Progress In Electromagnetics Research, Vol. 174, 127–141, 2022

A Fast Deep Learning Technique for Wi-Fi-Based Human Activity Recognition

Federico Succetti, Antonello Rosato, Francesco Di Luzio, Andrea Ceschini, and Massimo Panella*

(*Invited*)

Abstract—Despite recent advances, fast and reliable Human Activity Recognition in confined space is still an open problem related to many real-world applications, especially in health and biomedical monitoring. With the ubiquitous presence of Wi-Fi networks, the activity recognition and classification problems can be solved by leveraging some characteristics of the Channel State Information of the 802.11 standard. Given the well-documented advantages of Deep Learning algorithms in solving complex pattern recognition problems, many solutions in Human Activity Recognition domain are taking advantage of those models. To improve the time and precision of activity classification of time-series data stemming from Channel State Information, we propose herein a fast deep neural model encompassing concepts not only from state-of-the-art recurrent neural networks, but also using convolutional operators with added randomization. Results from real data in an experimental environment show promising results.

1. INTRODUCTION

Among the wide set of corollary applications of Wi-Fi network protocols and IEEE 802.11 standards, which are mainly used for local area networking and internet access, there is Human Activity Recognition (HAR), which is quite useful for indoor monitoring of human subjects. While HAR itself is not restricted to Wi-Fi only, recent advances on the analysis of the change of radio waves associated with human actions have been found useful in many real-world applications such as environmental monitoring, health-care assistance, indoor tracking, and behavior analysis. In particular, the use of HAR for elderly people monitoring is an active and valuable field of research.

Tracking human activities and classifying them is a difficult task. In recent years, a plethora of devices such as cameras for closed-circuit television (CCTV) surveillance systems, with added depth and infrared sensors, and portable smart devices have been put in use to carry out the detection, tracking, and identification of human activities. All of these hardware applications bear a consistent effort in terms of intrinsic costs and installation, mainly due to invasive obstruction. The analysis of electromagnetic data stemming from these devices is the subject of several studies, as the transition between human-based tracking and autonomous recognition is made possible by state of the art techniques that use artificial intelligence. In particular, recent advances in Wi-Fi-based machine learning algorithms could become a simplified alternative solution to usability and personal privacy issues. Also, new developments in Wi-Fi technology could empower device-less motion detection algorithms in a way that preserves personal privacy and enables broad, non-invasive HAR solutions.

Received 26 April 2022, Accepted 22 June 2022, Scheduled 5 July 2022
* Corresponding author: Massimo Panella (massimo.panella@uniroma1.it).
The authors are with Department of Information Engineering, Electronics and Telecommunications, University of Rome "La Sapienza", Rome 00184, Italy.

In the 802.11 wireless communication standard, Channel State Information (CSI) contains useful data in the preamble of a Wi-Fi packet. It consists of amplitude attenuation information and phase shift of the transmitted signal along the transmitted path. For the subject matter of this work, CSI data analysis can provide useful insights into surroundings and extraordinary variations during Wi-Fi signal transmission. By using tailored solutions for extracting this information from the network hardware, it is possible to develop a solid alternative to classic HAR monitoring. Since CSI can be essentially retrieved as time-series data, the paradigm of Deep Learning (DL) represents a straightforward approach to the data-driven modeling of HAR systems to be adopted in this context. Namely, Recurrent Neural Networks (RNNs) are one of the most reliable tools to develop HAR using Wi-Fi; in fact, many accurate solutions for time series analysis are based on Deep RNNs [1], where information flows through several neural layers organized in sequential stacks or acyclic graphs. However, while being quite accurate and precise, Deep RNNs are computationally expensive, and they require extensive time and computing resources. This could hinder the premises of HAR, which should be employed in situations where fast detection and reduced computational time are key aspects of the application.

In order to solve the aforementioned problems, in this paper we propose a method for Wi-Fi-based HAR that processes CSI time series by means of a fast Deep Neural Network (DNN). The latter is based on a feed-forward architecture and contains some randomized layers, which avoid the training of their model parameters. In this way, we are able to retain a satisfactory accuracy in identifying the activity of any human subject, while reaching a fast operation in terms of training time for the adopted DNN. Although the focus of the present work is on the acceleration of the training procedure, in the paper we will also prove that the adopted models are very fast during the inference phase, therefore coping with real-time constraints in operative scenarios, so it is not necessary to also consider neural network compression techniques [2] to speed up the inference phase.

The remainder of this paper is organized as follows. The HAR problem and the related solutions based on Wi-Fi data and DL models are introduced in Section 2. The original method proposed in this paper is illustrated in Section 3, while the numerical results related to the experiments for its validation on a real dataset are presented and discussed in Section 4. Finally, our comments and conclusions are drawn in Section 5.

2. HUMAN ACTIVITY RECOGNITION

HAR is a field of study focused on identifying specific actions or movements of people based on sensor data. It extracts information from a series of observations on peoples' motion and on environmental conditions in order to recognize known patterns in such data and to provide relevant information about the users being monitored.

Since its introduction in the 1990's [3], HAR has attracted an increasing number of researchers as a result of its effectiveness in a multitude of real-world applications, such as context awareness, national defense, surveillance, and healthcare [4]. However, there are many conflicting requirements that must be taken into account when a system operating around the human body is designed. One of the main challenges in this context is related to power management, since the attenuation of electromagnetic waves propagating around lossy media is high [5]. In the last years, HAR has been managed following two antithetical approaches: the first one concerning external equipment and the second one characterized by the employment of wearable sensors. External equipment includes various types of devices installed inside the monitored area, operating in autonomous fashion to collect data. Instead, the adoption of wearable sensors requires the person being monitored to wear, hold or be connected to a device with manifold sensors. Depending on the application and experimental settings, both these approaches have advantages and downsides, and their use is strictly related to the underlying task.

By using external sensors, the devices are fixed in predetermined points of interest, and thus, the inference of activities entirely depend on the voluntary interaction of the user with the sensors. They can also be designed to work with multiple users simultaneously, although nothing can be done if the users are out of the reach of the sensors or if they perform activities that do not require an explicit interaction with the equipment. This kind of sensors is particularly suitable for security, intrusion detection, and interactive applications. Instead, in the wearable sensors approach, most of the measured attributes are related to the user's movement, to its interaction with the surrounding environment, or to its

psychological signals. Even if they outdo the aforementioned limitations of external sensors, they are more obtrusive with high-maintenance.

Although the external sensors category includes a vast variety of devices like infrared sensors, thermal sensors, distance sensors, and magnetometers, camera-based sensors are the most common systems used for external sensing for HAR [6]. In general, multiple cameras are positioned in several predetermined locations where the subject of interest performs his activities.

On the other hand, in the wearable sensors approach the monitored subjects require either to be connected to multiple sensors (i.e., smartwatches) or to hold the sensors themselves (i.e., smartphones), with the risk of damaging the device or fully deplete its battery. Furthermore, wearable devices require the approval of the monitored people, they are more invasive, more obtrusive, and can also become an obstacle during the sensing with human body [7].

