, Fall, Walk, Run, Sit down* and *Stand up*. Six persons in an indoor office performed the activities, with each person repeating each activity a total of 20 times. We downloaded this dataset from [https://github.com/ermongroup/Wifi_Activity_ Recognition] and selectively collected data of activity samples for classification. The number of samples chosen for each activity is shown in Table 2.

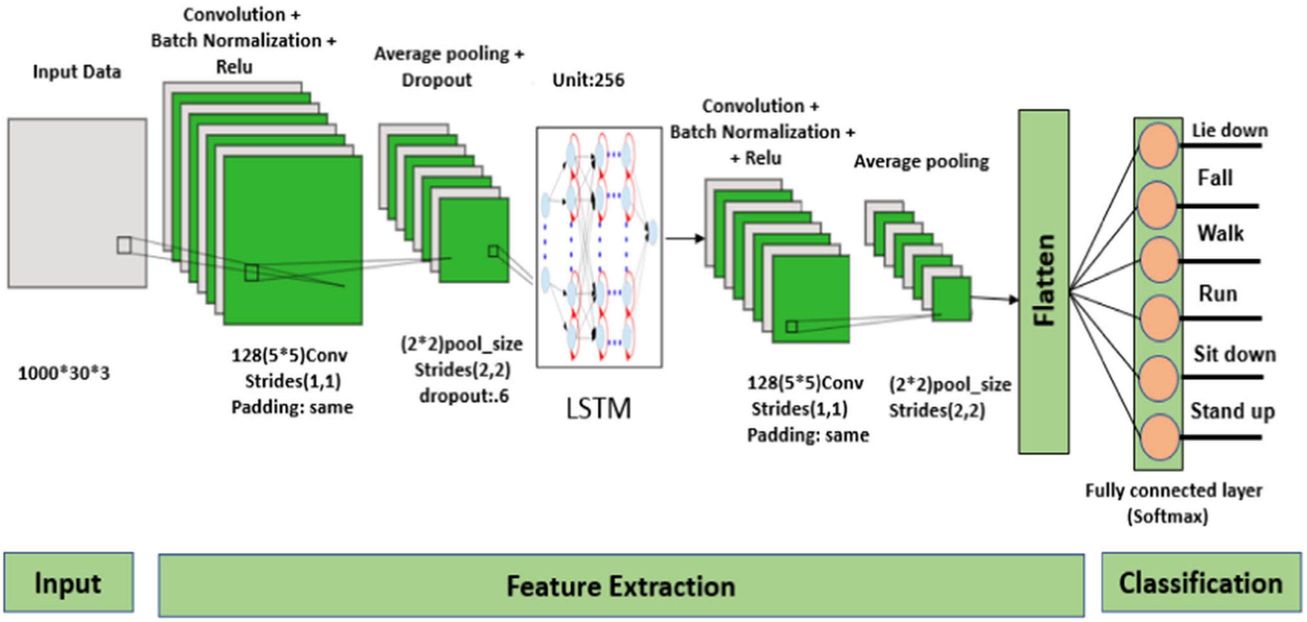


Fig. 9 The CNN-LSTM-CNN human activity recognition model

Table 2 Number of samples per activity

Activity	No. of Samples
Lie Down	1318
Fall	889
Walk	2931
Run	2408
Sit down	812
Stand up	601
Total	8959

This dataset has previously been used in a number of published studies [11, 12, 30, 39, 40]. We compared our results against two of these studies [11, 12]. The work in [30] evaluates the model for behavior recognition using CSI on 6 activities, with only 120 instances of each activity not all instances, while we use all the instances which will make the comparison unfair for them. The work in [39] uses 6 activities different than those we consider including (Fall, pick up, Run, Sit down, Stand up and Walk) from the public dataset are used but our work based on 6 activities in [11] including (Lie down, Fall, Walk, Run, Sit down and Stand up). Finally, the work in [40], they use only 50% of each class of this benchmark dataset which is less than we have considered in our work (we used all the dataset).

To experiment with our proposed models, we used a Google Colab that ran commands in pro-version Python. Our work was conducted using a GPU Tesla P100-PCI-E-16 GB with 25 GB of memory.

5.2 Experiments conducted

As previously discussed, this study experimented with four Wi-Fi CSI-based HAR DL systems: a CNN-GRU model, a CNN-GRU-Attention model, a CNN-GRU-CNN model and a CNN-LSTM-CNN model. The performances of these models were evaluated and comparatively assessed against two recently introduced state-of-the-art DL models, the LSTM [11] and the ABLSTM [12]. As in the [11, 12] studies, tenfold cross-validation (which divided data 10-folds) was used for dataset training and testing, with one-fold selected for testing, and the other nine folds used for training. This process was repeated ten times with the average of all runs used as a benchmark of the performance of the model being evaluated.

In such cases, the model being trained using the10-fold training method may not recognize new samples, and its performance is thus only tested for data seen before. This creates a problem when new data are fed into the system. Consequently, we choose to conduct another training method to ensure real-time accuracy of the system when dealing with unseen data. This method splits the dataset into 70% and 30% groups (instead of the10-fold method) for the purposes for training and testing, respectively. Samples used for training are not used for testing. The performance of a system which is trained using the spilt method is thus closer to real-world performance compared with a system trained using the tenfold training method.

