



Device free human activity and fall recognition using WiFi channel state information (CSI)

Neena Damodaran¹ · Elis Haruni¹ · Muyassar Kokkhkharova¹ · Jörg Schäfer¹

Received: 16 October 2019 / Accepted: 4 January 2020 / Published online: 30 January 2020
© The Author(s) 2020

Abstract

Human activity recognition (HAR) is a broad research area. While there exist solutions based on sensors and vision-based technologies, these solutions suffer from considerable limitations. Thus in order to mitigate or avoid these limitations, device free solutions based on radio signals like (home) WiFi, in particular 802.11 are considered. Recently, channel state information (CSI), available in WiFi 802.11n networks have been proposed for fine-grained analysis. We are able to detect human activities like WALK, SIT, STAND, RUN (in the sequel, any human activity used for classification is capitalised, i.e. is denoted by its corresponding label. For example, “standing“ is denoted as STAND, the activity “sitting“ is denoted by SIT and so on), etc. in a line-of-sight (LOS) scenario and a non-line-of-sight (N-LOS) scenario within an indoor environment. We propose two algorithms—one using a support vector machine (SVM) for classification and another one using a long short-term memory (LSTM) recurrent neural network. While the former uses sophisticated pre-processing and feature extraction techniques based on wavelet analysis, the latter processes the raw data directly (after denoising). We show that it is possible to characterize activities and/or human body presence with high accuracy and we compare both approaches with regard to accuracy and performance. Furthermore, we extend the experimental setup to detect human falls, too which is a relevant use-case in the context of ambient assisted living (AAL) and show that with the developed algorithms it is possible to detect falls with high accuracy. In addition, we also show that the algorithms can be used to count the number of people in a room based on the CSI-data, which is a first step towards detecting more complex social behavior and activities. Our paper is an extended version of the paper (Damodaran and Schäfer, Device free human activity recognition using wifi channel state information, in: 16th IEEE International Conference on Ubiquitous Intelligence and Computing (UIC 2019), 5th IEEE Smart World Congress, Leicester, vol 16, IEEE, 2019).

Keywords Activity recognition · Ambient assisted living (AAL) · Human activity recognition (HAR) · Channel state information (CSI) · Fingerprinting · Localization · Machine learning · Neural networks · Object detection · Passive radar · Passive (microwave) remote sensing · Recurrent neural networks · Remote monitoring · Wireless networks

Mathematics Subject Classification 68T05 · 68T10 · 62H30

As mentioned in the abstract, this article is an extended version of Damodaran and Schäfer (2019). Henceforth, the material presented in Sects. 1–5 is an (almost) nearly verbatim adaptation, but more detailed version of this publication. The material in Sects. 6–8 contains new and original results.

✉ Jörg Schäfer
jschaefer@fb2.fra-uas.de

Neena Damodaran
neena.damodaran@gmail.com

Elis Haruni
haruni@stud.fra-uas.de

1 Introduction

While human activity recognition (HAR) is an interesting fundamental research problem on its own, it is also becoming increasingly important in areas such as health care of

Muyassar Kokkhkharova
kokkhkharovam@gmail.com

¹ Faculty of Computer Science and Engineering, Frankfurt University of Applied Sciences, Nibelungenplatz 1, 60318 Frankfurt am Main, Germany

elderly and sick or otherwise impaired people. Due to demographic trends, there is a tremendous increase in the elderly population and while some elderly suffer from the loss of cognitive or physical autonomy, they choose to live independently at their residence instead of living under the care of a hospital. This raises safety and security concerns. Monitoring of human activity and fall detection systems might mitigate some of the risks. Also monitoring day to day activity would give health personal a better insight into the lifestyle of their patients and it would allow them to assist them in a more informed manner to maintain good health and ensure quick recovery (Tan et al. 2018). Human activity recognition is also a key component in context aware computing, for energy efficient smart homes, fitness tracking and many internet of things (IoT) based solutions (Yousefi et al. 2017; Wang et al. 2015a), and in the context of disaster recovery cases (Scheurer et al. 2017). For the relevance of fall as a specific activity and the importance of fall detection, we refer to Sect. 6.

A lot of research has gone into the sensor-based and visual-based solutions for this purpose, but their obtrusive nature has made their use limited and cumbersome in a residential environment. Wearable sensor based solutions may not monitor the activity accurately or correctly because old people might forget to wear the device(s) or simply find it too cumbersome or inconvenient to wear it all the time. This would lead to inaccurate data. Visual based solution will only function in scenarios where the subject is in the line-of-sight (LOS). They also require good lighting and are intrusive as they impact the privacy of the individual. To overcome these limitations, the research community has started to investigate device-free sensing technologies. In these type of sensing technologies, radio signals such as WiFi signals are utilized to track human motion and activities.

Radio frequency (RF) based device free sensing has the advantage of being non-intrusive. RF-based approaches include ultra-wide band, continuous-wave radar, Zigbee, WiFi and other technologies. Out of all the approaches WiFi based solutions are gaining more attention from the research community (Ma et al. 2016). Basically, these solutions require a WiFi access point and WiFi enabled devices (laptops, tablets, mobile phones etc.) at various locations in the indoor environment. WiFi setup is available easily in mostly all the indoor or residential environments today and therefore no additional setup cost is incurred. WiFi signals can travel through the wall so it is not necessary for the person to be in the line-of-sight (LOS). The technologies based on WiFi are based on the fact that radio signals are affected by human movement. The estimated wireless channel will have a different amplitude and phase because the movement of human and objects changes the multipath characteristics of the channel.

Recently channel state information from the WiFi network interface cards (NIC) (Al-Qaness et al. 2017) has gained a lot of attention. Unlike received signal strength indication (RSSI), CSI is measured from radio links per orthogonal frequency division multiplexing (OFDM) subcarriers for each received packet (Al-Qaness et al. 2016, 2017; Cheng and Chang 2017). RSSI provides coarse grained MAC layer information whereas CSI provides a fine grained, PHY layer information such as subcarriers and amplitude/phase information for each subcarrier (Al-Qaness et al. 2016, 2017; Cheng and Chang 2017). Therefore, CSI seems to be an attractive candidate delivering sensor information to be used as an input for HAR.

2 Related works

2.1 Prior work

One of the first prominent research results on WiFi based sensing for user location and tracking was achieved by Bahl and Padmanabhan (2000). In this research WiFi RSSI was used for (indoor) localization. Since then WiFi RSSI information has been used in localization (Ma et al. 2016; Schäfer 2014; Aversente et al. 2016) for human activity recognition (Wang et al. 2015a, b; Al-Qaness et al. 2016), and for gesture recognition (Pu et al. 2013). RSSI is a very simple metric and does not require any special hardware changes neither at the access point end nor at the mobile end. Using RSSI for human activity recognition is very easy but RSSI suffers from multipath fading, severe distortions and instability in a complex environment (Al-Qaness et al. 2016, 2017; Cheng and Chang 2017). RSSI is a coarse-grained information and it does not leverage the subcarriers of an OFDM channel (Wang et al. 2016b).

Pu et al. (2013) proposed a novel gesture recognition system called WiSee which leverages WiFi. This method requires modified WiFi hardware which incorporates WiFi USRP-N210 software defined radio (SDR) system. There are other systems like Wi-Vi (Adib and Katabi 2013) and WiTrack (Adib et al. 2014) which are built on a similar platform. WiTrack is used for 3D tracking of a user. All these systems are based on the measurement of doppler shift in OFDM signals, caused by movements of the human body.

In 2011 Halperin et al. (2011) released a tool that measures WiFi channel information especially CSI according to 802.11n standard. This tool enables the usage of CSI data for a specific Intel chip set. For an open source alternative based on Atheros chip sets, we refer to Xie et al. (2015) and Tsakalaki and Schäfer (2018). Recently, other chip sets have been reverse engineered (Schulz 2018) to support—among other things—research on CSI.

A lot of research has happened since then in the area of localization (Kotaru et al. 2015; Tsakalaki and Schäfer 2018), and HAR using CSI based on commercial Wi-Fi devices. In recent years, many articles have emerged that have used CSI for detecting human activity, for a comprehensive survey, see (Wang et al. 2017c). In particular, we refer to Wi-Hear (Wang et al. 2016a), Wi-Eyes (Wang et al. 2014), CARM (Wang et al. 2015a), for gesture recognition to (Pu et al. 2013; Al-Qaness et al. 2017), and for fall to RT-Fall (Wang et al. 2017a) and WiFall (Han et al. 2014). In Xin et al. (2018) the phase differences between waveforms of multiple antennas are used to detect human activities quite generically. WiFall e.g. is a fall detection system focusing on a one-class classification (Fall) using an anomaly detection (least outlier factor) based approach to retrieve the activity's pattern segment, whereas in our work we address multiclass problems. Furthermore, in our work we use a single MIMO system rather than multiple ones (three in case of WiFall). RT-Fall is another fall detection system focusing on a one-class classification exclusively and exploiting the sharp power profile decline associated with fall and fall-like activities as opposed to non-fall like activities. Our approach differs from these two as we employ different classification algorithms and compare the classification using wavelets plus support vector machine (SVM) vs. long short-term memory (LSTM).

Please note, that alternative device free approaches using other information but RF such as sound e.g. (Xu et al. 2019) are not considered in this paper. Also, with the exception of fall detection we are considering only basic, atomic activities, and in this paper we deliberately ignore the context of the activities i.e. we ignore contextual information such as spatial information, time or any personal patterns in detecting the data and reserve this to future work. For a recent example of using contextual information for activity detection, see e.g. (Chen et al. 2019).

