

FallDeFi: Ubiquitous Fall Detection using Commodity Wi-Fi Devices

FallDeFi: 使用商用Wi-Fi设备进行无处不在的跌倒检测

SAMEERA PALIPANA, Nimbus Centre, Cork Institute of Technology, Ireland

DAVID ROJAS, Nimbus Centre, Cork Institute of Technology

PIYUSH AGRAWAL, United Technologies Research Centre, Ireland

DIRK PESCH, Nimbus Centre, Cork Institute of Technology

Falling or tripping among elderly people living on their own is recognized as a major public health worry that can even lead to death. Fall detection systems that alert caregivers, family members or neighbours can potentially save lives. In the past decade, an extensive amount of research has been carried out to develop fall detection systems based on a range of different detection approaches, i.e, wearable and non-wearable sensing and detection technologies. In this paper, we consider an emerging non-wearable fall detection approach based on WiFi Channel State Information (CSI). Previous CSI based fall detection solutions have considered only time domain approaches. Here, we take an altogether different direction, time-frequency analysis as used in radar fall detection. We use the conventional Short-Time Fourier Transform (STFT) to extract time-frequency features and a sequential forward selection algorithm to single out features that are resilient to environment changes while maintaining a higher fall detection rate. When our system is pre-trained, it has a 93 % accuracy and compared to RTFall and CARM, this is a 12 % and 15 % improvement respectively. When the environment changes, our system still has an average accuracy close to 80 % which is more than a 20 % to 30 % and 5 % to 15 % improvement respectively.



CCS Concepts: • Human-centered computing → Ubiquitous and mobile computing systems and tools;

Additional Key Words and Phrases: Wi-Fi, Activity recognition, Feature extraction, Device-free

ACM Reference Format:

Sameera Palipana, David Rojas, Piyush Agrawal, and Dirk Pesch. 2017. FallDeFi: Ubiquitous Fall Detection using Commodity Wi-Fi Devices. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 1, 4, Article 155 (December 2017), 25 pages. <https://doi.org/10.1145/3161183>

155

1 INTRODUCTION

With the increase in life expectancy, the world's population of older (or elderly) people, defined as above 65 years of age, has been steadily growing and compared to the age group 20-64 will reach 25% by 2040 [1]. Each year, one third of older people over 65 years fall, resulting in injuries and unfortunately sometimes even death [2]. Moreover, many older people are also not able to stand up without external help after a fall. Reports suggest that the event of a fall has been fatal for half of the elderly who were unable to get up for an hour or more even without direct injuries [3]. Therefore, an effective fall detection mechanism has the potential to save the lives

This work has been funded by the Irish Research Council in collaboration with United Technologies Research Centre, Cork, Ireland.
Author's addresses: S. Palipana, D. Rojas and D. Pesch, Nimbus Centre Cork Institute of Technology, Cork, Ireland; P. Agrawal, United Technologies Research Centre, Cork, Ireland;

Emails: dhanapala.palipana@mycit.ie, f.rojas-calvente@mycit.ie, dirk.pesch@cit.ie, AgrawaP@utrc.utc.com .
Authors' addresses: Sameera Palipana, Nimbus Centre, Cork Institute of Technology, Ireland; David Rojas, Nimbus Centre, Cork Institute of Technology; Piyush Agrawal, United Technologies Research Centre, Cork, Ireland; Dirk Pesch, Nimbus Centre, Cork Institute of Technology.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2017 Association for Computing Machinery.

2474-9567/2017/12-ART155 \$15.00

<https://doi.org/10.1145/3161183>

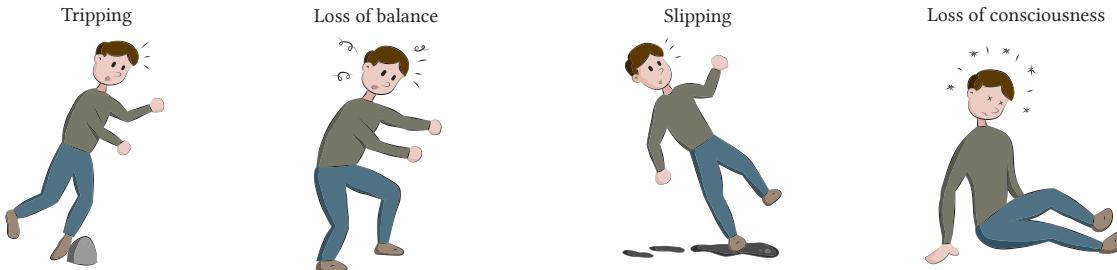


Fig. 1. An example of different types of falls. [©David Rojas]