2.1. Wi-Fi-Based HAR

In the recent years, CSI in Wi-Fi 802.11 has been extensively employed to acquire and collect data pertaining to HAR, which is notoriously a challenging task [8, 9]. The intuition behind a CSI use for HAR is that each user's movement introduces a unique multi-path distortion in Wi-Fi data, which consequently generates specific patterns in the CSI time series. Research on HAR has mainly focused on the use of this technology for several real-time applications, from healthcare to military-related solutions, as well as in the context of smart homes [10–13]. Wi-Fi-based HAR systems can be easily integrated into pre-existing Wi-Fi infrastructures and offer satisfactory performances at reasonable costs compared to other sensing technologies, such as radar and laser-based systems.

The concerns regarding external devices and wearable sensors led to the adoption of Wi-Fi-based approaches in HAR. In this regard, the most commonly employed signal for Wi-Fi indoor localization is the Received Signal Strength (RSS) [14]. In spite of its extensive application, the use of RSS for HAR may lead to unstable and noise-dependent performance. In fact, the information on channel variations provided by the RSS indicator is coarse-grained and cannot deal with small scale fading or multi-path effects caused by micro-movements. CSI signal provides more fine-grained information and overcomes the aforementioned issues; for this reason, it represents a stable alternative to RSS [15]. In particular, the measured CSI retrieved from off-the-shelf Wi-Fi devices is sufficient to build an accurate intrusion detection system under different moving speeds [16].

In 2017, Wang et al. implemented a Wi-Fi-based system that makes use of CSI to detect falls [11]. In addition to the action of falling, walking, sitting down, and standing up were also analyzed. The experiments took place in three different locations: a laboratory, a student's dormitory, and a radio frequency shielding system, where wireless signals are absorbed instead of being reflected. The data collection was carried out through the employment of various Tx-Rx antennas layouts. In order to build a robust CSI model, an anomaly detection algorithm based on local outlier factors was developed to pick out human activities and isolate the corresponding anomaly patterns. Consequently, singular value decomposition was applied to reduce the processing time. Finally, two different algorithms were implemented for the classification step: a Support Vector Machine [17] to detect falls and a Random Forest [18] for the remaining activities. The algorithm achieved an accuracy of 94% with Random Forest classification in all testing scenarios.

A device-Free Crowd Counting (FCC) approach based on CSI was proposed in [19]. The FCC technique demonstrated to be robust against environment variation and showed increased accuracy, scalability, and reliability compared to state-of-the-art solutions. Moreover, a new metric called the "percentage of nonzero elements" (PEM) was introduced to characterize the relationship between the crowd number and various features of CSI.

A research on leveraging CSI signals from off-the-shelf Wi-Fi devices to monitor a person's sleep was proposed in [20]. More precisely, fine-grained CSI is collected around a person to extract rhythmic patterns associated with his respiration, sleeping posture, and body movements. In [21], another CSI-based system was developed to extract and monitor human vital signs like respiration and heart beat rate in a non-clinical setting. Both time and frequency domains were used in the development of the model, achieving state-of-the-art performances when either a single individual or two people are lying in a bed. Also, fine-grained features of CSI available in the 802.11n Wi-Fi protocol is used in [22] to recognize nine different daily human activities with a detection rate of 92%.

2.2. Deep Learning for HAR

DL models are neural networks with multiple layers that have emerged in the last years from the Machine Learning (ML) field. These systems are widely applied in various application domains such as image classification [23], natural language processing [24], speech recognition [25], healthcare [26], and smart grids [27]. They are also imposing as a *de facto* standard for time series processing due to their high generalization capability, long-term sample analysis, and resilience to ill-posed data. The success of this paradigm is confirmed by the continuously increasing body of literature, and it is enabled by the computing power of actual workstations, due to CPUs and GPUs, as well as by the large amount of available data that is used for training DNNs.

DL is typically employed in those situations where humans are unable to exploit their expertise and when the solution of the problem changes over time or needs to be adapted to particular cases. As a matter of fact, DNNs are able to extract meaningful information and learn hidden patterns among data without any preprocessing step, as it was typical, for instance, in early audio applications where data descriptors were determined independently of the statistical model or the neural network adopted to solve a specific task [28].

DNNs use an adaptive combination of feature extraction and classification layers [29, 30]. Thanks to their modularity, several of these models were proposed in the past. Among the most popular ones, Convolutional Neural Networks (CNNs) were introduced in [31], which make use of some layers performing image filtering by convolution, so, they are mainly suited to image processing and object detection tasks. Instead, RNNs were proposed to deal with temporal data through a dynamical architecture where the outputs depend not only on the current inputs but also on the current model's state, which is a memory taking into account the past behavior of the input time series. Long Short-Term Memory (LSTM) networks, introduced in [32], are a special type of RNN which handle long-term dependencies, recalling past information for long periods of time. For this reason, they are particularly suited to solve time series prediction problems.

HAR implementations deal with the identification of some specific activities, and hence, several DL-based systems have been proposed in literature for this purpose. In [33], a plain LSTM was built for HAR; the robustness of the LSTM approach was demonstrated even when experimental conditions deteriorate. Authors in [34] also used a vanilla LSTM to classify seven human activity tasks, reaching remarkable performances above 96% for each of them. Another LSTM-based system for HAR was presented in [35], where an attention-based bidirectional LSTM (BiLSTM) is designed to learn features using known CSI sequences. Since learned features could give different contributions to the final activity recognition task, the attention mechanism is exploited to assign different weights for all of the learned features. The proposed system, named ABiLSTM, reported better detection of human activities than other simpler architectures.

A hybrid convolutional-recurrent architecture was also proposed in [36], where a CNN is employed to perform features extraction, and then, some LSTM layers are used to classify such data. In addition to this, Principal Component Analysis (PCA) is used to eliminate the noise deriving from the objects in the setting of the experiment. Another interesting model was designed in [37], consisting of a CNN and a BiLSTM so as to learn spatial-temporal information from CSI data. In this case, CSI streams are segmented into several series of patches, from which the spatial features are extracted by the convolutional layer. Then, the BiLSTM is employed to further catch temporal dependencies among the processed features. The authors conducted the experiments both with indoor and outdoor data in order to validate the effectiveness of the proposed approach. Finally, a device-free method was presented in [38] for HAR, it uses an ensemble of SVM and LSTM algorithms for classification, where the preprocessing and feature extraction tasks are based on a wavelet analysis.

Four DL models to perform HAR using CSI data considering an environment with six different subjects were evaluated in [4]. The DNNs considered therein are: (i) a CNN with a Gated Recurrent Unit (GRU), which is a simpler version of LSTM; (ii) a CNN with a GRU and an attention mechanism; (iii) a CNN with a GRU and a second CNN; (iv) a CNN with an LSTM and a second CNN. All the networks achieved superior results compared to other recent approaches, in particular for the CNN-GRU model reaching performances above 99% of accuracy for the recognition of several activities such as walking, standing, sitting, and running. Finally, a CSI-based HAR approach considers two different environments in [39]: an office, where only authorized people could enter, and a university hallway, with

students and university employees. An overall accuracy of 91.27% was achieved, in particular with a fall detection accuracy of 96.16% by using the SVM classifier.

3. MATERIALS AND METHODS

Having introduced in the previous Section the general HAR problem and the related works, in the following we illustrate the details of the present study and the techniques used to unravel and solve a specific real-world application. Since CSI monitoring can be assimilated to an elementary collection of timely events, we develop here a solution for HAR employing time series analysis. In particular, we retrieve time series from a set of sensors (e.g., routers) placed in a confined space where subjects are allowed to move and perform the activities to be classified.