A number of performance metrics can be used to evaluate DL models, including accuracy, precision, recall, and AUC. These metrics are computed upon the following equations [44, 45]:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3}$$

$$\text{Recall} = \frac{TP}{TP + Fn} \tag{4}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \tag{5}$$

$$\text{AUC} = \frac{S_p - (n_p(n_n + 1)/2}{n_p n_n} \tag{6}$$

where TP is *true positive*, FP is *false positive*, TN is *true negative*, FN is *false negative* , S_p is the sum of all positive samples,n_p is the number of positive samples, and n_n is the number of negative samples.

5.3 Results of experiments

As aforementioned, the performances of the proposed CNN-GRU, CNN-GRU-Attention, CNN-GRU-CNN and CNN-LSTM-CNN models were compared with the performances of two recent activity recognition models that use CSI signals: the LSTM [11] and ABLSTM [12]. The resultant confusion matrices of the six models (the four from our study and the two they were being compared to) are presented in this section. Tables 3 and 4 show the confusion matrices for both the LSTM [11] and ABLSTM [12] models, as reported by the authors of the two studies. The LSTM model [11] reportedly achieved an accuracy level at recognizing collaborative activities of up to 75%, while the ABLSTM model [12] achieved an accuracy of up to 95% with all activities.

We tested our models twice: once using the k-fold method and then after randomly splitting the dataset into 70% for training and 30% for testing. We also plotted accuracies and compared training and validation phases of each epoch with respect to loss, precision, accuracy, recall, and AUC.

Results indicated that our CNN-GRU model had achieved accuracies for activities of up to 99.14% with the

Table 3 Confusion matrix of the (LSTM) model [11]

LSTM [11] Predicted							
		Lie down	Fall	Walk	Run	Sit down	Stand up
Actual	Lie down	*0.95*	0.01	0.01	0.01	0.00	0.02
	Fall	0.01	*0.94*	0.05	0.00	0.00	0.00
	Walk	0.00	0.01	*0. 93*	0.04	0.01	0.01
	Run	0.00	0.00	0.02	*0.97*	0.01	0.00
	Sit down	0.03	0.01	0.05	0.02	*0.81*	0.07
	Stand up	0.01	0.00	0.03	0.05	0.07	*0.83*

Table 4 Confusion matrix of the (ABLSTM) model [12]

ABLSTM [12] Predicted							
		Lie down	Fall	Walk	Run	Sit down	Stand up
Actual	Lie down	*0.96*	0.0	0.01	0.0	0.02	0.01
	Fall	0.0	*0.99*	0.0	0.01	0.0	0.0
	Walk	0.0	0.0	*0.98*	0.02	0.0	0.0
	Run	0.0	0.0	0.02	*0.98*	0.0	0.0
	Sit down	0.01	0.01	0.01	0.0	*0.95*	0.02
	Stand up	0.01	0.0	0.0	0.0	0.01	*0.98*

Table 5 Confusion matrix of the CNN – GRU learning model:(a) with k-fold, and (b) with randomly splitting of the dataset

		Lie down	Fall	Walk	Run	Sit down	Stand up
(a) CNN-GRU with k-fold Predicted							
Actual	Lie down	*0.9969*	0.0	0.0015	0.0	0.0008	0.0008
	Fall	0.0	*1.00*	0.0	0.0	0.0	0.0
	Walk	0.0	0.0	*1.00*	0	0.0	0.0
	Run	0.0	0.0	0.0	*0.9996*	0.0	0.0004
	Sit down	0.0037	0.0024	0.0	0.0	*0.9914*	0.0024
	Stand up	0.0017	0.0	0.0	0.0017	0.0033	*0.9933*
(b) CNN-GRU with randomly splitting of the dataset Predicted							
Actual	Lie down	*0.9975*	0.0	0.0025	0.0	0.0	0.0
	Fall	0.0	*1.00*	0.0	0.0	0.0	0.0
	Walk	0.0	0.0	*1.00*	0	0.0	0.0
	Run	0.0	0.0	0.0	*1.00*	0.0	0.0
	Sit down	0.0	0.0	0.0	0.0	*1.00*	0.0
	Stand up	0.0	0.0	0.0	0.0	0.0	*1.00*

for k-fold method, an accuracy level of 100% recognizing all activities, except lying down, for which the activity recognition accuracy stood at 99.46% after using the split method (see Table. 5). Plots of model performance are presented in Fig. 10

The CNN-GRU-Attention model achieved accuracies of accuracy for activities of up to 99.33% with the use of the k-fold method and of up to 99.49% after randomly splitting the dataset, as shown in Table 6. Plots of the model's performance are shown in Fig. 11. As shown in Table 7, after the use of the k-fold method, the CNN-GRU-CNN model achieved accuracies for activities of up to 98.89%, and after the use of the split method of up to 98.63%. Model performance is presented in Fig. 12.