2.2 Our contribution

Our contribution is twofold. First, we combine discrete wavelet transform (DWT), principal component analysis (PCA), power spectral density (PSD) and frequency of center of energy and Haart wavelet analysis to extract the lower frequency bins (Sect. 4) into a unique algorithm (Algorithm 1) using a support vector machine as a classifier and show that through these pre-processing techniques, we yield strong classification results. In fact, the accuracy achieved is better than in previous reported work (Wang et al. 2017a; Han et al. 2014), despite the fact that we classify multiple activities, see Tables 1, 2, 3 and 4 for quantitative results. In addition, we define a second algorithm (Algorithm 2) using LSTM that operates directly on the raw data and uses only denoising via DWT as the underlying preprocessing technique. We

compare both results, see Tables 1, 2, 3 and 4. We show that LSTM together with this light preprocessing is almost on a par with the more sophisticated former algorithm considering classification performance.

3 Experimental setup

3.1 Hardware

In our experiments, we use Intel WiFi Link (IWL) 5300 Network Interface Card (NIC). IWL 5300 supports 802.11n standard and hence makes it possible to record channel state information. There are 64 subcarriers in 20 MHz channel and 128 subcarriers in 40 MHz channel. Irrespective of the width of the channel, the subcarriers are grouped in 30 subcarrier groups. The number of indexed subcarriers that would be represented by a group is based on the width of the channel. For 20 MHz channel a subcarrier group represents 2 physical subcarriers and for 40 MHz channel a subcarrier group represents 4 physical subcarriers. Channel state information is reported in the form of 30 matrices, where each matrix represents a subcarrier group.

For our experiments, we have used two Lenovo laptops, that are equipped with IWL 5300 NIC. The operating system installed on each of the laptop is 64 Bit Ubuntu version 14.04 LTS. The kernel version is 4.2.0-42. In order to obtain channel state information from the NIC, the existing kernel has to point to a modified wireless driver and the existing IWL 5300 firmware has to be replaced with a modified firmware as IWL 5300 firmware does not allow direct access to the NIC's memory to read CSI. By using the modified wireless driver and modified firmware, the debug mode of IWL 5300 can be enabled. These modifications cause the NIC to report the CSI to main memory. Halperin et al. (2011) proposed the “Linux 802.11n CSI Tool” and all the instruction to modify the firmware is provided as part of the installation instruction.

3.2 Apartment

Data collection was conducted in the living room and hallway of an apartment depicted in Fig. 1a, b. In each figure, the position of the transmitter and receiver is indicated. All activities took place in the living room. Samples had to be collected for line-of-sight (LOS) and non-line-of-sight (N-LOS). Note that for the N-LOS the transmitter was moved into the hallway. For the LOS scenario, the transmitter and the receiver were both placed in the living room. For the N-LOS scenario, the transmitter was placed in the hallway and the receiver was left in the living room.

Table 1 Comparison of normalized confusion matrices for five activities WALK, RUN, SIT, STAND, and EMPTY for (a) SVM LOS, (b) SVM N-LOS, (c) LSTM LOS, and (d) LSTM N-LOS

Actual	Predicted				
	WALK	RUN	SIT	STAND	EMPTY
(a) SVM confusion matrix—LOS					
WALK	0.80	0.15	0.00	0.05	0.00
RUN	0.05	0.95	0.00	0.00	0.00
SIT	0.00	0.10	0.45	0.45	0.00
STAND	0.00	0.00	0.60	0.40	0.00
EMPTY	0.00	0.00	0.00	0.00	1.00
Actual	Predicted				
	WALK	RUN	SIT	STAND	EMPTY
(b) SVM confusion matrix—N-LOS					
WALK	1.00	0.00	0.00	0.00	0.00
RUN	0.00	0.80	0.00	0.05	0.15
SIT	0.10	0.00	0.60	0.30	0.00
STAND	0.00	0.00	0.35	0.65	0.00
EMPTY	0.05	0.00	0.00	0.00	0.95
Actual	Predicted				
	WALK	RUN	SIT	STAND	EMPTY
(c) LSTM confusion matrix—LOS					
WALK	0.80	0.00	0.10	0.10	0.00
RUN	0.00	0.95	0.00	0.05	0.00
SIT	0.00	0.20	0.80	0.00	0.00
STAND	0.00	0.00	0.00	1.00	0.00
EMPTY	0.00	0.00	0.00	0.00	1.00
Actual	Predicted				
	WALK	RUN	SIT	STAND	EMPTY
(d) LSTM confusion matrix—N-LOS					
WALK	0.85	0.10	0.00	0.00	0.05
RUN	0.00	0.60	0.40	0.00	0.00
SIT	0.05	0.00	0.70	0.20	0.05
STAND	0.05	0.00	0.00	0.90	0.05
EMPTY	0.00	0.00	0.00	0.00	1.00

3.3 Human activities

Samples were collected for five activities namely SIT, STAND, RUN, WALK, and EMPTY. The “Linux 802.11n CSI Tool” proposed by Halperin et al. (2011) was configured in injection and monitor mode to collect the activity samples. For sending packets we configured one laptop in the injection mode and in order to receive packets we configured the other laptop in the monitor mode. We initiated the transmission of 2500 packets with an interval of 15 ms between each packet transmitted and one of the activities was performed during this transmission. The packets were captured and their

corresponding CSI data was logged in a file in the laptop running in monitor mode.

4 Algorithms

We implemented two different algorithms for activity classification—the first one (Algorithm 1) is using an SVM for classification and the second one (Algorithm 2) is using a long short-term memory (LSTM) recurrent neural network for this task. The overall data or process flow is depicted in Fig. 2.

Table 2 Comparison of normalized confusion matrices for *three* activities Mov (WALK and RUN), N-Mov (SIT and STAND) and EMPTY for (a) SVM LOS, (b) SVM N-LOS, (c) LSTM LOS, and (d) LSTM N-LOS

Actual	Predicted		
	MOV	N-Mov	EMPTY
(a) SVM LOS			
Mov	0.975	0.025	0.00
N-Mov	0.05	0.95	0.00
EMPTY	0.00	0.00	1.00
Actual	Predicted		
	Mov	N-Mov	EMPTY
(b) SVM N-LOS			
Mov	0.90	0.025	0.075
N-Mov	0.05	0.95	0.00
EMPTY	0.053	0.00	0.947
Actual	Predicted		
	Mov	N-Mov	EMPTY
(c) LSTM LOS			
Mov	0.875	0.125	0.00
N-Mov	0.10	0.90	0.00
EMPTY	0.00	0.00	1.00
Actual	Predicted		
	Mov	N-Mov	EMPTY
(d) LSTM N-LOS			
Mov	0.775	0.225	0.00
N-Mov	0.05	0.90	0.05
EMPTY	0.00	0.00	1.00

After the seminal work presented in WiFall (Han et al. 2014) many researchers used SVMs for classification of HAR. Therefore we decided to use SVM as well as a benchmark. SVMs require a careful designing of the proper features “by hand”, which we will describe in detail in the sequel. This is true in particular as classifying activities involves classification of temporal data, i.e. data that has a time dimension and a sequential character. Therefore, the time correlations of the signals play a paramount role in the definition and classification of activities and have to adequately be modelled in the feature space—we used discrete wavelet transform (DWT) for this task. In addition, SVMs do not scale well to very high dimensional feature spaces, thus we employed feature reduction techniques as well.

On the other hand in recent works, see e.g. DeepFalls (Chowdhury 2018), deep neural networks have been used for classification, in particular convolutional neural networks (CNN). Henceforth, we wanted to use neural networks as well for comparison. However, CNN are *not* well suited for the classification of temporal data. To this end, recurrent neural network (RNN) (Rumelhart et al. 1986) have been

proposed long time ago. They possess internal memory and a feedback mechanism to allow for information to be “remembered” and thus are able to correlate temporal data. They do not, however, achieve good performance for long sequential data, due to the gradient vanishing problem. LSTMs (Hochreiter and Schmidhuber 1997) were proposed as an alternative to solve this problem. As for HAR classification long-term correlations are important, the choice of LSTM was natural. Similar to other deep networks, LSTM can “automatically” select the features from the data and therefore a manually designed feature extraction is no longer required.

Both algorithms require a careful data cleansing approach after raw data extraction involving CSI-denoising using wavelets and are described in detail in the sequel.

4.1 Activity-SVM-classification

The activity-SVM-Classification algorithm Algorithm 1 is comprised of the following steps which are serially executed. Its steps are described in detail in the sequel. The first four

Table 3 Precision, recall and F-score for LOS

	Precision		Recall		F-score	
	SVM	LSTM	SVM	LSTM	SVM	LSTM
Mov	0.95	0.90	0.98	0.88	0.96	0.89
N-Mov	0.97	0.88	0.95	0.90	0.96	0.89
EMPTY	1.00	1.00	1.00	1.00	1.00	1.00

Table 4 Precision, recall and F-score for N-LOS

	Precision		Recall		F-score	
	SVM	LSTM	SVM	LSTM	SVM	LSTM
Mov	0.92	0.94	0.90	0.78	0.91	0.85
N-Mov	0.97	0.80	0.95	0.90	0.96	0.85
EMPTY	0.86	0.90	0.95	1.00	0.90	0.95

steps extract and prepare the data, and the final classification is achieved by a standard SVM.