3.1. Problem Definition

As the main goal is to recognize the activity of a human subject, let us assume that there are C different activities to be classified by the proposed HAR method. Generally speaking, each activity is a class associated with a semantic or linguistic label belonging to a set $\mathcal{L}$ of C elements, i.e., $\mathcal{L} = \{standing,\ laying,\ \ldots\}$. In order to let mathematical models, among them neural networks, deal with these labels, the latter are usually coded by numerical targets, i.e., $\mathcal{L} = \{1, 2, \ldots, C\}$, or by vectors of binary (class) indicators.

The dataset consists of I observations, where the i-th observation X_i, $i = 1 \ldots I$, is a collection of M different CSI time series having length T. More precisely, $X_i = \{S_{i1}(t), S_{i2}(t), \ldots, S_{iM}(t)\}$, where the time series $S_{im}(t)$, $m = 1 \ldots M$, is the sub-carrier evolution in a chosen Wi-Fi frequency band during the time window $t = 1 \ldots T$. Each element X_i is assigned a label $L_i \in \mathcal{L}$ indicating which one of the C possible activities has been taking place for that observation. It is noted that the length of the time window T can be different among observations, and it is independent of the activity, since it is only relevant to the time the subject spent during the data recording task.

The general purpose of the HAR method presented in this paper is to build a classifier based on a novel DNN architecture, whose model parameters are estimated in a data-driven way by using a suited learning algorithm. The aim is to obtain a good generalization capability in such a way that the trained DNN will be able to correctly predict the activity $\hat{L}$ of any observation X during the inference task, which is for those observations never used during the training process of the DNN. Moreover, the specific goal of the proposed approach is to obtain a fast DNN model to be trained, leveraging the strengths of feed-forward convolutional layers rather than using recurrent architectures.

3.2. Deep Neural Models

Nowadays, most of the literature pertaining to time series analysis is focused on RNNs. Although this is of course a good and reliable paradigm for time series prediction and classification problems, often RNNs have long training and inference times, and they exhibit several issues when dealing with multiple input time series in multivariate/multifactor analysis tasks, as the one previously introduced.

To work out a faster alternative to RNNs for solving this multivariate time series classification problem, we propose a DNN architecture based on feed-forward layers only, which will be referred to in the following as '1D-CNN'. It is shown in Fig. 1, and the layers in the stack are:

- 1-D Convolutional layer: it applies F sliding convolutional filters of dimension 1×1 to each input time series, with no padding.
- Rectified Linear Unit (ReLU) layer: it performs a threshold operation to each element of the input time series, where any value less than zero is set to zero.
- Normalization layer: it normalizes a mini-batch of data across all channels for each time series independently and, at the same time, reduces the sensitivity to network initialization.
- 1-D Global Max Pooling layer: it performs down-sampling by outputting the maximum of the time dimension for each processed time series.

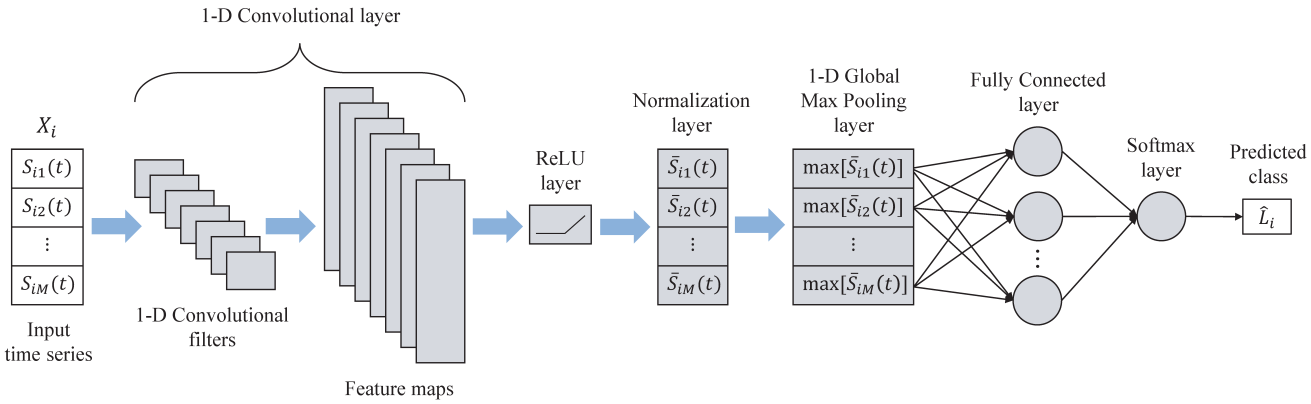

Figure 1. Architecture of the proposed 1D-CNN neural network for time series classification and activity recognition: the observation at the network's input is made of M time series; the predicted class $\hat{L}_i$ is the one scoring the highest probability at the output of the Softmax layer.

- Fully Connected layer: it applies a linear transformation from the M outputs of the previous layer to C outputs, each corresponding to a class.
- Softmax layer: it normalizes the C inputs from the Fully Connected layer in such a way that each output of this layer can be interpreted as a probability that the observation present at the network's input belongs to the corresponding class. The class scoring the highest probability will be the one definitely predicted for the current input.

To efficiently solve the HAR problem using a fast trainable DNN, we also propose in this paper the architecture denoted as the '1D-RCNN', which is identical to the 1D-CNN shown in Fig. 1 with the difference that the 1-D Convolutional layer is randomized, and hence, its weights are not estimated during the training process. In general, randomization in DNNs allows a significant reduction in training time while maintaining a high level of accuracy, since the learning algorithm can operate on a reduced set of weights while losing a (possibly negligible) part of accuracy.

For the sake of benchmarking and assessing the performances of the proposed DNNs, we consider two further models: a hybrid DNN, made up of a 1D convolutional layer plus an LSTM layer and the LSTM-based RNN presented in [34]. They will be denoted in the following as '1D-LSTM' and 'LSTM', respectively. The network's architecture of the 1D-LSTM is reported in Fig. 2 and consists of the following layers:

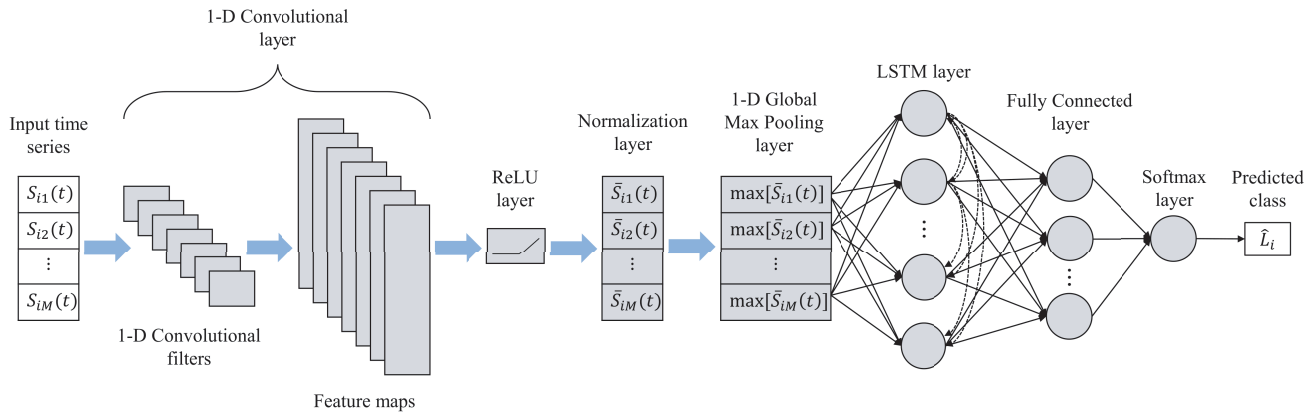

Figure 2. Architecture of the hybrid 1D-LSTM network for time series classification and activity recognition.