Lastly, the CNN-LSTM-CNN model achieved accuracies for activities of up to 96.43% after using k-fold and of up to 98.44% after using the split method, as detailed in Table 8. Model performance is illustrated in Fig. 13.

A summary of the above results is shown in Table 9. Results indicate that our models had achieved consistently high levels of accuracy with new data; providing robust and stable performance.

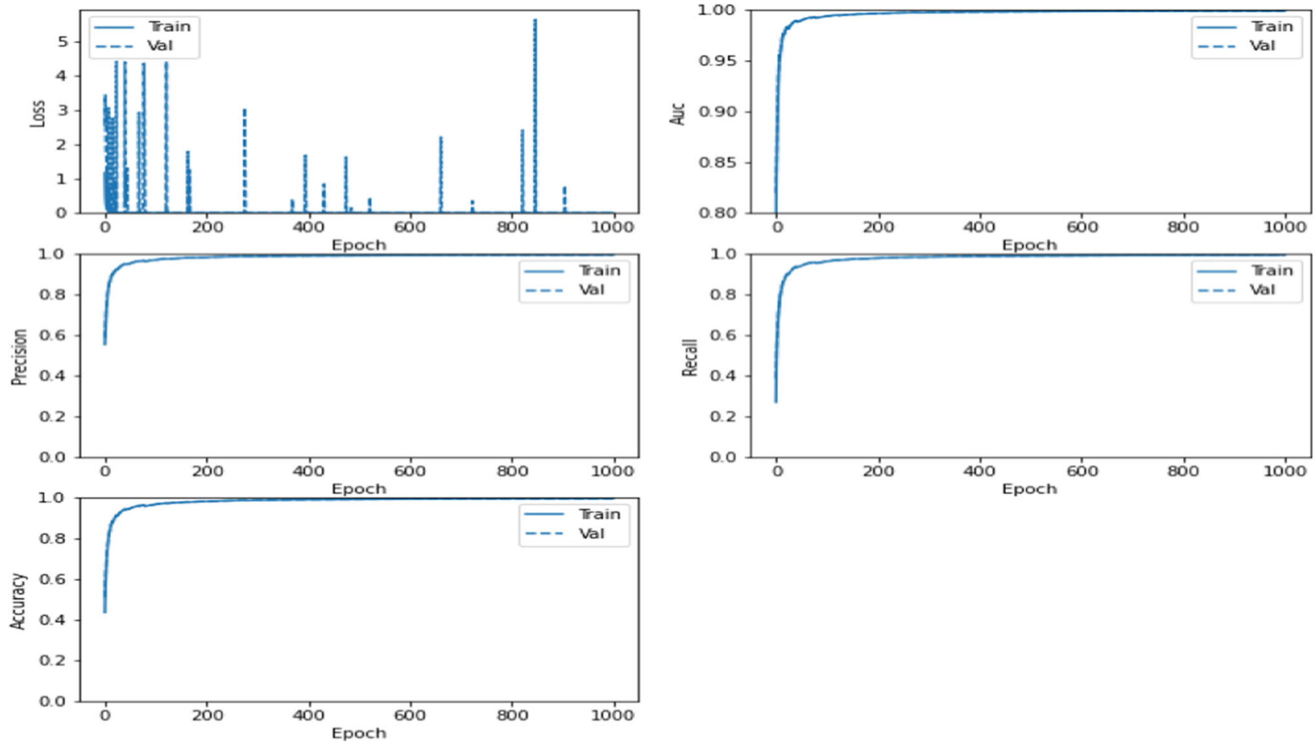


Fig. 10 CNN-GRU model performance results

Table 6 Confusion matrix of the CNN-GRU-Attention deep learning model:(a) with k-fold, and (b) with randomly splitting of the dataset

	Lie down	Fall	Walk	Run	Sit down	Stand up	
(a) CNN-GRU-Attention with k-fold Predicted							
Actual	Lie down	*0.9977*	0.0	0.0008	0.0	0.0015	0.0
	Fall	0.0	*1.00*	0.0	0.0	0.0	0.0
	Walk	0.0	0.0003	*0.9990*	0.0003	0.0003	0.0
	Run	0.0	0.0	0.0	*1.00*	0.0	0.0
	Sit down	0.0012	0.0	0.0	0.0	*0.9951*	0.0037
	Stand up	0.0	0.0017	0.0017	0.0	0.0033	*0.9933*
(b) CNN-GRU-attention with randomly splitting of the dataset Predicted							
Actual	Lie down	*0.9949*	0.0	0.0051	0.0	0.0	0.0
	Fall	0.0	*1.00*	0.0	0.0	0.0	0.0
	Walk	0.0023	0.0	*0.9954*	0.0	0.0	0.0023
	Run	0.0	0.0	0.0	*1.00*	0.0	0.0
	Sit down	0.0	0.0	0.0041	0.0	*0.9959*	0.0
	Stand up	0.0	0.0	0.0	0.0	0.0	*1.00*

Table 10 illustrates our model's metrics including Loss, Accuracy, precision, Recall, and AUC. Those are the validation values that our models reached at the last epoch. From those results, it is found that CNN-GRU model has the best value in each metric (i.e., lowest Loss, highest accuracy, precision, recall and AUC values).