of older people, especially those who are living independently. Figure 1 shows an example set of falls that can frequently occur in domestic environments such as tripping, loss of balance, slipping and loss of consciousness. These falls can have different variations including forward, backward, lateral and on position falls [4, 5].

Due to the importance of fall detection several competing detection technologies exist, which can be broadly classified as wearable and non-wearable. Among wearable technologies a range of fall sensors have been proposed such as ECG, accelerometers, RFID, gyroscopes, pressure sensors and smart phones. Non-wearable technologies include ambient sensors, computer vision, passive infra-red sensors, and wireless technologies. Wearable technologies require the sensors to be attached to the human body. While wearing sensors can be very effective, in particular in outdoor environments, it may not always be complied with by older people. More importantly, among elderly who live independently, about half of the falls occur inside their own premises [6], which suggests the need for a reliable non-wearable indoor fall detection scheme. Among non-wearable technologies, computer vision based approaches are usually highly accurate, however, they have issues especially in regard to occupant privacy [7] and occlusion [8]. Ambient sensing approaches proposed in the literature generally include microphone arrays, pressure sensors and vibration sensors and measure changes in the ambience to detect falls. However, other sources of pressure or sound in the environment can cause false alarms in these systems [9]. Wireless technologies include radar, wireless communication signal information such as WiFi signals, and wireless sensor networks. For a comprehensive survey on wearable/non-wearable technologies we refer the interested readers to the following publications [10–12].

In this paper, we propose FallDeFi, a fall detection technique based on WiFi signals as the enabling sensing technology. WiFi has the advantage of being ubiquitous and low cost compared to radar for example. More specifically, we use the amplitude of WiFi Channel State Information (CSI) provided by commodity WiFi devices as the signal strength descriptor to detect falls. WiFi devices based on the IEEE 802.11a/g/n/ac standards use Orthogonal Frequency Division Multiplexing (OFDM) as the modulation scheme with multiple sub-carriers in a WiFi channel and multiple antennas as a solution for frequency selective fading. The receiver measures a discrete Channel Frequency Response (CFR) in time and frequency as phase and amplitude in the form of CSI for each antenna pair. CSI is a better signal strength descriptor than the standard received signal strength indicator (RSSI) because of high resolution and sub-carrier diversity [13]. As not all commercially available IEEE 802.11 chipsets currently allow access to CSI, we use the Intel 5300 network interface card (NIC) in our work as this is the most commonly used device from which CSI can be extracted from the physical layer.

Although our work is not the first, fall detection using WiFi CSI is a relatively new topic. Prior work on fall detection using CSI have considered both time domain and time-frequency analysis and in this work we follow the latter approach because time-frequency feature extraction and analysis has significant advantages compared to pure time domain or frequency domain analysis. Even though existing WiFi activity recognition systems classify falls among other generic activities, these systems do not analyse the robustness of fall detection

among activities that can be mistaken as falls [14]. As an example, daily activities like sitting down, standing up, jumping, and bending and picking up objects have similar time-frequency patterns that can mislead the system. Therefore, specialized features that distinguish these activities are required especially given that fall detection is closely related to safety and health of elderly people. Additionally, these features have to be resilient to changes in the environment enabling easy deployment of a pre-trained system and not require retraining due to changes in the environment. In FallDeFi, we combine fall detection of commodity WiFi devices with well established signal processing and feature extraction techniques from radar based fall detection to create a low cost, reliable and environment resilient fall detection approach. In summary, the main contributions of this paper are as follows.