1. CSI value extraction
2. Denoising
3. Principal component analysis (PCA)
4. Feature extraction
5. SVM classification

Algorithm 1 ACTIVITY-SVM-CLASSIFICATION

Input: The time series x_τ containing raw CSI data

Output: The predicted activity label l

- ```

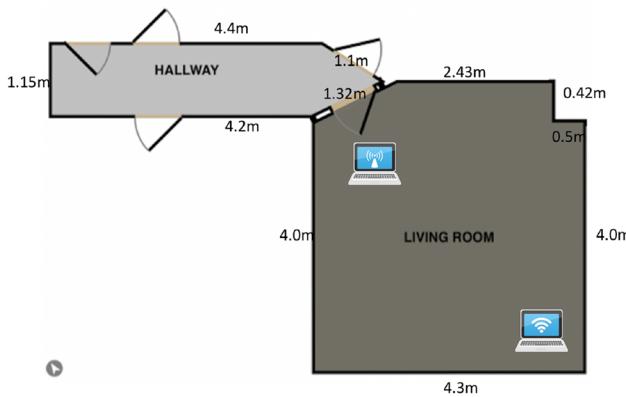
1: $x_\tau \leftarrow \text{CSI_VALUE_EXTRACTION}(x_\tau)$
2: $x_\tau \leftarrow \text{DENOISE}(x_\tau)$
3: $x_\tau \leftarrow \text{PCA}(x_\tau)$
4: $f \leftarrow \text{FEATURE_EXTRACTION}(x_\tau)$
5: $l \leftarrow \text{SVM-CLASSIFICATION}(f)$

```

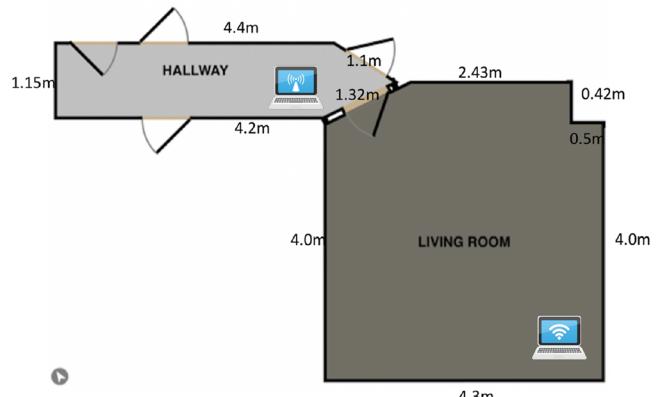
#### 4.1.1 CSI value extraction

Each sample file consists of CSI values for approx. 2500 packets which are logged using one transmitter and three receivers. There exist 30 subcarrier groups between each transmitter and receiver pair. For each packet reception, CSI values are extracted into a matrix of dimension  $N_T \times N_R \times 30$ , where  $N_T$  and  $N_R$  represent the number of transmitters and receivers respectively. The CSI matrix is then flattened to yield a vector of 90 columns, which is then added to a matrix of dimension  $2500 \times 90$ . Each column forms the time series of CSI values for each of the 90 subcarrier group. Note that a CSI value is a complex number and for activity recognition, only the amplitude of the CSI value is considered (i.e. the phase is ignored).

An example for the CSI raw data for the RUN, SIT, WALK, and STAND activity is shown in Fig. 3.

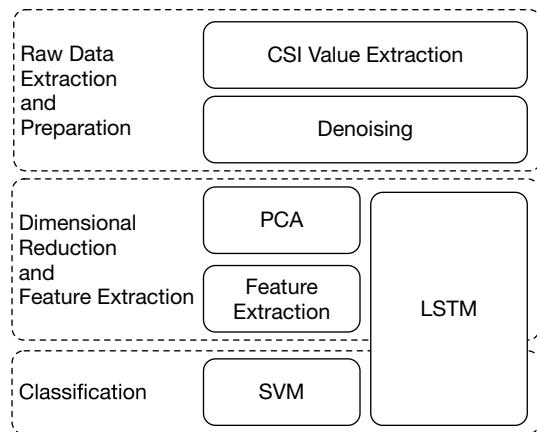


(a) Setup for LOS Scenario



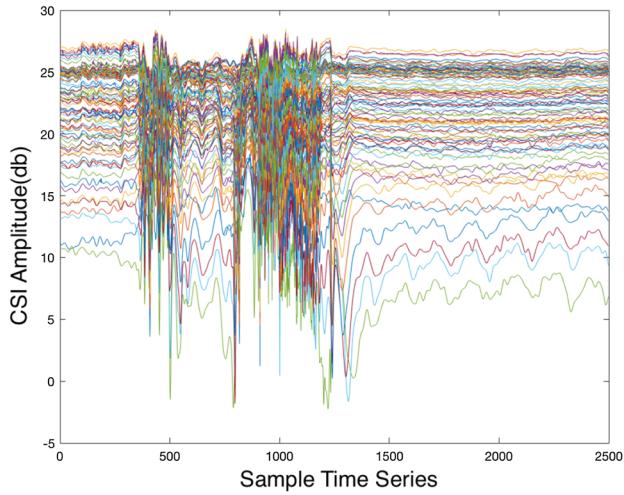
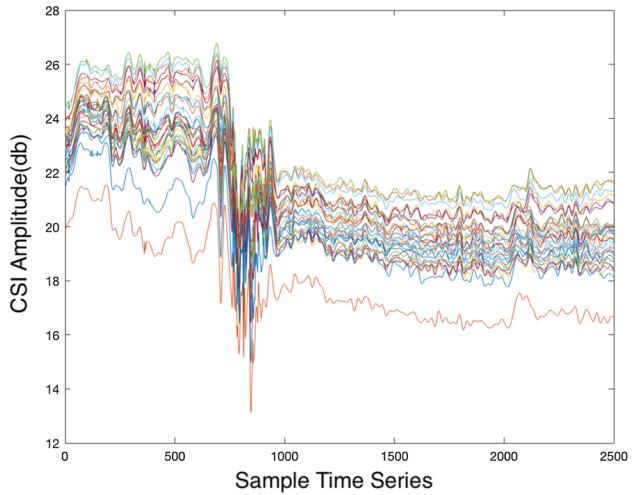
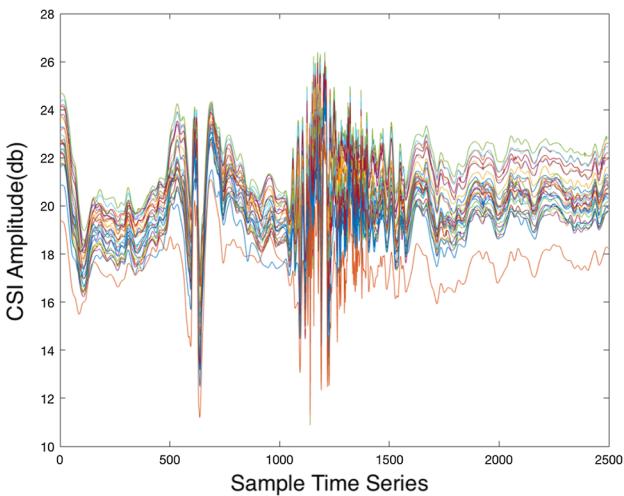
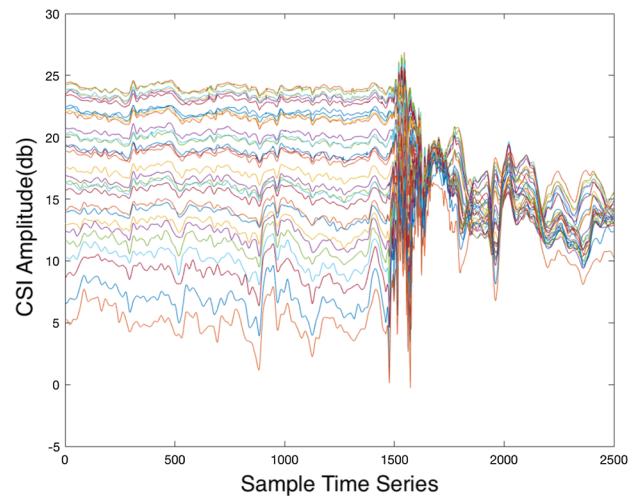
(b) Setup for N-LOS Scenario

**Fig. 1** Experimental setup

**Fig. 2** Data architecture

#### 4.1.2 Denoising

The main goal is to remove the noise, but preserve the sharp spikes caused by human activity. Discrete wavelet transform (DWT) is a common procedure to achieve that. In DWT, a multilevel decomposition of signal is performed by passing it through a set of high pass and low pass filters at each level. The output from the high pass and low pass filters provides the detailed and approximation coefficients respectively. The first level detailed coefficient contains the information about the noise and the sharp changes caused by human activity and therefore the first level detailed co-efficient is used to calculate a threshold. This threshold is then applied to all the detailed coefficients obtained in the different levels and the signal is then reconstructed using the new detailed coefficients. For the denoise algorithm DENOISE, we use

**(a) RUN****(b) SIT****(c) WALK****(d) STAND****Fig. 3** CSI data for different activities

the MATLAB (The MathWorks 2019) function `wden` based on wavelet decomposition. We use “heursure”, the heuristic variant of Stein’s unbiased risk, soft thresholding and the `sym6` wavelet as parameters for our algorithm. `wden` performs the wavelet decomposition, applies the threshold to all the detailed coefficients and reconstructs the signal (time series of each subcarrier obtained in Sect. 4.1.1).