Another yet important performance evaluation metric is the consumed time, which indicates the time taken by the model for training and testing. Table 11 shows the time consumption comparison for the six DL approaches: LSTM [11], ABLSTM [12], the proposed CNN-GRU, CNN-GRU-Attention, CNN-GRU-CNN and CNN-LSTM-CNN

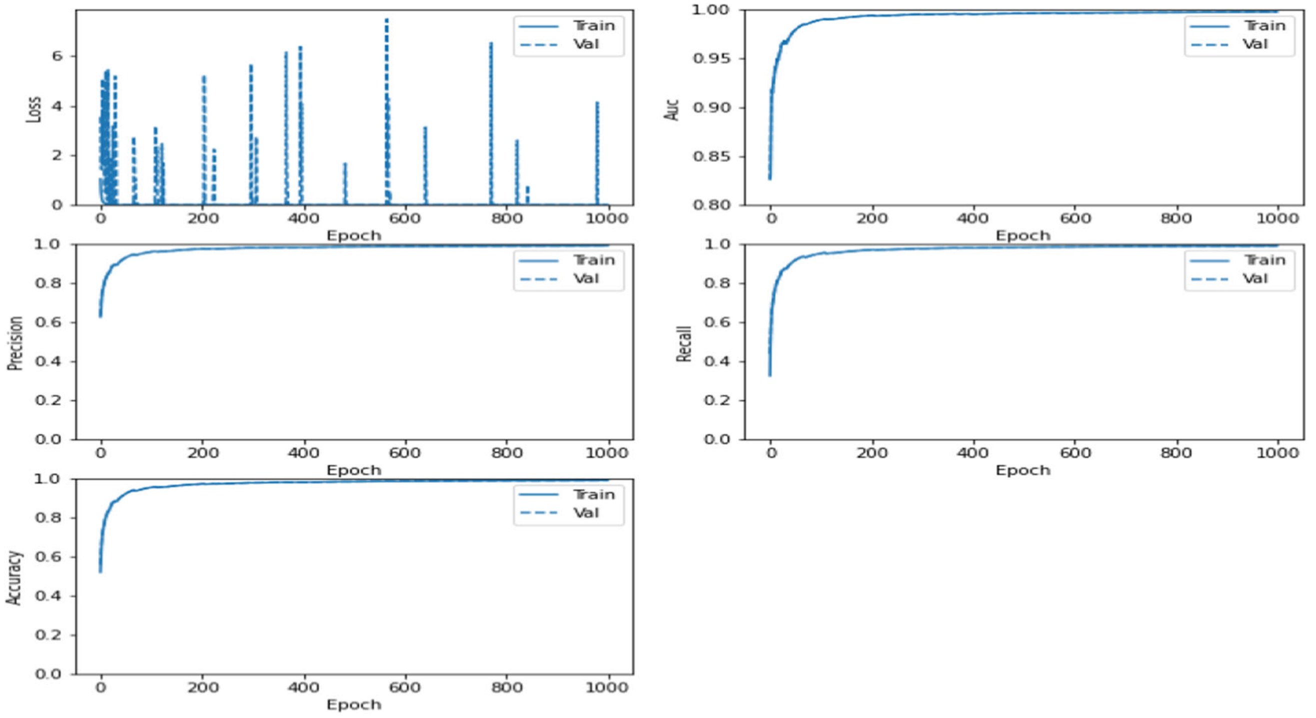


Fig. 11 CNN-GRU-Attention model performance results

Table 7 Confusion matrix of the CNN-GRU-CNN deep learning model:(a) with k-fold and (b) with randomly splitting of the dataset

	Lie down	Fall	Walk	Run	Sit down	Stand up	
(a) CNN-GRU-CNN with k-fold Predicted							
Actual	Lie down	*0.9939*	0.0	0.0	0.0008	0.0053	0.0
	Fall	0.0	*0.9989*	0.0	0.0011	0.0	0.0
	Walk	0.0	0.0	*0.9990*	0.0010	0.0	0.0
	Run	0.0004	0.0	0.0008	*0.9988*	0.0	0.0
	Sit down	0.0025	0.0	0.0012	0.0	*0.9889*	0.0073
	Stand up	0.0	0.0017	0.0017	0.0	0.01033	*0.9933*
(b) CNN-GRU-CNN with randomly splitting of the dataset Predicted							
Actual	Lie down	*0.9873*	0.0	0.0	0.0	0.0127	0.0
	Fall	0.0	*0.9963*	0.0	0.0037	0.0	0.0
	Walk	0.0	0.0	*0.9875*	0.0079	0.0	0.0046
	Run	0.0	0.0	0.0028	*0.9972*	0.0	0.0
	Sit down	0.0	0.0	0.0041	0.0	*0.9877*	0.0082
	Stand up	0.0	0.0	0.0	0.0	0.0111	*0.9889*

models. We can see that our models have the lowest time among the time of LSTM and ABLSTM, and the model that has the lowest time is CNN-GRU-Attention as shown in Table 11.