- (i) We provide a novel fall detection system using time-frequency analysis of WiFi CSI that includes a CSI noise filtering technique for feature extraction in high frequencies.
- (ii) We use a rigorous feature selection procedure to extract features in classifying a wide range of falls e.g. trip, slip, lose consciousness, lose balance from similar activities that can get confused with falls e.g. sitting down (on a chair, on the floor), standing up (from a chair, from the floor), bend and pick up object and jumping.
- (iii) We develop a time-frequency based pre-screener for event detection prior to detecting falls and use its outputs such as event duration as features in fall detection.
- (iv) We evaluate the performance of the proposed system through experiments using human subjects considering different environments, link types and up to two persons using the data collected from commodity WiFi devices. When the system is pre-trained, FallDeFi achieves close to 93% average accuracy and when the environment changes, our system still has an average accuracy close to 80%.
- (v) We compare our results to the current state of the art systems [9, 14] and achieve greater than 8 % and 10% accuracy improvement when the system is pre-trained and accuracy improvements of 5 % to 15 % and 20 % to 30 % when environment changes occur respectively.

The rest of this paper is organized as follows. We first review the related work in § 2 and then we provide an overview of the FallDeFi system architecture in § 3. We explain our CSI preprocessing steps in § 4, spectrogram processing steps and the pre-screener for event detection in § 5, and feature extraction and selection of environment independent features in § 6. Finally, we present results of an evaluation our system in four different environments in § 7, discuss limitations and possible improvements in § 9, and conclude in § 10.

2 RELATED WORK

Fall detection can be treated as a specific type of activity recognition because it involves similar signal processing techniques. We therefore review in this section related activity recognition and fall detection schemes by dividing them into four categories: i) RSSI-based activity and fall detection, ii) radar-based activity and fall detection iii) CSI-based activity recognition, and iv) CSI-based fall detection.

RSSI-based Activity and Fall Detection. Among these works, Mager et al. [15] use radio tomographic imaging using IEEE 802.15.4 compliant radios deployed at two different levels (two links at different heights, radios are along the boundary of the observed space) to detect falls from the time difference in shadowing of the two link levels. WiGest [16] and Harmony [17] use RSSI from commodity WiFi devices extracting time-frequency features to classify hand gestures and generic human activities respectively. These schemes depend on signal strength fluctuations caused by human activities similar to how CSI variations are used in our system. However, it is well known that RSSI measurements are fundamentally limited by their low resolution and lack of diversity in the number of available streams compared to CSI, which impacts the overall accuracy [13, 18] of the approach.

Radar Activity and Fall Detection. From radar-based activity recognition approaches, we borrowed techniques for time-frequency signal processing and analysis by Kim and Ling [19] and the extraction of gait related features such as leg and torso speeds by Van Dorp and Groen [20].

FallDeFi is related to radar fall detection schemes, especially with STFT-based time frequency analysis [21–23], Doppler time frequency analysis [4, 24] and Power Burst Curve (PBC) analysis [25], which have been tested with different types of radars and under different sensing conditions. Radars operate with higher bandwidths (in the range of GHz) and use dedicated hardware, which allows for higher location and frequency resolutions. In contrast, FallDeFi uses cheap commodity WiFi devices in lower bandwidth (20 MHz), yet, with effective CSI signal processing we are also able to extract the same time-frequency features in order to classify falls with high accuracy.

WiFi CSI in Activity Recognition. Among these techniques, E-eyes [26] detects generic activities of a single person using location oriented time domain features. CARM [14] classifies generic activities of a single person using time-frequency features such as torso and limb velocities by characterizing the changing speed of the reflected path length on sub-carrier amplitudes. These velocities are quite similar to the velocities obtained from Doppler Radars without the movement direction. WiFiU [27] extends CARM to recognize human gait from limb and torso velocities. WiDance [28] extracts Doppler shifts using commodity WiFi devices without any hardware modifications to classify exergamers related gestures. They exploit antenna diversity to retain relevant Doppler shifts which enables the extraction of motion directions. CARM is closely related to our work because it includes falls in a multi class classification problem and uses similar signal processing techniques to ours. In contrast, CARM classifies generic activities among falls while we specialize only on fall detection, in essence, we treat fall detection as binary classification and include a wider range of falls, and non-falls that can be easily mistaken as falls. Therefore, our features are more refined to classify falls through feature selection. Furthermore, we evaluate the robustness of the proposed method in multiple environments with and without an additional person and show better performance in detecting falls and also resilience to environment changes. Although our intended application is different from WiFiU and WiDance, our CSI signal processing techniques build upon CARM, WiFiU and WiDance with the necessary adjustments for high frequency falls in noisy environments. Especially, CARM and WiFiU inspired us to extract environment independent features for fall detection.