#### 4.1.3 Principal component analysis (PCA)

As a standard feature reduction technique we apply principal component analysis (Karl 1901) on the denoised subcarrier time series data  $x_\tau$ . The first three components explain 70–80% of the variance for all the activities. As the first component contains information due to reflection from stationary objects like furniture, walls etc., only the second and the third principal component are used for the prediction of human activity in the sequel.

#### 4.1.4 Feature extraction

The feature extraction algorithm FEATURE\_EXTRACTION (Algorithm 1.1) makes use of spectral analysis techniques as described in the sequel.

---

##### Algorithm 1.1 FEATURE\_EXTRACTION

---

**Input:** The preprocessed time series  $x_\tau$   
**Output:** The extracted feature vector  $f$

- 1:  $PSD, v \leftarrow$  Compute Power Spectral Density PSD  
 (and Frequency  $v$  for Center of Energy) from  $x_\tau$
  - 2:  $m_{PSD}, m_v \leftarrow$  HAART WAVELET TRANSFORMATION( $PSD, v$ )
  - 3:  $f \leftarrow$  STATISTICAL DATA EXTRACTION( $m_{PSD}, m_v$ )
- 

Different human activities lead to variations in the energy and power of a signal. Power spectral density analysis (PSD) (Stoica and Moses 2005) is a common technique to analyze these effects. Thus, we compute the spectral density  $PSD$  including the frequency  $v$  for the center of energy of our time series  $x(\tau)$  as follows. Let the autocorrelation function  $R(\tau)$  be defined as

$$R(\tau) := x(\tau) \times x(-\tau),$$

then the  $PSD$  is simply the expected value of the Fourier transform of the autocorrelation function

$$PSD := \mathbb{E}[\hat{R}] = \mathbb{E}\left[\int_{-\infty}^{\infty} e^{j\omega\tau} R(\tau) d\tau\right] = \mathbb{E}[|\hat{x}|^2].$$

As the energy corresponding to human activity lies in the lower frequency bins we perform a discrete Haar 1-D wavelet transformation in HAART WAVELET TRANSFORMATION to obtain these coefficients by using the MATLAB function `haar(x, level)`. We set the (maximum) level for

the Haar transform, i.e. the variable `level` to 5. Finally, the statistical data extraction algorithm STATISTICAL DATA EXTRACTION computes the following data:

- For each of the selected subcarrier’s time series we calculate *mean*, *median*, *standard deviation*, *interquartile range*, *second central moment*, *third central moment*, *skewness*, and *kurtosis*.
- For each of row of the selected subcarrier’s PSD matrix we calculate *mean*, *max*, *standard deviation*, *interquartile range*, *skewness*, and *kurtosis*.
- For each of row of the selected subcarrier’s frequency of center of energy we calculate *mean*, *max*, *standard deviation*, and *interquartile range*.

#### 4.1.5 SVM classification

For the final multi-label classification a one-against-all (Bishop 2006) linear support vector machine (Cortes and Vapnik 1995) with a penalty parameter  $C$  set to 0.01 has been used.

### 4.2 Activity-LSTM-classification

As an alternative approach we have used long short-term memory (LSTM) networks, see Algorithm 2. LSTM (Hochreiter and Schmidhuber 1997) are artificial recurrent neural networks which are suitable for processing times-series of data and relevant in our context. For the LSTM algorithm we only perform CSI VALUE EXTRACTION and DENOISE because a manual feature design or selection is not necessary as explained above.

We used TensorFlow (Abadi et al. 2015) and a basic LSTM (BasicLSTMCell) as our implementation technology with the following configuration:

- *Sequence Layer*: It takes as input prepossessed CSI data, i.e. the feature vector is a 90 dimensional vector which contains the raw CSI amplitude of each of the 90 subcarriers.
- *LSTM layer*: TensorFlow BasicLSTMCell with 128 hidden units
- *Softmax layer*: It normalizes and prepares data for classification known also as multi-class generalization.
- *Classification layer*: It applies cross entropy to classify and give the final output. For minimizing the cross entropy loss, Adam Optimizer is used with a batch size of 128 and a learning rate set to  $10^{-3}$ .

The drawback of this approach compared to using an SVM is that it takes a lot of time to train the model (for 1,900–2,400 time steps), if the computer is not using a GPU. For example, on a MAC with a 3.3 GHz Intel core i5 processor it

takes more than an hour to train the model, whereas by using NVIDIA GeForce GTX 1080 Ti, the model could be trained in 9 min approx. (The computation of algorithm Algorithm 1 on the other hand is almost instantaneous.)

---

**Algorithm 2** ACTIVITY-LSTM-CLASSIFICATION

---

**Input:** The time series  $x_\tau$  containing raw CSI data  
**Output:** The predicted activity label  $l$

- 1:  $x_\tau \leftarrow \text{CSI.VALUE.EXTRACTION}(x_\tau)$
- 2:  $x_\tau \leftarrow \text{DENOISE}(x_\tau)$
- 3:  $l \leftarrow \text{LSTM-CLASSIFICATION}(x_\tau)$

---

## 5 Evaluation

For each activity we collected 200 samples and analyzed them using algorithm Algorithm 1 (SVM-classification) and algorithm Algorithm 2 (LSTM-classification).

### 5.1 Results

Table 1a, b depict the confusion matrix obtained for the testing of activity recognition using a SVM for LOS and N-LOS resp. and Table 1c, d depict the confusion matrix obtained for the testing of activity recognition using a LSTM for LOS and N-LOS resp.

In addition, in order to compare our algorithm better to classifiers detecting moving from non-moving activities, we computed a confusion matrix by aggregating WALK and RUN into MOV and STAND and SIT into N-Mov, see Table 2.

### 5.2 Interpretation

As one can infer from Table 1 both algorithms are able to detect presence (Non-EMPTY) and non-presence (EMPTY) very well. From Table 2 we infer that in particular the classification between moving and non-moving activities is almost perfect.

Both algorithms have difficulties differentiating between similar activities, i.e. between SIT and STAND or WALK and RUN. In general, the SVM algorithm out-performs the LSTM algorithm as can be inferred from Tables 3 and 4 comparing precision, recall and F-scores.

However, given the fact, that the LSTM algorithm does not use *any* sophisticated pre-processing it is an interesting result, that the LSTM algorithm performs so well in comparison.

## 6 Extension 1: fall detection

Fall is a prominent health problem in particular with older people due to its severe consequences both physically and mentally. According to the World Health Organization

(WHO) (2019), fatal falls occur 646,000 times each year worldwide. In most of the cases, these incidences occur with people over the age of 60. Among non-fatal falls 37.3 million of them are serious enough to require medical help (WHO 2019). Falls can cause very serious consequences—as expressed in Allcock and O’Shea (2000) “of those admitted to hospitals after a fall, only about half are alive 1 year later”. Therefore, fall detection and prevention is an active and critical area of research because it can help elderly people to depend less on caregivers and allow them to live and move more independently. Another high-risk group to fall is children. Falls may occur while they are developing and exploring the environment (WHO 2019). In everyday life, we are surrounded by risks of fall. An example could be an employee in a factory who suffers a fall and no one is around to provide fast assistance. Another example could be a person at home falling after trying to change a light bulb.

### 6.1 Prior work

Most approaches require patients to wear sensor belts or equivalent devices, see (Özdemir 2014; Gutiérrez-Madroñal et al. 2019; La Blunda and Wagner 2016a, b). All of the above fall detection technologies are based on sensor fusion of specific sensors e.g. including ECG data (La Blunda and Wagner 2016a). Another solution called WiFall based on CSI data is proposed by Wang et al. (2017b) (as an improvement of Han et al. (2014)). In WiFall local outlier factor (Breunig et al. 2000) was used for anomaly detection. From anomaly pattern the following features were extracted: *normalized standard deviation, offset of signal strength, period of the motion, median absolute deviation, interquartile range, signal entropy and velocity of signal change*. As a classifier one-class SVM was used. Activities were performed in a chamber, laboratory, and dormitory. On average WiFall detected falls with 87% precision and with 18% of false fall detection in a dormitory (i.e. non-laboratory) environment. Another solution called RT-Fall is proposed by Wang et al. (2017a). This solution used 1-D linear interpolation and band-pass filter or signal preprocessing. In RT-Fall the same features as in WiFall were extracted and additionally time lag and power decline ratio (PDR) were extracted. As a classifier a one-class SVM with Gaussian radial basis kernel function was used. Results of RT-Fall were 91% for sensitivity and 92% for specificity.

### 6.2 Our contribution

In order to support a device-free use case, we try to transfer the activity classification described in Sect. 4 to include fall activity as well. Therefore, we aim to detect falls without any additional hardware to be worn or carried by the patient. To (partially) achieve that goal, we had to perform a two step

classification combining two SVMs to eliminate false alarms as much as possible, see below.

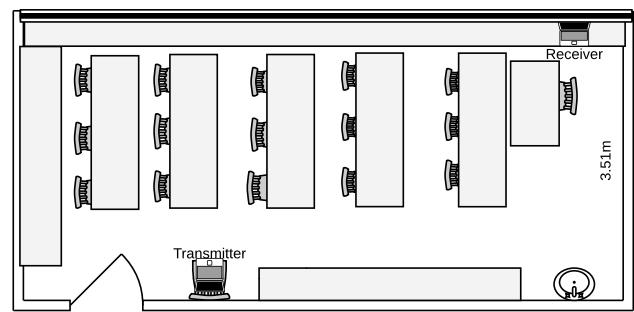
### 6.3 Setup

The collection of data is performed for the following human activities: WALKING, STANDING UP, SITTING DOWN, FALL and EMPTY where no activities were performed. For each activity, we collected 140 samples. As a FALL activity, it was decided to perform two types of falls. The first type is walking and falling shown in Fig. 4a and the second type is sitting on the chair, standing up and then falling as shown in Fig. 4b.