As shown in Table 12, it is found that LSTM model [11] has the lowest number of parameters (234,006) which takes small memory size compared to the other models. However, its accuracy is lower than ABLSTM [12] and the four proposed models. This is because it works sequential, so it is very slow taking large time 0.006 s for each sample when testing the data.

The model that comes after LSTM [11] in number of parameters (914,567) which represent their existence in the memory is the proposed CNN-GRU-Attention model. It has the highest accuracy which is up to 99.33% for all activities and also has the smallest response time (0.0019 s) for each sample. Therefore, we recommend using the proposed CNN-GRU-Attention model in real-time applications where the response time is the main concern. It also

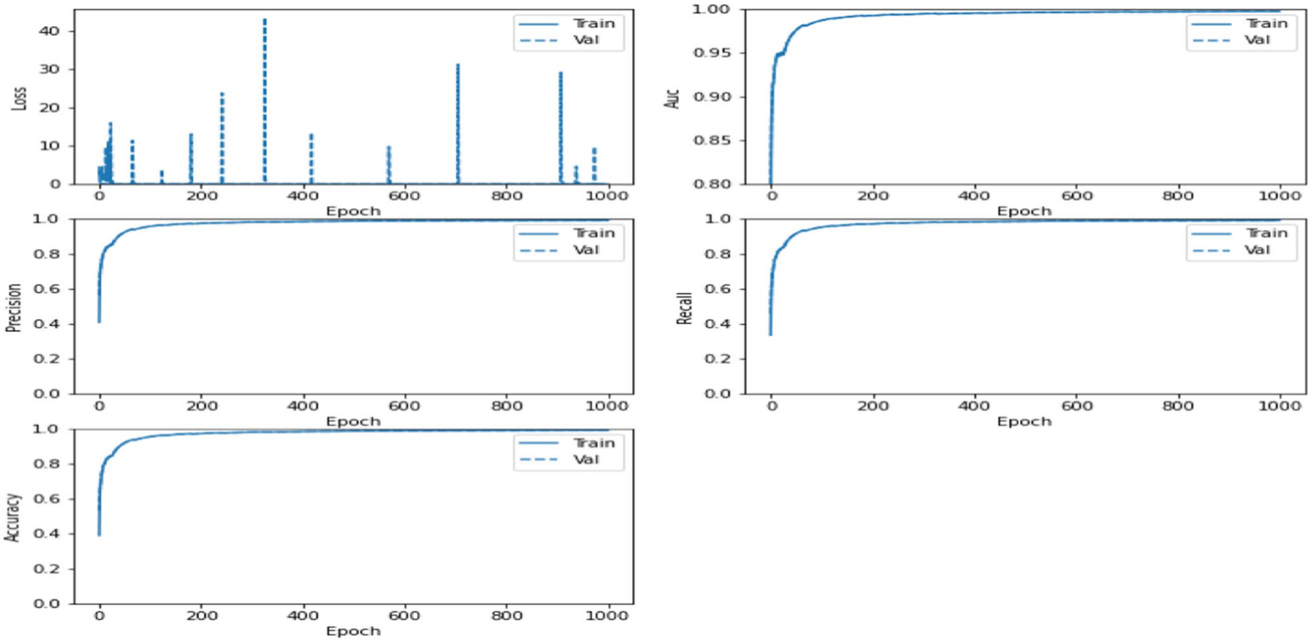


Fig. 12 CNN-GRU-CNN model performance results

Table 8 Confusion matrix of the CNN-LSTM-CNN deep learning model:(a) with k-fold, and (b) with randomly splitting of the dataset

	Lie down	Fall	Walk	Run	Sit down	Stand up	
(a) CNN-LSTM-CNN with k-fold Predicted							
Actual	Lie down	*0.9772*	0.0008	0.0061	0.0	.0061	0.0099
	Fall	0.0	*0.9989*	0.0011	0.0	0.0	0.0
	Walk	0.0003	0.0	*0.9976*	0.0017	0.0	0.0003
	Run	0.0004	0.0012	0.0120	*0.9855*	0.0	0.0008
	Sit down	0.0111	0.0012	0.0062	0.0	*0.9643*	0.0172
	Stand up	0.0	0.0	0.0017	0.0	0.0116	*0.9867*
(b) CNN-LSTM-CNN with randomly splitting of the dataset Predicted							
Actual	Lie down	*0.9899*	0.0	0.0	0.0	.0101	0.0
	Fall	0.0	*1.00*	0.0	0.0	0.0	0.0
	Walk	0.0	0.0	*0.9966*	0.0034	0.0	0.0
	Run	0.0	0.0	0.0	*1.00*	0.0	0.0
	Sit down	0.0041	0.0	0.0082	0.0	*0.9877*	0.0
	Stand up	0.0	0.0	0.0	0.0	0.0056	*0.9944*

is preferable for applications that has restricted size of memory (like for mobile applications) where is has low memory requirements and faster which will reduce the power consumption of such devices.