WiFi CSI in Fall Detection. To the best of our knowledge, all existing CSI-based fall detection schemes extract time domain features, the most influential of them being WiFall [29, 30] and RTFall [9]. WiFall uses CSI amplitude related time domain features to characterize falls of a single person. RTFall uses both CSI amplitude and phase and builds upon the features extracted in WiFall to increase the fall detection rate and reduce the false positives. In contrast, we use time-frequency analysis using only the CSI amplitude, which requires a different set of signal processing techniques. WiFall classifies four types of activities, fall, sit, walk and stand and uses multi-class classification to distinguish those activities. From this perspective, WiFall is primarily an activity recognition scheme specialized on fall detection. In contrast, RTFall distinguishes falls from nine other activities through binary classification. In addition to the fall like activities mentioned in RTFall, we consider four variations of falls such as tripping, slipping, loss of balance, and loss of consciousness and their sub-variations such as forward, backward, lateral and on position falls to avoid over-fitting to a particular type of fall. Both RTFall and WiFall are evaluated based on a scenario where a single elderly person is living independently, whereas we consider two cases, a situation where the person is alone and a situation where there is another person involved in regular activities or is motionless. Both make our evaluations more extensive than the previous works. More importantly, FallDeFi classifies falls from features robust to environment changes which is not addressed by the current CSI-based fall detection research.

3 FALLDEFI OVERVIEW

Our proposed system, FallDeFi is a non-wearable/device-free, indoor fall detection system that uses commodity WiFi devices as the physical fall monitoring infrastructure. FallDeFi continuously monitors the occupant activities by processing CSI captured from WiFi signal receivers to identify falls. When it detects an activity, it classifies

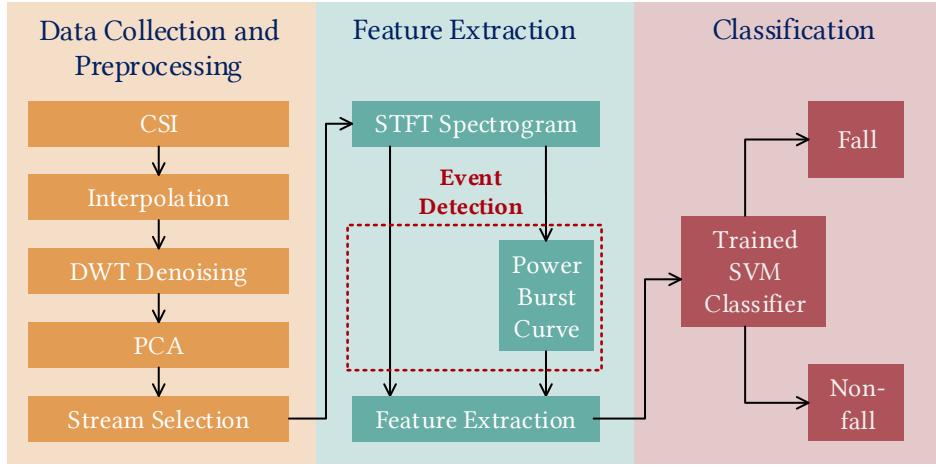


Fig. 2. FallDeFi system architecture.

this as either a fall or a non-fall using a pre-trained model. However, this model has to be accurate and resilient to variations in the channel caused by changes in the monitored environment either as a result of changes in furniture locations, other big items in the space or even WiFi transceiver/antenna positions. To address these requirements, our system, illustrated in Figure 2 is built in a modularized manner: (i) Data Collection and Preprocessing, (ii) Feature Extraction and (iii) Fall Classification. In this section, we summarize the objectives of each module and the components therein.