In both cases there are two possibilities of what can happen after the fall namely post fall, either person stands up or keeps lying. If a person stands up after falling it is labeled as FALSE ALARM in the sequel. Out of the 140 fall activities we chose to have 60 followed by a person standing up, henceforth the fall activities were split into 80 FALLS and 60 FALSE ALARMS resulting in 700 activities in total.

Data collection is performed at the Frankfurt University of Applied Sciences in building 1, room 235 shown in Fig. 5.

Transmitter and receiver are placed as far from each other as it is possible. All the activities are performed in different parts of the room to have more realistic data that does not depend on the location of the human in the room. For each sample of activity 500 packets are sent with an interval of 10 ms between packets which sum up to 5 s for an activity. The exception is the FALL activity since 5 s are not enough to perform the fall and post-fall event. Thus FALL is divided into two parts namely “Falling” and “Post Fall”. The first part, “Falling” is performed for 5 s and contains 500 packets, the second part, “Post Fall”, is performed for 15 s which is 1500 packets with 10 ms interval. “Post Fall” is observed for 15 s because this time will be enough for the human to stand up if it was not a real fall. In case of a real fall 15 s of lying down after the fall event proves that the person suffered a real fall. The second part of FALL activity was done to avoid situations



**Fig. 5** Experimental setup

where a person does not need help after a fall and the fall will be considered as a false alarm and labeled as such, i.e. as FALSE ALARM. Captured packets are logged into the file that is later used to extract CSI values. The interval between logged in packets between samples is 5 s. To separate one sample from the next one the timestamp of packets is used.

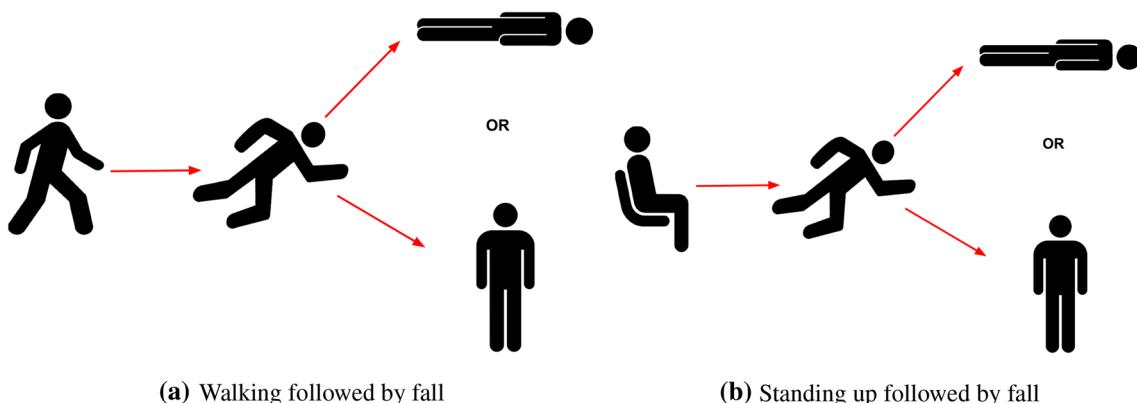
### 6.4 Evaluation metrics

For performance evaluation, we will use sensitivity, precision and specificity. Sensitivity shows the percentage of correctly detected falls and specificity is the percentage of correctly detected non-fall activities. Precision shows the percentage of correctly detected falls from all predicted falls (including false positives). They are defined as follows:

$$\text{sensitivity} := \frac{TP}{TP + FN}, \quad \text{specificity} := \frac{TN}{TN + FP},$$

$$\text{and} \quad \text{precision} := \frac{TP}{TP + FP},$$

where  $TP$ ,  $FP$ ,  $TN$  and  $FN$  denote the true and false positives and negatives, resp.



**Fig. 4** Fall scenarios

## 6.5 Algorithms

### 6.5.1 Data denoising and feature extraction

The algorithm Algorithm 1 was used for classification. However, the number of extracted principal components was increased due to the lack of information to classify properly. Dropping the first principal component led to the loss of important activity information. Thus, it was decided to include the first principal component into the analysis as well. In an effort to keep the algorithm as simple as possible and using as little statistical information as possible we modified the feature extraction algorithm Algorithm 1.1 by dropping the computation of the power spectral density (PSD) and frequency  $\nu$  for the center of energy altogether. For the statistical feature extraction, the following parameters were used: *mean value, standard deviation, median, second, and third central moments, skewness and kurtosis*.

### 6.5.2 Combining two SVMs

Since the FALL activity consists of two parts, we had to modify the classification approach. We used two different SVMs optimized for detecting the two independent activities. For the first SVM, we classified the five activities namely WALKING, STANDING UP, SITTING DOWN, FALL and EMPTY. If the output of SVMs gives the FALL result then the second check will be performed. For the second check, a new SVM is created to analyze only “Post Fall” information. It has only two classes namely FALL and FALSE ALARM. FALL means that the person fell and kept lying—however, FALSE ALARM means that person stood up after the fall. If the second check SVM votes for FALL then the activity will be labeled correspondingly if it was not fall followed by lying then it is labeled as FALSE ALARM. The second check is done to avoid cases when a person falls but does not need any medical help.

## 6.6 Result

In Table 5a the confusion matrix for four activities *excluding* fall is depicted and in Table 5b the confusion matrix for six activities *including* fall scenarios is depicted.

It can be seen that a perfect classification of WALKING, STANDING UP and EMPTY can be achieved. The only exception is SITTING DOWN. After adding (additional) fall events to the classifier it can be observed in Table 5b that the results for some classes naturally became worse.

For SITTING DOWN the true labeling number has decreased. The reason for that can be inferred from observing that signals for these two activities are very similar as shown in Fig. 6a, b. During the SITTING DOWN activity the person sits down and keeps sitting which makes it similar to FALL and keep lying. However WALKING, STANDING UP, EMPTY, FALL

and FALSE ALARM activities are classified with high accuracy. It is a common, known (and intuitive to understand) phenomenon that falling is difficult to distinguish from sitting down, see (Gutiérrez-Madroñal et al. 2019) or (Li et al. 2009).

Using Table 5b we calculate the performance measures sensitivity and specificity and precision for all fall activities, i.e. FALL and FALSE ALARM activities together as well as for just FALL. The results are depicted in Table 6. Compared to the results in WiFall and RT-Fall, the sensitivity and specificity of our model are higher. Both of the above-mentioned approaches used one SVM classifier to recognize activities. Whereas our solution had one SVM for detecting all activities and the second for detecting only fall activities. Combining two SVMs increased the rate of both sensitivity and specificity. On the other hand, our precision of 62% considering only FALL labeled activities and 74% considering *all* fall activities, i.e. FALL and FALSE ALARM activities together, is less than the one reported in WiFall. Recent results of our research group indicate that using LSTM for FALL detection can improve the specificity to 98.7% at the expense of sensitivity which drops to of 81.3%, see (Yaşar 2019). For a practical deployment however, it is necessary to capture almost every FALL and still having almost no FALSE ALARMS. Therefore, in future research we will investigate, how to combine these approaches to achieve maximum sensitivity *and* specificity.

## 6.7 Conclusion

As the results can be observed in Table 5a, b, most of the activities can be classified with 95% and higher accuracy. There are some difficulties classifying similar activities like SITTING DOWN and FALL since there is no activity after SITTING DOWN which makes the signal for the SITTING DOWN activity look very similar to FALL. However, real FALL activities are not confused with other activities and in 100% of test data “Fall” is detected correctly. To improve correct labeling for SITTING DOWN activity the statistical information provided might not be enough. Extraction of speed information or according to the solution proposed by Damodaran and Schäfer (2019) energy and power of a signal might be helpful to increase values for SITTING DOWN activity. These are left for future research, in particular we will investigate whether including additional statistical information derived from energy and power of the signals will improve the results any further.

## 7 Extension 2: counting people

While it is yet difficult and almost impossible to predict and understand crowd *behavior*, it is not infeasible to *count* the number of people in a certain environment—called crowd

**Table 5** Confusion matrices

| Actual                                    | Predicted |             |              |       |
|-------------------------------------------|-----------|-------------|--------------|-------|
|                                           | WALKING   | STANDING UP | SITTING DOWN | EMPTY |
| (a) Four activities <i>excluding</i> FALL |           |             |              |       |
| WALKING                                   | 1         | 0           | 0            | 0     |
| STANDING UP                               | 0         | 1           | 0            | 0     |
| SITTING DOWN                              | 0.25      | 0.05        | 0.7          | 0     |
| EMPTY                                     | 0         | 0           | 0            | 1     |
| Actual                                    | Predicted |             |              |       |
|                                           | WALKING   | STANDING UP | SITTING DOWN | EMPTY |
| (b) Six activities <i>including</i> FALL  |           |             |              |       |
| WALKING                                   | 0.95      | 0           | 0            | 0.05  |
| STANDING UP                               | 0         | 1           | 0            | 0     |
| SITTING DOWN                              | 0.1       | 0.05        | 0.55         | 0.3   |
| EMPTY                                     | 0         | 0           | 0            | 0     |
| FALL                                      | 0         | 0           | 0            | 1     |
| FALSE ALARM                               | 0         | 0           | 0            | 0     |

counting. Crowd counting is not only an interesting and challenging task, it can be used in different applications such as crowd control, marketing research, and guided tours. For us, the crowd-counting use-case is not only of theoretical interest but bears also some practical consequences in a campus setup, see e.g. (Aversente et al. 2016).