6 Conclusions and future work

HAR has become one of the most popular and most wildly researched topics of computer science. HAR technology has seen rapid progress and is now used in a wide variety of applications. Further advances in HAR technology are achievable through the development of new DL techniques. In this work, HAR involved the use of Wi-Fi devices. CSI was presented through the use of a combination of convolution layers and one GRU layer in a first model; convolution layers and one GRU layer and attention in a second model, and convolution layers with one LSTM layer in a third model. Ten-fold cross-validations of six activities were used in order to compare the performances of our models with the reported performances of an LSTM and an ABLSTM model. Our models achieved accuracies of 99.31%, 99.16, 98.88% and 98.95%, compared with accuracies of 75%, and 95% achieved by the LSTM and ABLSTM models, respectively.

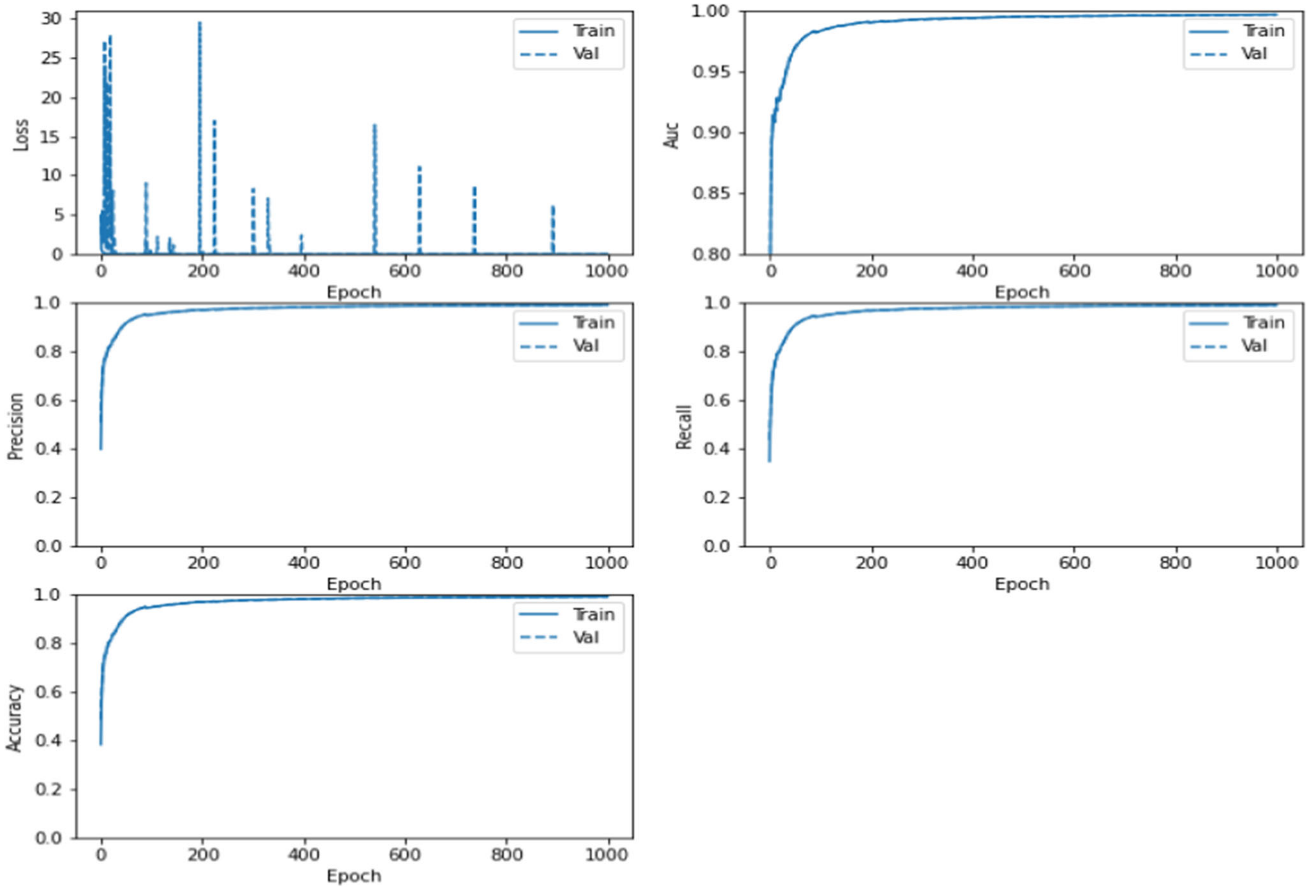


Fig. 13 CNN-LSTM-CNN model performance results

Table 9 Average Accuracy for our four proposed models

Models	Accuracy with k-fold method (%)	Accuracy with split method (%)
CNN-GRU	99.31	99.46
CNN-GRU-Attention	99.16	99.05
CNN-GRU-CNN	98.88	99.05
CNN-LSTM-CNN	98.71	98.99