- 1-D Convolutional layer: it applies F sliding convolutional filters of dimension 1×1 to each input time series, with no padding.
- Rectified Linear Unit (ReLU) layer: it performs a threshold operation to each element of the input time series, where any value less than zero is set to zero.
- Normalization layer: it normalizes a mini-batch of data across all channels for each time series independently and, at the same time, reduces the sensitivity to network initialization.
- 1-D Global Max Pooling layer: it performs down-sampling by outputting the maximum of the time dimension for each processed time series.
- LSTM layer: it is a standard LSTM layer with H hidden units.
- Fully Connected layer: it maps the H hidden states of the LSTM layer to the C scalar outputs associated with present classes.
- Softmax layer: it normalizes to class probabilities the C inputs from the Fully Connected layer similarly to the previous networks.

The architecture of the LSTM-based RNN is shown in Fig. 3; the layers in the architecture are:

- LSTM layer: it is a standard LSTM layer with H hidden units.
- Dropout layer: it randomly sets input elements to zero with a given probability in order to prevent overfitting.
- Fully Connected layer: it maps the H hidden states of the LSTM layer to the C scalar outputs associated with present classes.
- Softmax layer: it normalizes to class probabilities the C inputs from the Fully Connected layer similarly to the previous networks.

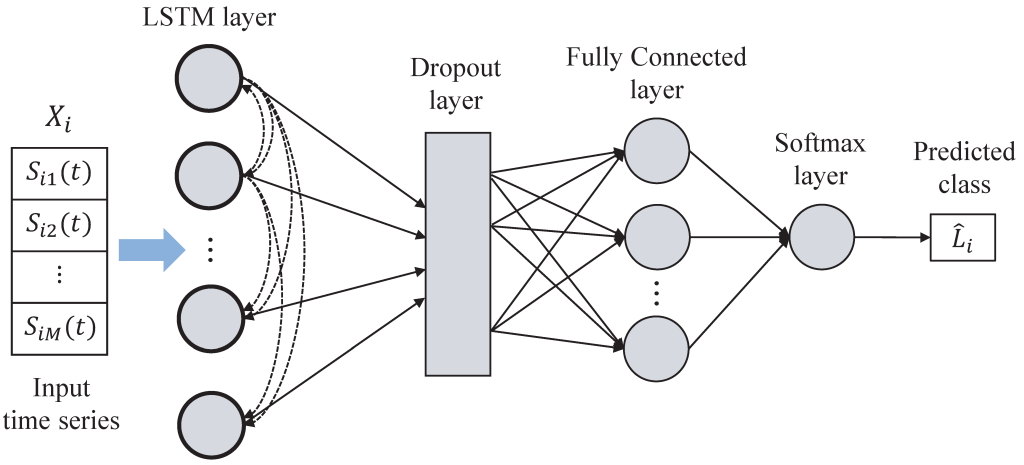

Figure 3. Architecture of the LSTM network proposed in [34] for time series classification and activity recognition.

4. EXPERIMENTS

The HAR classification models considered in this paper are evaluated on a real case study, which represents a typical setup of the experimental analysis carried out in this context. In addition to the aforementioned DNNs, the performance of two classic Machine Learning (ML) models are also evaluated: Random Forest (RF) and Support Vector Machine (SVM).

4.1. Dataset and Experimental Setup

All models are trained on the same dataset retrieved from [40]; the dataset was also used in other research works. In particular, it was built for the single person (the target) and is adopted in the experiment 2 carried out in [34]. The data for the experiment were collected through the use of two routers ASUS RT-AC86U in the frequency band of 80 MHz, with 256 sub-carriers (i.e., $M = 256$). One of the routers is used as an Access Point (AP) while the other one as a CSI data extractor. The data capture scheme is reported in Fig. 4(a).

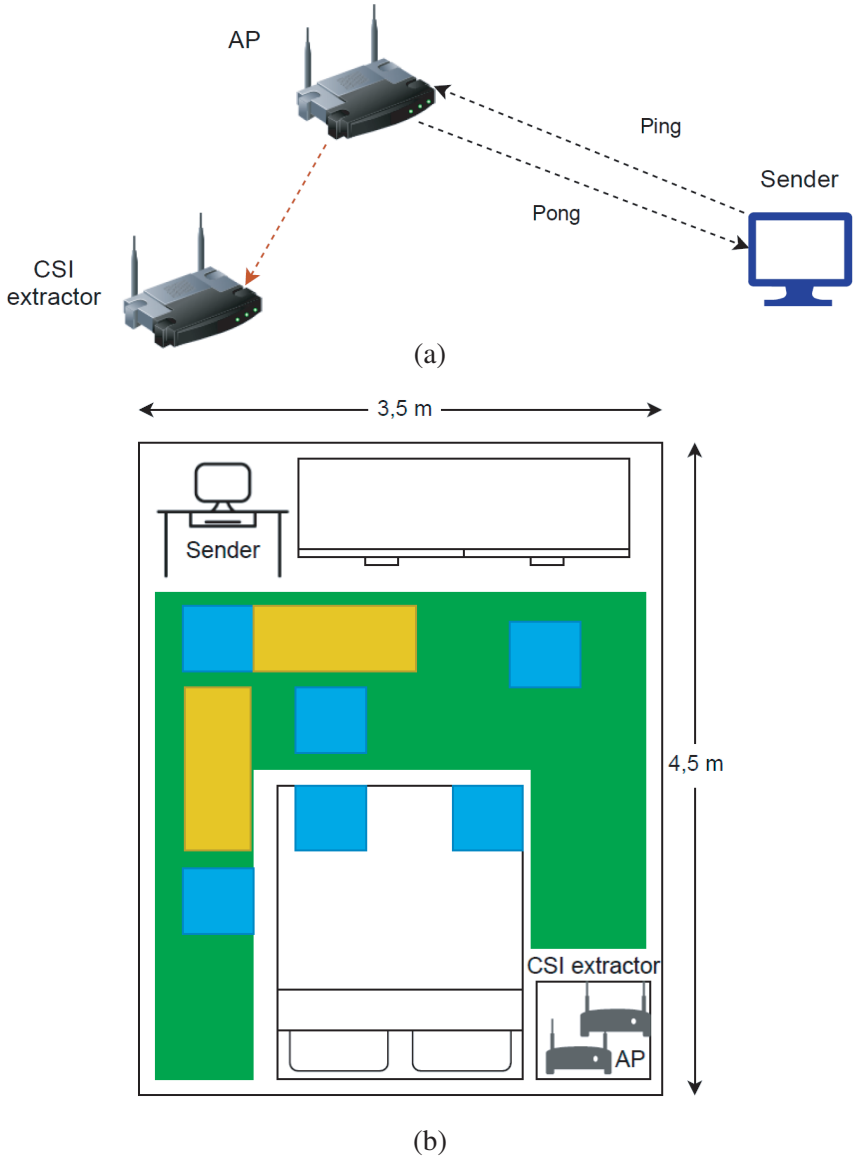

Figure 4. Experimental setup [34]: (a) data capture scheme; (b) physical layout.