## 7.1 Prior work

Crowd counting approaches are mainly classified into two categories: video based recognition and non-image based localization (Xi et al. 2014). Most of the existing crowd-counting models are based on computer vision techniques (Li et al. 2008), however these approaches suffer from blind spots for the cameras and the absence of light. They are also more sensitive towards privacy issues. There are also special hardware systems such as WiTrack (Adib et al. 2014) that provide a well-constructed signal utilizing time-of-flight (TOF) by frequency modulated continuous wave (FMCW). In addition, the setup for such devices is a lot more expensive than for ubiquitous WiFi devices.

## 7.2 Our contribution

We analyzed how our algorithms could be used to support crowd counting. We chose algorithm Algorithm 2 for its versatility as a test candidate for our experiments. It should be understood, however, that the work presented in this section represents work in progress and we provide only preliminary, actual interim results here and leave a full analysis to a forthcoming publication (including a thorough comparison

to other approaches including e.g. the much more sophisticated analysis presented in Xi et al. (2014).

## 7.3 Setup

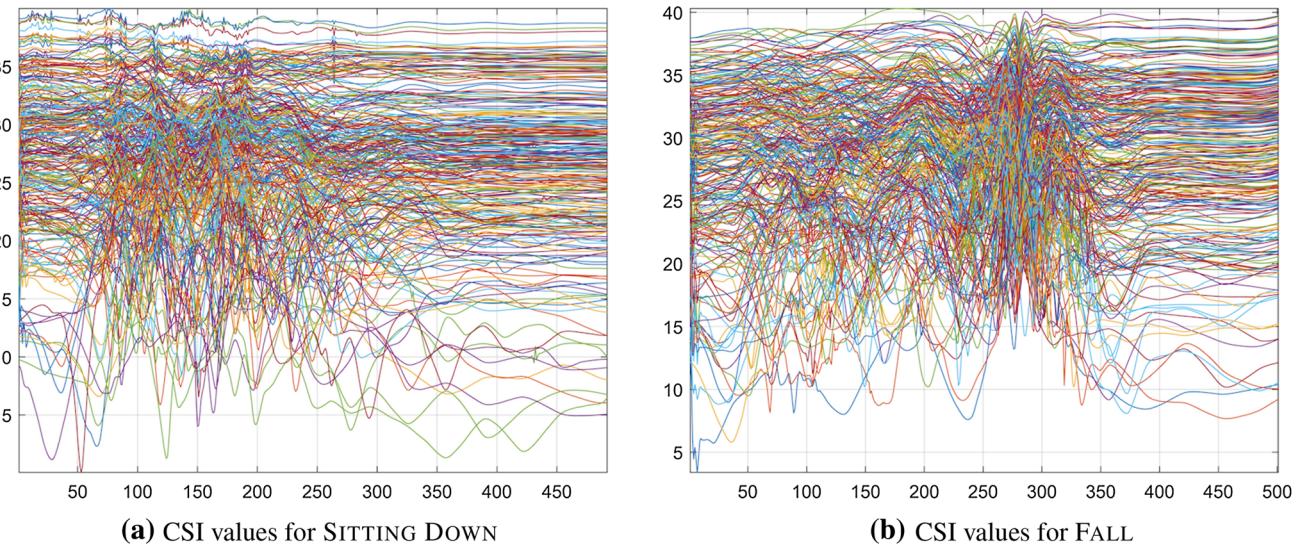
The experiments were performed in building 1, room 235, Frankfurt University of Applied Sciences. The transmitter was placed at the entrance of the room and the receiver at the window on the opposite corner of the laboratory as shown in Fig. 5.

For experiments we used as subjects three students, one female and two males, and we placed the laptops into different corners of the room. Then we performed the following activities:

1. for a single subject: sitting somewhere in the room, sitting down, standing up, standing somewhere in the room, walking around the room,
2. for two subjects: walking around the room, standing somewhere in the room and sitting in different places in the room, and
3. for three subjects: sitting in the room in different places.

Henceforth, we created the data-set as depicted in Fig. 7. This time we used 3 Rx and 3 Tx antennas from which we obtain a CSI matrix with dimensions  $500 \times 270$ . Each column forms a time series of CSI values for each of the 270 Rx-Tx pairs of subcarriers.

This way we recorded 1500 samples altogether; each sample was recorded in a time of 5 s transmitting 100 packet per second. Thus, from our experiments we got 10 classes

**Fig. 6** CSI values**Table 6** Sensitivity, specificity and precision for Fall

|                                            | Sensitivity | Specificity | Precision |
|--------------------------------------------|-------------|-------------|-----------|
| FALL (80 cases) and FALSE ALARM (60 cases) | 1.00        | 0.91        | 0.74      |
| FALL alone (80 cases)                      | 1.00        | 0.92        | 0.62      |

altogether. These classes are used as the following labels: EMPTY, SIT DOWN, SITTING 1 SUBJECT, SITTING 2 SUBJECTS, SITTING 3 SUBJECTS, STAND UP, WALKING 1 SUBJECT, STANDING 2 SUBJECTS, WALKING 1 SUBJECT, WALKING 2 SUBJECTS.

## 7.4 Algorithms

We implemented algorithm Algorithm 2 described in Sect. 4.2 with the following properties. Firstly we denoised the signal using symlets and a level of decomposition 6. The level of decomposition is related to the length of the time-series which is 500, and symlets because they are designed to have the least asymmetry and the maximum number of vanishing moments for a given compact (Meenakshi Chaudhary 2013). Then we implemented an LSTM in MATLAB (The MathWorks 2019) (using the function `lstmLayer`) with the following configuration:

- *Sequence layer*: It takes as input prepossessed CSI data, a vector with 270 dimensions/features.
- *LSTM layer*: Configured with 200 hidden units
- *Dropout layer*: Set to a value of 0.5 which will decrease the probability of overfitting
- *Fully connected layer*: The classification layer or the result of the training

- *Softmax layer*: It normalizes and prepares the data for classification known also as multi-class generalization.
- *Classification layer*: It applies cross entropy to classify and give the final output.

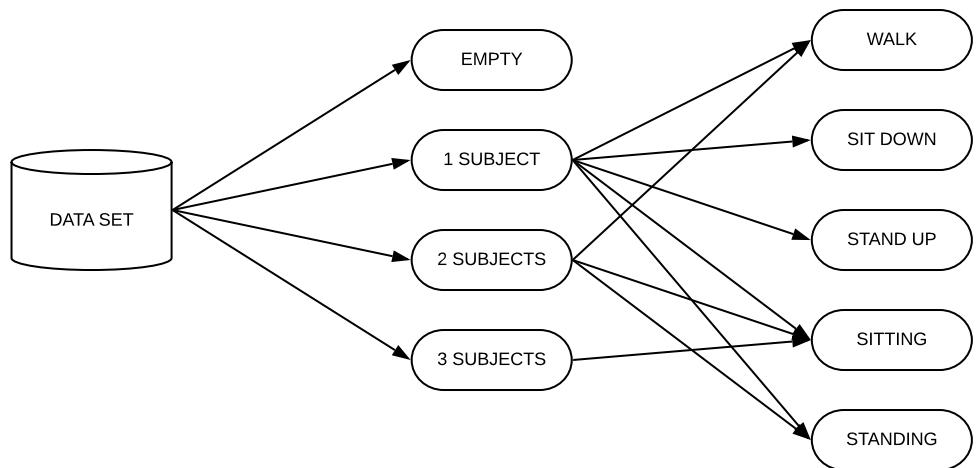
We used as well the Adam Optimizer with a batch size of 20 to minimize the loss of the data samples and a learning rate of  $10^{-3}$ , we also shuffled data after every epoch in the learning process to increase the accuracy. We build two different networks with similar properties with some small changes between them described below.

### 7.4.1 Crowd counting activity detection LSTM (CCAC-LSTM)

For this approach, we use the activities of the people in the room to classify how many people are in the room. Specific properties for this approach were 10 epochs, 200 hidden units and a batch size of 20, with 10 classes. With this configuration we achieved good performance, which is discussed below.

### 7.4.2 Crowd counting LSTM (CC-LSTM)

From the confusion matrix depicted in Table 7 we can conclude that the CCAC-LSTM model can very well classify if there are one, two or three person(s) in the room. Based on that we reconstructed the data to form a data-set with four classes: EMPTY, 1 SUBJECT, 2 SUBJECTS, 3 SUBJECTS. This way we try to predict only the number of people in the room which was our original goal. This time we decreased the batch size to 10 and number of classes to 4, because the entry data is different and it fits better with these properties.