Table 10 Performance metrics results for the four proposed deep learning models

Proposed Model	Loss	Accuracy	Precision	Recall	AUC
CNN-GRU	.0026	99.46	99.52	99.43	99.90
CNN-GRU-Attention	.0103	99.05	99.14	99.01	99.77
CNN-GRU-CNN	.0411	99.05	99.09	99.03	99.74
CNN-LSTM-CNN	.0340	98.99	99.03	98.96	99.70

Our CNN-GRU and CNN-GRU-Attention models yielded superior performance, with the highest improvements in recognition accuracy among the proposed models as well as the LSTM and ABLSTM models. A randomly splitting dataset method was also used to predict performance in a real-time environment where the input data may be unseen before the training phase. Resultant evaluation metric values pertaining to the proposed CNN-GRU model were: accuracy 99.46%, precision 99.52%, recall 99.43% and AUC 99.90%. The proposed CNN-GRU-Attention model attained evaluation metric as follows: accuracy 99.05%, precision 99.14%, recall 99.01% and AUC 99.77. These accuracies indicate the robustness of our proposed models and that their superior results (compared with the LSTM and ABLSTM models) were independent of training method.

For future work, we may consider using a denoising algorithm for the data before training and testing. We also may train both amplitude and phase datasets instead of

Table 11 Consumed Time for the Deep Learning Models per sample

Time	LSTM [11]	ABLSTM [12]	CNN-GRU	CNN-GRU-Attention	CNN-GRU-CNN	CNN-LSTM-CNN
Training (sec)	.011	.042	.0063	.0041	.0047	.0082
Testing (sec)	.006	.019	.0033	.0019	.0022	.0045

Table 12 Models overall comparison in terms of Response Time, Accuracy for all activities, and Total model parameters

Model	Response time (sec)	Accuracy for all activities (up to)%	Total model parameters
LSTM [11]	.0060	81	*234,006*
ABLSTM [12]	.0190	95	3,866,606
CNN-GRU	.0033	99.14	1,307,526
CNN-GRU-Attention	*.0019*	*99.33*	914,567
CNN-GRU-CNN	.0022	98.89	2,469,766
CNN-LSTM-CNN	.0045	96.43	2,568,326

using a single characteristic of the CSI signal. In addition, other human activities recognition could be explored such as direction of movement and fine-grained activities, Multi-user consideration could be put in mind if a dataset that includes multi-user scenarios is available.

Declarations

Conflict of interest All authors have declare that they have no conflict of interest.

References

1. Wang Y, Wu K, Ni LM (2016) Wifall: device-free fall detection by wireless networks. IEEE Trans Mob Comput 16:581–594
2. Brémond F, Thonnat M, Zúniga M (2006) Video-understanding framework for automatic behavior recognition. Behav Res Methods 38:416–426
3. Politi O, Mporas I, Megalooikonomou V (2014) Human motion detection in daily activity tasks using wearable sensors. In: Proceedings of the 22nd European signal processing conference (EUSIPCO), pp 2315–2319
4. Zhang O, Srinivasan K (2016) Mudra: user-friendly fine-grained gesture recognition using WiFi signals. In: Proceedings of the 12th international on conference on emerging networking experiments and technologies, pp 83–96
5. Own CM, Hou J, Tao W (2019) Signal fuse learning method with dual bands WiFi signal measurements in indoor positioning. IEEE Access 7:131805–131817
6. Halperin D, Hu W, Sheth A, Wetherall D (2011) Tool release: gathering 802.11 n traces with channel state information. ACM SIGCOMM Comput Commun Rev 41:53
7. Horning N (2010) Random forests: an algorithm for image classification and generation of continuous fields data sets. In: Proceedings of the international conference on geoinformatics for spatial infrastructure development in earth and allied sciences,pp 9–11
8. Rabiner LR (1989) A tutorial on hidden Markov models and selected applications in speech recognition. Proc IEEE 77:257–286
9. Chapelle O, Vapnik V, Bousquet O, Mukherjee S (2002) Choosing multiple parameters for support vector machines. Mach Learn 46:131–159
10. Zhang S (2010) KNN-CF approach: Incorporating certainty factor to KNN classification. IEEE Intell Informatics Bull 11:24–33
11. Yousefi S, Narui H, Dayal S, Ermon S, Valaee S (2017) A survey on behavior recognition using WiFi channel state information. IEEE Commun Mag 55:98–104
12. Chen Z, Zhang L, Jiang C, Cao Z, Cui W (2018) WiFi CSI based passive human activity recognition using attention based BLSTM. IEEE Trans Mob Comput 18:2714–2724
13. Foerster F, Smeja M, Fahrenberg J (1999) Detection of posture and motion by accelerometry: a validation study in ambulatory monitoring. Comput Hum Behav 15:571–583
14. Delahoz YS, Labrador MA (2014) Survey on fall detection and fall prevention using wearable and external sensors. Sensors 14:19806–19842
15. Wang H, Zhang D, Wang Y, Ma J, Wang Y, Li S (2016) RT-Fall: A real-time and contactless fall detection system with commodity WiFi devices. IEEE Trans Mobile Comput 16511:26
16. Ibrahim M, Torki M, ElNainay M (2018) CNN based indoor localization using RSS time-series. In: 2018 IEEE symposium on computers and communications (ISCC), pp 01044–01049
17. Albawi S, Mohammed TA, Al-Zawi S (2017) Understanding of a convolutional neural network. In: Proceedings of the 2017 international conference on engineering and technology (ICET), pp 1–6
18. Chung J, Gulcehre C, Cho K, Bengio Y (2015) Gated feedback recurrent neural networks. In: Proceedings of the international conference on machine learning, pp 2067–2075
19. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9:1735–1780
20. Cho K, Van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, Bengio Y (2014) Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint 1406.1078