It is important to underline that there are several challenges in using the CSI information for the classification of the activities. The first one regards the transition of the activity states and sudden peaks due to abnormal change in human actions that gives poor segmentation to the activities' classification. A Fast Fourier Transform (FFT) was used to transform CSI data for power spectral density and extract the magnitude. Furthermore, in order to denoise the data, Hample filtering and Wavelet signal methods were used. It is also important to underline that the data were collected with Line of Sight (LOS). We refer to [34] for a detailed description in this regard.

The experiment was carried out in a room of size 3.5 m × 4.5 m, as reported in Fig. 4(b). The distance between transmitter and receiver is about 5 m. The colored areas in Fig. 4(b) represent the data collected for the following activities:

- WALK (green);
- SIT, STAND, SIT-DOWN and STAND-UP (blue);
- FALL (yellow).

They represent the set $\mathcal{L}$ of activity labels as well as the $C = 6$ classes to be predicted. An example of normalized CSI data is reported in Fig. 5 for each class. It clearly shows the start of each activity based on the CSI data. The latter differ from each other on the basis of the type of activity that is carried out, as clearly shown in Figs. 5(a)–(f). It is also important to remark that pilot and null sub-carriers were not removed.

Before applying the training procedure, the data are standardized by subtracting the mean and scaling by the standard deviation, thus they are randomly partitioned into training set and test set with a proportion of 80% and 20%, respectively. The training set is in turn divided into the actual training set and a validation set, maintaining the same ratio. This represents the static partition of the dataset. For a more complete and detailed comparison, we also performed a k-fold cross validation, with $k = 10$, on the entire dataset. It is important to underline that, during the training procedure, the actual training data are split into mini-batches, and the sequences are padded so that they have the same length. However, too much padding can have a negative impact on the final network performance. In order to overcome this drawback, the actual training data are sorted by the length of the sequences, and then, a mini-batch size is chosen so that the sequences in a mini-batch have similar length.

It is well known that the outcome of the training procedure and the consequent performance on the test set depend on the random partitions into the adopted training, validation, and test sets as well as on the random initialization of the network parameters. For this reason, 10 different runs are performed, and the average accuracy is reported together with the average training and inference times. This explains the difference in terms of performance between the results obtained in this paper and those obtained by the authors in [34] using the LSTM network. Furthermore, the standard deviation is not reported for the sake of conciseness, as it is quite stable in a confidence interval of ±1% for all models. All the experiments were performed using MATLAB® R2021b on a machine equipped with an AMD Ryzen™ 7 5800X 8-core CPU at 3.80 GHz and with 64 GB of RAM, using for training and inference either the CPU only or also an NVIDIA® GeForce™ RTX 3080 Ti GPU at 1.365 GHz with 12288 MB of GDDR6X RAM.

All the considered DNNs were trained using the Adam algorithm [41] with gradient decay factor 0.9. In order to avoid overfitting, further hyperparameters and training options have been optimized by using an inner grid search procedure on the training set only. This optimization has been carried out for both the proposed 1D-CNN and 1D-RCNN models (apart, in this case, for the 1-D randomized Convolutional layer), the 1D-LSTM model, and both the ML models (i.e., RF and SVM). The final setups of DNNs are reported in Table 1, in which the hyperparameters of the LSTM network have been set as reported in [34]. Considering the RF, a number of 2000 trees and a leaf size of 1 are used, where the leaf size refers to the minimum number of observations per tree leaf. For the SVM, a polynomial kernel function of order 3 and a One-to-One multiclass classification approach are used.

Table 1. Training setup of adopted DNNs and related hyperparameters.

Model	F	H	Initial Learning Rate	Mini-batch Size
1D-CNN	50	-	0.007	128
1D-RCNN	50	-	0.007	128
1D-LSTM	50	50	0.001	64
LSTM	-	100	0.001	32

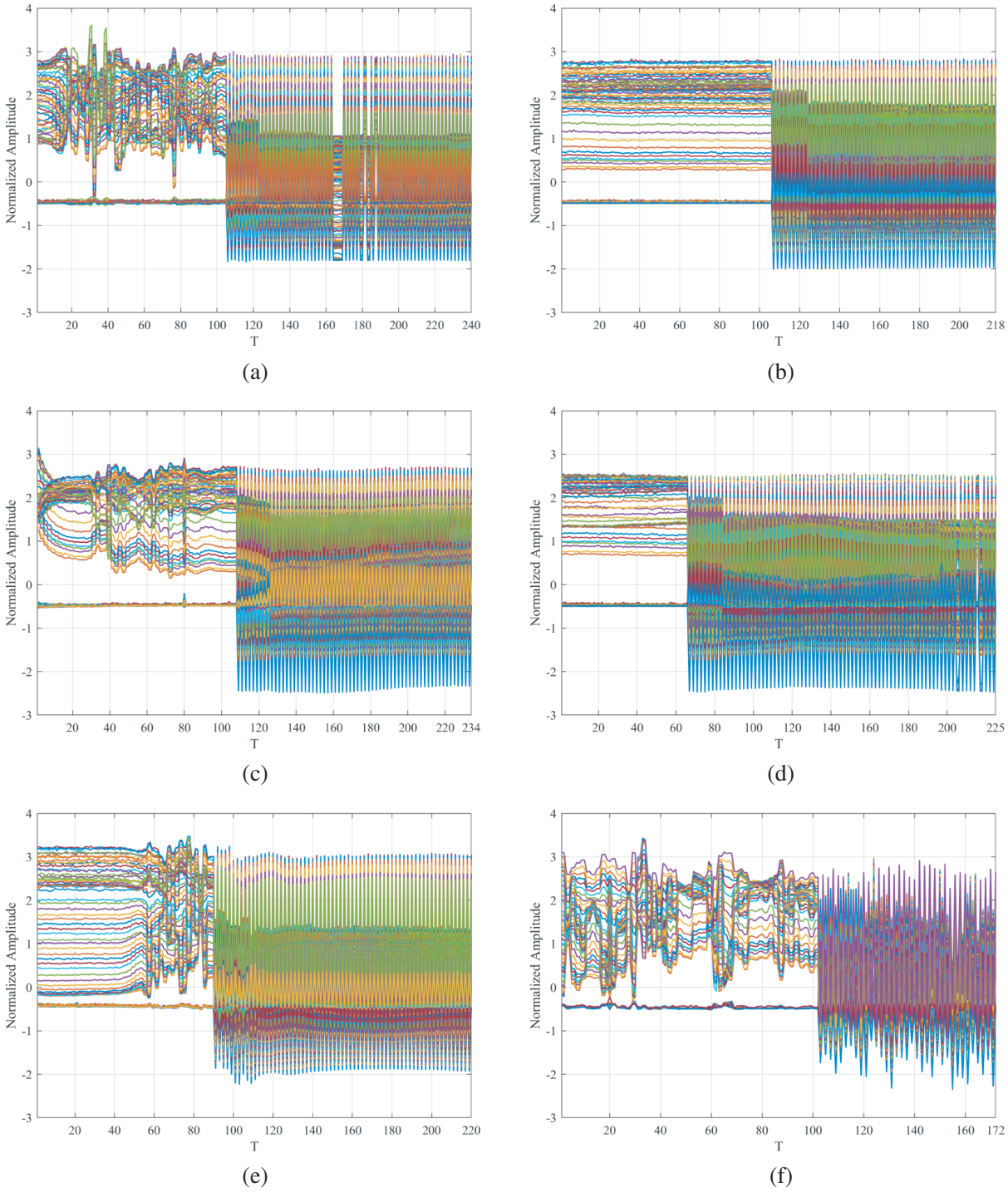

Figure 5. Some examples of normalized CSI data. (a) FALL. (b) SIT. (c) SIT-DOWN. (d) STAND. (e) STAND-UP. (f) WALK.