**Fig. 7** Data set structure**Table 7** CCAC-LSTM confusion matrix

| Predicted           | Actual |          |                   |                    |                    |          |                   |                     |                   |                    |   |   |
|---------------------|--------|----------|-------------------|--------------------|--------------------|----------|-------------------|---------------------|-------------------|--------------------|---|---|
|                     | EMPTY  | SIT DOWN | SITTING 1 SUBJECT | SITTING 2 SUBJECTS | SITTING 3 SUBJECTS | STAND UP | WALKING 1 SUBJECT | STANDING 2 SUBJECTS | WALKING 1 SUBJECT | WALKING 2 SUBJECTS |   |   |
| EMPTY               | 1      | 0        | 0                 | 0                  | 0                  | 0        | 0                 | 0                   | 0                 | 0                  | 0 | 0 |
| SIT DOWN            | 0      | 1        | 0                 | 0                  | 0                  | 0        | 0                 | 0                   | 0                 | 0                  | 0 | 0 |
| SITTING 1 SUBJECT   | 0      | 0        | 1                 | 0                  | 0                  | 0        | 0                 | 0                   | 0                 | 0                  | 0 | 0 |
| SITTING 2 SUBJECTS  | 0      | 0        | 0                 | 1                  | 0                  | 0        | 0                 | 0                   | 0                 | 0                  | 0 | 0 |
| SITTING 3 SUBJECTS  | 0      | 0        | 0                 | 0.02               | 0.98               | 0        | 0                 | 0                   | 0                 | 0                  | 0 | 0 |
| STAND UP            | 0      | 0        | 0                 | 0                  | 0                  | 1        | 0                 | 0                   | 0                 | 0                  | 0 | 0 |
| WALKING 1 SUBJECT   | 0      | 0        | 0                 | 0                  | 0                  | 0        | 1                 | 0                   | 0                 | 0                  | 0 | 0 |
| STANDING 2 SUBJECTS | 0      | 0        | 0                 | 0                  | 0                  | 0        | 0                 | 1                   | 0                 | 0                  | 0 | 0 |
| WALKING 1 SUBJECT   | 0      | 0        | 0                 | 0                  | 0                  | 0        | 0                 | 0                   | 1                 | 0                  | 0 | 0 |
| WALKING 2 SUBJECTS  | 0      | 0        | 0                 | 0                  | 0                  | 0        | 0                 | 0                   | 0                 | 0                  | 1 | 0 |

Training the same model with the new data we achieved again good results, which are discussed below, too.

## 7.5 Results

Table 7 represents the complete confusion matrix for five different activities and 1–3 person(s) are performing these activities.

From the confusion matrix we can see that the accuracy is pretty high except for a small misclassification between 2 and 3 subjects sitting in different parts of the room.

Table 8a is the normalized confusion matrix that we can derive from CCAC-LSTM only considering the number of subjects in the room, ignoring what are they doing since our goal was to count how many people were in the room not what they are doing. Table 8b is the confusion matrix for the second model CC-LSTM.

Comparing Table 8a with Table 8b we infer that the model that uses human activity is more accurate, but also we see that the misclassification happens mostly on one class EMPTY and this can happen because while recording the

empty room scenario, we left the laptops running recording the CSI. However, as the room was a regular seminar room of the university people passed by in the hallway directly next to the sender which could have affected the CSI. And from the result of scenario N-LOS in Table 2d we know that those factors could have a detrimental effect on the results.

From the confusion matrices we computed an overall accuracy of 99.72% for the CAC-LSTM algorithm, and an overall accuracy of 97.40% for the CC-LSTM algorithm.

## 7.6 Future work

For future work, we plan to more systematically analyze the counting problem and compare it to more sophisticated approaches. It would be interesting to compare whether a supervised learning approach using a deep-learning regression technique can compete with or even outperform explicit models like the one developed in Xi et al. (2014) which is based on Grey Verhulst Model.

**Table 8** Comparison of normalized confusion matrices for crowd counting of maximally three person(s) in a room, (a) CCAC-LSTM (derived from Table 7) normalized to consider only the number of subjects (b) CC-LSTM with four classes that represent number of subjects in room

| Actual                          | Predicted |           |            |            |
|---------------------------------|-----------|-----------|------------|------------|
|                                 | EMPTY     | 1 SUBJECT | 2 SUBJECTS | 3 SUBJECTS |
| (a) CCAC-LSTM normalized matrix |           |           |            |            |
| EMPTY                           | 1         | 0         | 0          | 0          |
| 1 SUBJECT                       | 0         | 1         | 0          | 0          |
| 2 SUBJECTS                      | 0         | 0         | 1          | 0          |
| 3 SUBJECTS                      | 0         | 0         | 0.02       | 0.98       |
| Actual                          | Predicted |           |            |            |
|                                 | EMPTY     | 1 SUBJECT | 2 SUBJECTS | 3 SUBJECTS |
| (b) CC-LSTM normalized matrix   |           |           |            |            |
| EMPTY                           | 0.67      | 0.02      | 0.31       | 0          |
| 1 SUBJECT                       | 0         | 1         | 0          | 0          |
| 2 SUBJECTS                      | 0         | 0         | 1          | 0          |
| 3 SUBJECTS                      | 0         | 0.025     | 0          | 0.975      |

## 8 Summary and outlook

We have shown that combining standard machine learning algorithms such as SVM and LSTM and combining these with sophisticated techniques from signal analysis such as wavelet decomposition it is possible to detect various human activities with high accuracy including fall, presence and counting the number of people in a room. For future research we will analyze how to further improve the accuracy by fine-tuning our algorithms. Furthermore, we will check the stability of the trained models, i.e. whether pre-trained models can be easily transferred into a different environment (such as a different flat)—ideally with minimal or no training. For future work, we will also combine pre-processing techniques with LSTM to further improve performance. Last but not least, we will combine contextual information and fuse the methods described in this paper with recent work on LSTM for FALL detection (Yaşar 2019) to achieve better sensitivity and specificity.

**Acknowledgements** Open Access funding provided by Projekt DEAL. The authors are grateful to Elpiniki Tsakalaki for supporting the project with her expertise on CSI analysis and to Joel Stein for technical support.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will

need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

- Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X.: TensorFlow: Large-scale machine learning on heterogeneous systems (2015). <https://www.tensorflow.org/>
- Adib, F., Katabi, D.: See through walls with wifi!. SIGCOMM Comput. Commun. Rev. **43**(4), 75–86 (2013). <https://doi.org/10.1145/2534169.2486039>
- Adib, F., Kabelac, Z., Katabi, D., Miller, R.C.: 3d tracking via body radio reflections. In: Proceedings of the 11th USENIX Conference on Networked Systems Design and Implementation, NSDI'14, pp. 317–329. USENIX Association, Berkeley (2014). <http://dl.acm.org/citation.cfm?id=2616448.2616478>
- Al-Qaness, M.A.A., Al-Eryani, Y., Al-Jallad, N.: Indoor human activity recognition method using CSI of wireless signals. In: International Symposium on Computer Science and Artificial Intelligence (ISCSAI), vol. 1, no. 3, pp. 49–51 (2017). <https://intelcomp-design.com/Archives/ISCSAI%20ISSUE3/ISCSAI019.pdf>
- Al-Qaness, M.A.A., Li, F., Ma, X., Zhang, Y., Liu, G.: Device-free indoor activity recognition system. Appl. Sci. **6**(11) (2016). <https://doi.org/10.3390/app6110329>. <http://www.mdpi.com/2076-3417/6/11/329>
- Allcock, L.M., O’Shea, D.: Diagnostic yield and development of a neurocardiovascular investigation unit for older adults in a district hospital. J. Gerontol. Ser. A **55**(8), M458–M462 (2000). <https://doi.org/10.1093/gerona/55.8.M458>
- Aversente, F., Klein, D., Sultani, S., Vronski, D., Schäfer, J.: Deploying contextual computing in a campus setting. In: 11th International Network Conference 2016 (INC2016) (2016)