21. Bengio Y, Lamblin P, Popovici D, Larochelle H (2007) Greedy layer-wise training of deep networks. Adv Neural Inf Process Syst 25:153–160
22. Salakhutdinov R, Hinton G (2009) Deep Boltzmann machines. Artif Intell Stat 2:448–455
23. Freund Y, Haussler D (1992) Unsupervised learning of distributions on binary vectors using two layer networks. Adv Neural Inf Process Syst 9:912–919
24. Hinton GE, Osindero S, Teh YW (2006) A fast learning algorithm for deep belief nets. Neural Comput 18:1527–1554
25. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial nets. Adv Neural Inf Process Syst 7:1039
26. Arulkumaran K, Deisenroth MP, Brundage M, Bharath AA (2017) A brief survey of deep reinforcement learning. arXiv preprint. 1708.05866
27. Chen L, Hoey J, Nugent CD, Cook DJ, Yu Z (2012) Sensor-based activity recognition. IEEE Trans Syst Man Cybern Part C 42:790–808
28. Wang W, Liu AX, Shahzad M, Ling K, Lu S (2017) Device-free human activity recognition using commercial WiFi devices. IEEE J Sel Areas Commun 35:1118–1131
29. Schuster M, Paliwal KK (1997) Bidirectional recurrent neural networks. IEEE Trans Signal Process 45:2673–2681
30. Khan DA, Razak S, Raj B, Singh R (2019) Human behaviour recognition using WiFi channel state information. In: Proceedings of 2019 IEEE international conference on acoustics, speech and signal processing (ICASSP), pp 7625–7629
31. Wold S, Esbensen K, Geladi P (1987) Principal component analysis. Chemom Intell Lab Syst 2:37–52
32. Al-qaness MA (2019) Device-free human micro-activity recognition method using WiFi signals. Geo-spatial Inf Sci 22:128–137
33. Liu J, Wang Y, Chen Y, Yang J, Chen X, Cheng J (2015) Tracking vital signs during sleep leveraging off-the-shelf WiFi. In: Proceedings of the 16th ACM international symposium on mobile ad hoc networking and computing, pp 267–276
34. Joudeh IO, Cretu AM, Wallace RB, Goubran RA, Alkhalid A, Allegue-Martinez M, Knoefel F (2019) WiFi channel state information-based recognition of sitting-down and standing-up activities. In: Proceedings of the 2019 IEEE international symposium on medical measurements and applications (MeMeA), pp 1–6
35. Xiao C, Han D, Ma Y, Qin Z (2019) CsiGAN: robust channel state information-based activity recognition with GANs. IEEE Internet Things J 6:10191–10204
36. Li H, He X, Chen X, Fang Y, Fang Q (2019) Wi-motion: a robust human activity recognition using WiFi signals. IEEE Access 7:153287–153299
37. Yan H, Zhang Y, Wang Y, Xu K (2019) WiAct: a passive WiFi-based human activity recognition system. IEEE Sens J 20:296–305
38. Damodaran N, Haruni E, Kokhkharova M, Schäfer J (2020) Device free human activity and fall recognition using WiFi channel state information (CSI). CCF Trans Pervasive Comput Interact 2:1–7
39. Sheng B, Xiao F, Sha L, Sun L (2020) Deep spatial–temporal model based cross-scene action recognition using commodity WiFi. IEEE Internet Things J 7:3592–3601
40. Moshiri PF, Navidan H, Shahbazian R, Ghorashi SA, Windridge D (2020) Using GAN to enhance the accuracy of Indoor Human Activity Recognition. arXiv preprint. 2004.11228
41. Aldweesh A, Derhab A, Emam AZ (2020) Deep learning approaches for anomaly-based intrusion detection systems: a survey, taxonomy, and open issues. Knowl-Based Syst 189:105124
42. Chung J, Gulcehre C, Cho K, Bengio Y (2014) Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint. 1412.3555
43. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. In: Advances in neural information processing systems, pp 5998–6008
44. Fritsch J, Kuehnl T, Geiger A (2013) A new performance measure and evaluation benchmark for road detection algorithms. In: Proceedings of the 16th international IEEE conference on intelligent transportation systems (ITSC 2013), pp 693–1700
45. Hossin M, Sulaiman MN (2015) A review on evaluation metrics for data classification evaluations. Int J Data Min Knowl Manag Process 5:1

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