4.2. Numerical results

The experimental results considering all models are reported in Table 2, where the average accuracies for each class and for all classes are shown. The latter is reported twice: the first one (item 'ALL') refers to the average accuracy for all classes obtained with the static partition of the dataset, while the

second one (item 'CV') is the average accuracy for all classes obtained using the k-fold cross-validation procedure. It is clear from Table 2 that considering only the DNNs, the average accuracies are quite similar in the two cases. It is also evident how the accuracy among the classes varies according to the specific model. In particular, this difference is quite relevant for the 1D-RCNN model, where it goes from an average value of 68.0% ('SIT') to 97.6% ('STAND-UP'), and the RF model, where it varies from 57.2% ('STAND') to 84.4% ('STAND-UP').

Table 2. Average accuracy (%) per class and for all classes, considering all models.

Model	FALL	SIT	SIT-DOWN	STAND	STAND-UP	WALK	ALL	CV
1D-CNN	94.0	98.0	90.0	100.0	95.2	100.0	96.2	94.8
1D-RCNN	89.6	68.0	77.6	92.4	97.6	96.4	86.9	86.8
1D-LSTM	94.8	98.0	93.6	100.0	99.6	99.6	97.6	97.0
LSTM	98.0	89.2	91.6	98.8	93.2	91.2	93.7	92.4
RF	67.6	61.2	62.8	57.2	84.4	52.4	64.3	63.7
SVM	36.0	36.0	32.0	40.0	32.0	32.0	34.7	34.4

Considering the 1D-RCNN, this variation is mostly due to the randomization process that, in this specific case, affects the accuracy in a non-negligible way. In fact, it shows the worst performance on average among DNNs. The difference in terms of accuracy among the classes is also present considering the other DNN models, albeit to reduced extent thanks to the full training procedure. Nevertheless, a possible reason that this happens can be because the classes are not perfectly balanced. A separate discussion must be made for ML models. They both achieve lower accuracies, especially the SVM, compared to DNNs. This is because these models are not able to automatically extract the relevant features from the raw data. Another important thing to point out is that the standard deviation of $\pm 1\%$ mentioned in Section 4.1 refers to the average accuracy considering 'ALL' and 'CV' classes.

The relative difference in terms of accuracy among the models for each class is reported in Table 3, where zero values highlight the best model for each class. Here, it is clear that the performances of the proposed 1D-CNN are better than those of LSTM on average, and at the same time, they are comparable with the ones of 1D-LSTM that is the best one in terms of accuracy. Furthermore, these models show only little variations in accuracy for each class, thus proving their robustness. This is not true for the 1D-RCNN model, which shows a higher difference in terms of accuracy with respect to the other DNN models, depending on the specific class. The same goes for the ML models, due to the low accuracy achieved by their usage.

Table 3. Relative difference (%) in accuracy among all models for each class.

Model	FALL	SIT	SIT-DOWN	STAND	STAND-UP	WALK	ALL	CV
1D-CNN	4.1	0.0	3.9	0.0	4.4	0.0	1.4	2.3
1D-RCNN	8.6	30.6	17.1	7.6	3.0	3.6	11.0	10.5
1D-LSTM	3.3	0.0	0.0	0.0	0.0	0.4	0.0	0.0
LSTM	0.0	9.0	2.1	1.2	6.4	8.8	4.0	4.7
RF	31.0	37.6	32.9	42.8	15.3	47.6	34.1	34.3
SVM	63.3	63.3	65.8	60.0	67.9	68.0	64.5	64.5

Considering the main goal of this paper, the most important results are related to the average training and inference times necessary to estimate the model parameters of each network. They are reported in Table 4 considering different hardware (HW) resources and both static and cross-validation partitions. It is important to highlight that the training times are computed considering the entire training process over all the 10 runs, while the inference times are related to the single sample, i.e., the single time series. As the inference time is independent of the adopted training procedure, the average

inference time per sample across both cases is reported. Another important thing to point out is that ML models are not optimized on GPU as usual for DNN training algorithms, and hence, in order to make a fair comparison, we included also the computational times of DNNs by using the CPU only.

As a general remark, a fast training procedure could become a crucial parameter in real-time operations. The values reported in Table 4 highlight several interesting properties of the considered models and confirm the main hypotheses about the proposed approach. For instance, the 1D-RCNN network turns out to be the fastest one to be trained on both the HW systems thanks to the randomization process. Nevertheless, the speed-up of the training procedure is achieved at the expense of accuracy. On the contrary, the LSTM network shows better accuracy than the 1D-RCNN, but at the same time, it is the slowest one among the DNNs on both HW platforms. This is a well-known disadvantage for any DNN based on recurrent layers, as the training algorithm relies on backpropagation through time (BPTT) routines and deals with a larger number of model parameters to be estimated. The ML models are the slowest ones and the worst ones in terms of training times and accuracy. As expected, by using 10 folds with the cross-validation procedure, the whole training speed is about 10 times slower than the one obtained considering the static partition of the dataset. Regarding the inference times, they are all very small apart from the RF model that relies on branched classification