- Bahl, P., Padmanabhan, V.N.: Radar: an in-building rf-based user location and tracking system. In: Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No. 00CH37064), vol. 2, pp. 775–784 (2000). <https://doi.org/10.1109/INFCOM.2000.832252>
- Bishop, C.M.: Pattern Recognition and Machine Learning (Information Science and Statistics). Springer, Berlin (2006)
- Breunig, M., Kriegel, H.P., Ng, R.T., Sander, J.: Lof: Identifying density-based local outliers. In: Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, pp. 93–104. ACM (2000)
- Chen, C., Liao, C., Xie, X., Wang, Y., Zhao, J.: Trip2vec: a deep embedding approach for clustering and profiling taxi trip purposes. Personal and Ubiquitous Computing **23**(1), 53–66 (2019). <https://doi.org/10.1007/s00779-018-1175-9>
- Cheng, Y., Chang, R.Y.: Device-free indoor people counting using wi-fi channel state information for internet of things. In: GLOBECOM 2017—2017 IEEE Global Communications Conference, pp. 1–6 (2017). <https://doi.org/10.1109/GLOCOM.2017.8254522>
- Chowdhury, T.Z.: Using wi-fi channel state information (CSI) for human activity recognition and fall detection. Master's thesis, University of British Columbia (2018)
- Cortes, C., Vapnik, V.: Support-vector networks. In: Machine Learning, pp. 273–297 (1995)
- Damodaran, N., Schäfer, J.: Device free human activity recognition using wifi channel state information. In: 16th IEEE International Conference on Ubiquitous Intelligence and Computing (UIC 2019), 5th IEEE Smart World Congress, Leicester, vol. 16. IEEE (2019)
- Gutiérrez-Madroñal, L., La Blunda, L., Wagner, M.F., Medina-Bulo, I.: Test event generation for a fall-detection IoT system. IEEE Internet Things J. (2019)
- Halperin, D., Hu, W., Sheth, A., Wetherall, D.: Tool release: gathering 802.11n traces with channel state information. SIGCOMM Comput. Commun. Rev. **41**(1), 53–53 (2011). <https://doi.org/10.1145/1925861.1925870>
- Han, C., Wu, K., Wang, Y., Ni, L.M.: Wifall: device-free fall detection by wireless networks. In: IEEE INFOCOM 2014—IEEE Conference on Computer Communications, pp. 271–279 (2014). <https://doi.org/10.1109/INFCOM.2014.6847948>
- Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997). <https://doi.org/10.1162/neco.1997.9.8.1735>
- Karl, P.: On lines and planes of closest fit to systems of points in space. Lond. Edinb. Dublin Philos. Mag. J. Sci. **2**(11), 559–572 (1901). <https://doi.org/10.1080/14786440109462720>
- Kotaru, M., Joshi, K., Bharadia, D., Katti, S.: Spotfi: decimeter level localization using wifi. SIGCOMM Comput. Commun. Rev. **45**(4), 269–282 (2015). <https://doi.org/10.1145/2829988.2787487>
- La Blunda, L., Wagner, M.: Fall-detection belt based on body area networks. In: BIOTELEMETRY 2016 21st Symposium of the International Society on Biotelemetry (ISOB 2016) (2016a)
- La Blunda, L., Wagner, M.: The usage of body area networks for fall detection. In: Proceedings of the Eleventh International Network Conference (INC 2016), p. 159 (2016b)
- Li, M., Zhang, Z., Huang, K., Tan, T.: Estimating the number of people in crowded scenes by mid based foreground segmentation and head-shoulder detection. In: 2008 19th International Conference on Pattern Recognition, pp. 1–4 (2008). <https://doi.org/10.1109/ICPR.2008.4761705>
- Li, Q., Stankovic, J.A., Hanson, M.A., Barth, A.T., Lach, J., Zhou, G.: Accurate, fast fall detection using gyroscopes and accelerometer-derived posture information. In: 2009 Body Sensor Networks, pp. 138–143 (2009)
- Ma, J., Wang, H., Zhang, D., Wang, Y., Wang, Y.: A survey on wi-fi based contactless activity recognition. In: 2016 Intl IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People and Smart World Congress, pp. 1086–1091 (2016)
- Meenakshi Chaudhary, A.D.: A brief study of various wavelet families and compression techniques. J. Glob. Res. Comput. Sci. **4**(4), 43–49 (2013)
- Özdemir, A.T., Barshan, B.: Detecting falls with wearable sensors using machine learning techniques. Sensors (Basel) **14**, 10691–708 (2014)
- Pu, Q., Gupta, S., Gollakota, S., Patel, S.: Whole-home gesture recognition using wireless signals. In: MobiCom 13 Proceedings of the 19th annual international conference on Mobile computing & networking, pp. 27–38 (2013). <https://dl.acm.org/citation.cfm?id=2500436>
- Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning representations by back-propagating errors. Nature **323**(6088), 533–536 (1986). <https://doi.org/10.1038/323533a0>
- Schäfer, J.: Practical concerns of implementing machine learning algorithms for w-lan location fingerprinting. In: Proceedings of ICUMT 2014, 6th International Congress on Ultra Modern Telecommunications, St. Petersburg, pp. 410–417. IEEE Computer Society (2014)
- Scheurer, S., Tedesco, S., Brown, K.N., O'Flynn, B.: Sensor and feature selection for an emergency first responders activity recognition system. In: 2017 IEEE SENSORS, pp. 1–3 (2017). <https://doi.org/10.1109/ICSENS.2017.8234090>
- Schulz, M.: Teaching your wireless card new tricks: Smartphone performance and security enhancements through wi-fi firmware modifications. Ph.D. thesis, Technische Universität, Darmstadt (2018). <http://tubiblio.ulb.tu-darmstadt.de/105239/>
- Stoica, P., Moses, R.: Spectral Analysis of Signals. Pearson Prentice Hall (2005). <https://books.google.de/books?id=h78ZAQAAIAAJ>
- Tan, B., Chen, Q., Chetty, K., Woodbridge, K., Li, W., Piechocki, R.: Exploiting wifi channel state information for residential healthcare informatics. IEEE Commun. Mag. **56**(5), 130–137 (2018). <https://doi.org/10.1109/MCOM.2018.1700064>
- The MathWorks, I.: Symbolic Math Toolbox, Natick (2019). <https://www.mathworks.com/help/symbolic/>
- Tsakalaki, E., Schäfer, J.: On application of the correlation vectors subspace method for 2-dimensional angle-delay estimation in multipath ofdm channels. In: 2018 14th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), pp. 1–8 (2018). <https://doi.org/10.1109/WiMOB.2018.8589183>
- Wang, Y., Liu, J., Chen, Y., Gruteser, M., Yang, J., Liu, H.: E-eyes: device-free location-oriented activity identification using fine-grained wifi signatures. In: MobiCom '14 Proceedings of the 20th annual international conference on Mobile computing and networking, pp. 617–628 (2014). <https://dl.acm.org/citation.cfm?id=2639143>
- Wang, W., Liu, A.X., Shahzad, M., Ling, K., Lu, S.: Understanding and modeling of wifi signal based human activity recognition. In: Proceedings of the 21st Annual International Conference on Mobile Computing and Networking, MobiCom '15, pp. 65–76. ACM, New York (2015a). <https://doi.org/10.1145/2789168.2790093>
- Wang, Y., Jiang, X., Cao, R., Wang, X.: Robust indoor human activity recognition using wireless signals. Sensors (Basel, Switzerland) **15**(7), 17195–17208 (2015b). <https://doi.org/10.3390/s150717195>
- Wang, G., Zou, Y., Zhou, Z., Wu, K., Ni, L.M.: We can hear you with wi-fi. In: IEEE Transactions on Mobile Computing, vol. 15, no. 11 (2016a). <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7384744>

- Wang, X., Gao, L., Mao, S., Pandey, S.: CSI-based fingerprinting for indoor localization: a deep learning approach. CoRR <https://arxiv.org/abs/1603.07080> (2016b)
- Wang, H., Zhang, D., Wang, Y., Ma, J., Wang, Y., Li, S.: Rt-fall: a real-time and contactless fall detection system with commodity wifi devices. IEEE Trans. Mob. Comput. **16**(2), 511–526 (2017a). <https://doi.org/10.1109/TMC.2016.2557795>
- Wang, Y., Wu, K., Ni, L.M.: Wifall: device-free fall detection by wireless networks. IEEE Trans. Mob. Comput. **16**(2), 581–594 (2017b). <https://doi.org/10.1109/TMC.2016.2557792>
- Wang, Z., Guo, B., Yu, Z., Zhou, X.: Wi-fi CSI based behavior recognition: from signals, actions to activities. CoRR <https://arxiv.org/abs/1712.00146> (2017c)
- World Health Organization (WHO).: Fact-sheets: Falls. <http://www.who.int/news-room/fact-sheets/detail/falls>. Accessed 06 Oct 2019
- Xi, W., Zhao, J., Li, X., Zhao, K., Tang, S., Liu, X., Jiang, Z.: Electronic frog eye: counting crowd using wifi. In: IEEE INFOCOM 2014—IEEE Conference on Computer Communications, pp. 361–369 (2014). <https://doi.org/10.1109/INFOCOM.2014.6847958>
- Xie, Y., Li, Z., Li, M.: Precise power delay profiling with commodity wifi. In: Proceedings of the 21st Annual International Conference on Mobile Computing and Networking, MobiCom '15, pp. 53–64. ACM, New York (2015). <https://doi.org/10.1145/2789168.2790124>
- Xin, T., Guo, B., Wang, Z., Wang, P., Lam, J.C.K., Li, V., Yu, Z.: Freesense: a robust approach for indoor human detection using wi-fi signals. Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. **2**(3), 143:1–143:23 (2018). <https://doi.org/10.1145/3264953>
- Xu, W., Yu, Z., Wang, Z., Guo, B., Han, Q.: Acousticid: gait-based human identification using acoustic signal. Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. **3**(3), 115:1–115:25 (2019). <https://doi.org/10.1145/3351273>
- Yaşar, G.: Device-free human fall detection in indoor environments using channel state information of wi-fi signals. Master's thesis, Frankfurt University of Applied Sciences (2019)
- Yousefi, S., Narui, H., Dayal, S., Ermon, S., Valaei, S.: A survey on behavior recognition using wifi channel state information. IEEE Commun. Mag. **55**(10), 98–104 (2017). <https://doi.org/10.1109/MCOM.2017.1700082>



**Neena Damodaran** was a master student of High Integrity Systems at Frankfurt University of Applied Sciences. She received her Bachelor of Engineering degree in Computer Science from Nagpur University, India and is currently working as a Software Engineer. Her main interest lies in Artificial Intelligence and Machine Learning.



**Elis Haruni** is a master student of High Integrity Systems at Frankfurt University of Applied Sciences. Graduated from Polytechnic University of Tirana with a Bachelor degree in Computer Engineering. Main interest research machine learning and autonomous driving.



**Muyassar Kokhkhova** is a master student of High Integrity Systems at Frankfurt University of Applied Sciences. She received her Bachelor degree in Applied Mathematics and Computer Science in 2017 at Moscow State University. Her research interest is machine learning and applied mathematics.



**Jörg Schäfer** is a professor of Computer Science at Frankfurt University of Applied Sciences. He got his PhD degree from Ruhr-Universität Bochum (RUB) in Mathematical Physics in 1992. His main research interest is in applying machine learning and probabilistic models in ubiquitous computing applications. He runs a research group on ubiquitous computing applications.