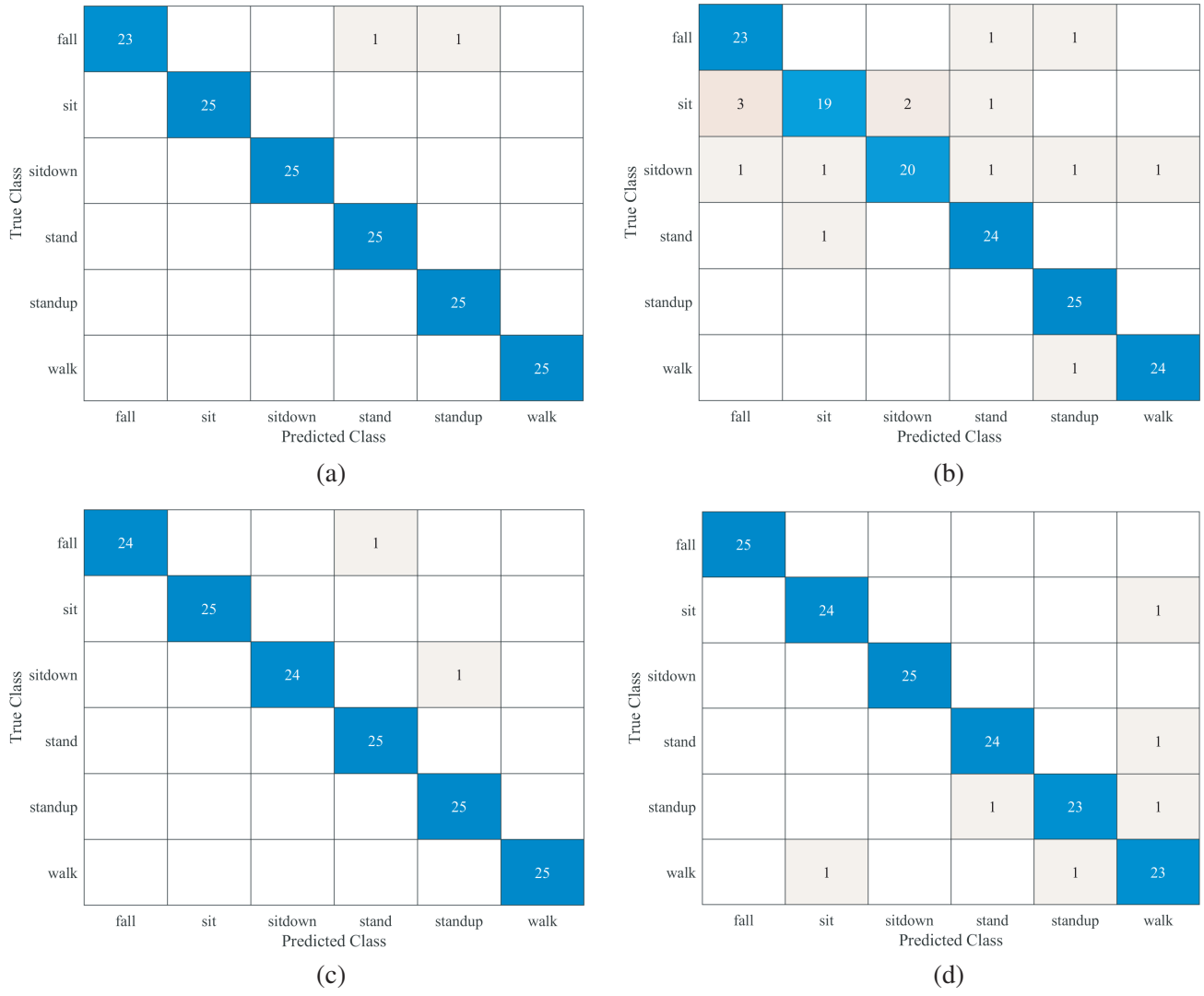

Figure 6. Average confusion matrices (rounded values) for all the considered DNN models. (a) 1D-CNN network. (b) 1D-RCNN network. (c) 1D-LSTM network. (d) LSTM network.

Table 4. Training and inference times (s) of all models.

HW	Model	Training time (ALL)	Training time (CV)	Inference time (CV/ALL)
GPU	1D-CNN	$2.36 \cdot 10^2$	$2.12 \cdot 10^3$	$3.16 \cdot 10^{-6}$
	1D-RCNN	$2.11 \cdot 10^2$	$2.10 \cdot 10^3$	$3.17 \cdot 10^{-6}$
	1D-LSTM	$2.84 \cdot 10^2$	$3.01 \cdot 10^3$	$3.64 \cdot 10^{-6}$
	LSTM	$4.30 \cdot 10^2$	$4.42 \cdot 10^3$	$3.56 \cdot 10^{-6}$
CPU	1D-CNN	$3.38 \cdot 10^2$	$3.37 \cdot 10^3$	$4.65 \cdot 10^{-6}$
	1D-RCNN	$2.71 \cdot 10^2$	$2.69 \cdot 10^3$	$4.09 \cdot 10^{-6}$
	1D-LSTM	$3.62 \cdot 10^2$	$3.84 \cdot 10^3$	$1.27 \cdot 10^{-5}$
	LSTM	$2.60 \cdot 10^3$	$2.77 \cdot 10^4$	$5.67 \cdot 10^{-6}$
	RF	$5.87 \cdot 10^3$	$5.81 \cdot 10^4$	$1.06 \cdot 10^{-3}$
	SVM	$8.79 \cdot 10^3$	$8.33 \cdot 10^4$	$3.60 \cdot 10^{-5}$

trees. Anyway, such times assure the application of the proposed approach in real-time operation cases, even using cheap hardware and slower CPUs.

Overall, the DNN model showing the best trade-off between accuracy and training time is the proposed 1D-CNN. It is slightly slower than the 1D-RCNN network, but it reaches comparable performances to the LSTM and 1D-LSTM, as reported in Table 2, by obtaining significantly shorter training times. For a further analysis, the average confusion matrices obtained by each DNN model are reported in Fig. 6. As expected, the 1D-RCNN presents the highest classification error, while the other models show better performances.

5. CONCLUSION

Nowadays, Wi-Fi devices supporting 802.11 wireless communication standard are commonly widespread, and the CSI signals can provide meaningful information on the surroundings; therefore, their use for HAR is straightforward. In this context, DL and DNNs stand as powerful tools for CSI time series analysis, although they suffer from large computational burdens. The innovative contribution of this work lies in proposing a 1D-CNN architecture for a Wi-Fi-based HAR approach. Our solution exploits a one-dimensional convolutional layer in order to efficiently extract relevant features from input CSI signals and to process them throughout the network.

In particular, the 1D-CNN model and its randomized variant 1D-RCNN are tested on a classification problem of six different human activities performed in a confined environment. The experimental results on a well-known dataset demonstrate the validity of the proposed approach compared to previous techniques based on recurrent LSTM networks [34]. In fact, our 1D-CNN model achieves remarkable results in all the six classes related to human's motion: it reaches an overall average accuracy of about 96%, which is 2% higher than that of the LSTM model. The 1D-CNN network also ensures a significantly faster training procedure, becoming the recommended choice in the case of real-time operations. The achieved results with the randomized fashion of the proposed architecture demonstrate a poorer accuracy performance, partially balanced by a reduction in the necessary training time.

The good performances achieved by this model can pave the way for the development of more efficient and fast feed-forward DNNs. For instance, complex scenarios pertaining to experimental setups, in which two or more targets are present, and low-cost embedded systems for HAR processing in real-time can be considered.

REFERENCES

1. Tian, Y., S. Li, C. Chen, Q. Zhang, C, Zhuang, and X. Ding, "Small CSI samples-based activity recognition: A deep learning approach using multidimensional features," *Security and Communication Networks*, Vol. 2021, 5632298, 2021.
2. O'Neill, J., "An overview of neural network compression," 2020.
3. Foerster, F., M. Smeja, and J. Fahrenberg, "Detection of posture and motion by accelerometry: A validation study in ambulatory monitoring," *Computers in Human Behavior*, Vol. 15, No. 5, 571–583, 1999.
4. Shalaby, E., N. ElShennawy, and A. Sarhan, "Utilizing deep learning models in CSI-based human activity recognition," *Neural Computing and Applications*, 1–18, 2022.
5. Golestani, N. and M. Moghaddam, "Magnetic induction-based human activity recognition (MI-HAR)," *2019 IEEE International Symposium on Antennas and Propagation and USNC-URSI Radio Science Meeting*, IEEE, 2019.
6. Delahoz, Y. S. and M. A. Labrador, "Survey on fall detection and fall prevention using wearable and external sensors," *Sensors*, Vol. 14, No. 10, 19806–19842, 2014.
7. Politi, O., I. Mporas, and V. Megalooikonomou, 'Human motion detection in daily activity tasks using wearable sensors," *2014 22nd European Signal Processing Conference (EUSIPCO)*, IEEE, 2014.
8. Jobanputra, C., J. Bavishi, and N. Doshi, "Human activity recognition: A survey," *Procedia Computer Science*, Vol. 155, 698–703, 2019.
9. Vrigkas, M., C. Nikou, and I. A. Kakadiaris, "A review of human activity recognition methods," *Frontiers in Robotics and AI*, Vol. 2, 2015.
10. Wang, H., D. Zhang, Y. Wang, J. Ma, Y. Wang, and S. Li, "RT-Fall: A real-time and contactless fall detection system with commodity WiFi devices," *IEEE Transactions on Mobile Computing*, Vol. 16, No. 2, 511–526, 2016.
11. Wang, Y., K. Wu, and L. M. Ni, "Wifall: Device-free fall detection by wireless networks," *IEEE Transactions on Mobile Computing*, Vol. 16, No. 2, 581–594, 2016.
12. Murad, A. and J.-Y. Pyun, "Deep recurrent neural networks for human activity recognition," *Sensors*, Vol. 17, No. 11, 2017.
13. Moshiri, F., R. Shahbazian, M. Nabati, and S. A. Ghorashi, "A CSI-based human activity recognition using deep learning," *Sensors*, Vol. 21, No. 21, 2021.
14. Ibrahim, M., M. Torki, and M. ElNainay, "CNN based indoor localization using RSS time-series," *2018 IEEE Symposium on Computers and Communications (ISCC)*, IEEE, 2018.
15. Halperin, D., W. Hu, A. Sheth, and D. Wetherall, "Tool release: Gathering 802.11 n traces with channel state information," *ACM SIGCOMM Computer Communication Review*, Vol. 41, No. 1, 53–53, 2011.
16. Lv, J., W. Yang, X. Du, and M. Yu, "Robust WLAN-based indoor intrusion detection using PHY layer information," *IEEE Access*, Vol. 6, 30117–30127, 2017.
17. Cortes, C. and V. Vapnik, "Support-vector networks," *Machine Learning*, Vol. 20, 273–297, 1995.
18. Breiman, L., "Random forests," *Machine Learning*, Vol. 45, 5–32, 2001.
19. Xi, W., J. Zhao, X.-Y. Li, K. Zhao, S. Tang, X. Liu, and Z. Jiang, "Electronic frog eye: Counting crowd using WiFi," *IEEE INFOCOM 2014 — IEEE Conference on Computer Communications*, IEEE, 2014.
20. Liu, X., J. Cao, S. Tang, and J. Wen, "Wi-Sleep: Contactless sleep monitoring via WiFi signals," *2014 IEEE Real-Time Systems Symposium*, IEEE, 2014.
21. Liu, J., Y. Wang, Y. Chen, J. Yang, X. Chen, and J. Cheng, "Tracking vital signs during sleep leveraging off-the-shelf WiFi," *Proceedings of the 16th ACM International Symposium on Mobile ad hoc Networking and Computing*, 2015.

22. Wang, Y., J. Liu, Y. Chen, M. Gruteser, J. Yang, and H. Liu, "E-eyes: Device-free location-oriented activity identification using fine-grained WiFi signatures," *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking*, 2014.
23. He, Z., "Deep learning in image classification: A survey report," *2020 2nd International Conference on Information Technology and Computer Application (ITCA)*, IEEE, 2020.
24. Otter, D. W., J. R. Medina, and J. K. Kalita, "A survey of the usages of deep learning for natural language processing," *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 32, No. 2, 604–624, 2020.
25. Kumar, A., S. Verma, and H. Mangla, "A survey of deep learning techniques in speech recognition," *2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*, IEEE, 2018.
26. Esteva, A., A. Robicquet, B. Ramsundar, V. Kuleshov, M. DePristo, K. Chou, C. Cui, G. Corrado, S. Thrun, and J. Dean, "A guide to deep learning in healthcare," *Nature Medicine*, Vol. 25, No. 1, 24–29, 2019.
27. Rosato, A., M. Panella, A. Andreotti, A. M. Osama, and R. Araneo, "Two-stage dynamic management in energy communities using a decision system based on elastic net regularization," *Applied Energy*, 291, 2021.
28. Rizzi, A., N. M. Buccino, M. Panella, and A. Uncini, "Genre classification of compressed audio data," *2008 IEEE 10th Workshop on Multimedia Signal Processing*, 654–659, IEEE, 2008.
29. Erhan, D., P.-A. Manzagol, Y. Bengio, S. Bengio, and P. Vincent, "The difficulty of training deep architectures and the effect of unsupervised pre-training," *Artificial Intelligence and Statistics*, PMLR, 2009.
30. Vincent, P., H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," *Proceedings of the 25th International Conference on Machine Learning*, 2008.
31. Fukushima, K., "Neocognitron: A hierarchical neural network capable of visual pattern recognition," *Neural networks*, Vol. 1, No. 2, 119–130, 1988.
32. Hochreiter, S. and J. Schmidhuber, "Long short-term memory," *Neural Computation*, Vol. 9, No. 8, 1735–1780, 1997.
33. Grushin, A., D. D. Monner, J. A. Reggia, and A. Mishra, "Robust human action recognition via long short-term memory," *The 2013 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2013.
34. Schäfer, B., B. R. Barrsiwal, M. Kokhkharova, H. Adil, and J. Liebehenschel, "Human activity recognition using CSI information with nexmon," *Appl. Sci.*, Vol. 11, 8860, 2021.
35. Chen, Z., L. Zhang, C. Jiang, Z. Cao, and W. Cui, "WiFi CSI based passive human activity recognition using attention based BiLSTM," *IEEE Transactions on Mobile Computing*, Vol. 18, No. 11, 2714–2724, 2018.
36. Khan, D. A., S. Razak, B. Raj, and R. Singh, "Human behaviour recognition using WiFi channel state information," *ICASSP 2019 — 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2019.
37. Sheng, B., F. Xiao, L. Sha, and L. Sun, "Deep spatial-temporal model based cross-scene action recognition using commodity WiFi," *IEEE Internet of Things Journal*, Vol. 7, No. 4, 3592–3601, 2020.
38. Damodaran, N., E. Haruni, M. Kokhkharova, and J. Schäfer, "Device free human activity and fall recognition using WiFi Channel State Information (CSI)," *CCF Transactions on Pervasive Computing and Interaction*, Vol. 2, No. 1, 1–17, 2020.
39. Alsaify, B. A., M. M. Almazari, R. Alazrai, S. Alouneh, and M. I. Daoud, "A CSI-based multi-environment human activity recognition framework," *Appl. Sci.*, Vol. 12, 930, 2022.
40. Schäfer, J., "CSI human activity," *IEEE Dataport*, August 2, 2021.
41. Kingma, D. and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations*, 2014.