The Data Collection and Preprocessing module is responsible for collecting the raw CSI signals from WiFi devices and processing them to obtain a clean and informative spectrogram for time-frequency analysis. To this end, this module addresses three issues that occur with CSI capture: (i) obtaining a fixed rate of CSI amid packet losses due to weak signals in through-wall and non line of sight (non-LoS) links, (ii) removing noise in the sub-carriers without losing high frequency components especially when the signal is weak, and (iii) extracting the most effective streams from the sub-carriers as input to the spectrogram for increased computational efficiency. To solve these issues we use linear interpolation, Discrete Wavelet Transform (DWT) based noise filtering and Principal Component Analysis (PCA) based stream decorrelation and selection respectively.

The Feature Extraction module has three main functions: (i) detection of start and end of an event, this component should detect 100% of human induced events, (ii) extraction of features during the detected event interval, and (iii) selection of features that classify events. Since we use a spectrogram to extract features, we detect events from a Power Burst Curve (PBC) which is commonly used in radar fall detection to detect events in a spectrogram. Once an event is detected, we extract features both from the spectrogram and the PBC. While the spectrogram provides spectral features, the PBC provides time and amplitude related features. Once the features are extracted, we use sequential forward selection to determine the environment independent features and feed them to the classifier. Finally, the Fall Classification module classifies the detected events into falls and non-falls using the extracted features. Algorithm 1 provides the detailed flow of events and the inputs and outputs of each module from the moment CSI is collected until falls are detected, as implemented in FallDeFi.

4 CSI SIGNAL PROCESSING

Due to weak signals in certain links caused by non line of sight connections and wall penetration in indoor environments, we experience some packet losses. In order to obtain the same sample rate for all the traces, we

ALGORITHM 1: FallDeFi, fall detection algorithm

```

input : $CSI_{raw}$  - CSI measurements of a  $t_{int}$  second interval from  $m \times n$  antenna pairs
output: Fall detection
1 Initialization :  $\hat{N}_{th}, PBC_{th}$  ;
2 foreach antenna pair do
    3 foreach subcarrier do
        4      $CSI_{interp} \leftarrow$  LinearInterpolate ( $CSI_{raw}$ );
        5      $CSI_{wden} \leftarrow$  WaveletDenoise ( $CSI_{interp}$ );
        6 end
        7     PCs  $\leftarrow$  PCA ( $CSI_{wden}$ );
        8     selected PCs  $\leftarrow$  SelectOptimumPrincipalComponents (PCs);
        9 foreach selected PC do  $S_{raw}(n,k) \leftarrow$  ComputeSpectrogram (selected PC);
10 end
11      $S_{avg}(n,k) \leftarrow$  AverageAllSpectrograms ( $S_{raw}(n,k)$ );
12      $S(n,k) \leftarrow$  ProcessSpectrogram ( $S_{avg}(n,k)$ );
13      $PBC(n) \leftarrow$  ConstructPowerBurstCurve ( $S(n,k)$ ) ;
14 if  $PBC(n) > PBC_{th}$  then // Fall-like event detection
    15     feature vector  $\leftarrow$  ExtractFeatures ( $PBC(n), S(n,k)$ );
16     Fall  $\leftarrow$  SVM(trained model, feature vector) // Classify falls and non-falls
17 end

```

make use of linear interpolation as the first step. Here we set our transmission rate (1000 Hz) well above the rates usually required for a successful fall detection (the Nyquist sampling frequency of a fall is < 350 Hz as in Eq. 1 for WiFi at 5.2 GHz and even lower at 2.4 GHz), and then interpolate signal strengths on individual sub-carriers to counter the loss in information due to packet loss.

As mentioned in the introduction, we use Intel 5300 NICs to extract CSI from WiFi receivers. CSI sub-carriers from these devices consist of additive white Gaussian noise [31]. When sub-carrier streams are decorrelated through principal component analysis (PCA), noise present in the sub-carriers makes the first principal component (PC) unusable. Therefore, some research work has neglected the first PC and relied only on the information in other principal components [14, 32]. However, as the first PC contains the most information as it corresponds to the highest eigenvalue among all the principal components, neglecting this could result in losing information. First, in §4.1 we introduce our noise filtering approach and then in § 4.2 discuss its impact on the first PC.

4.1 Noise Filtering

Initially, we considered several time and frequency domain noise filtering approaches. Time domain approaches such as median or mean filters can distort the signal, and vital high frequency components may also be lost. Although frequency domain approaches like bandpass filters (e.g. Butterworth) remove the noise well for narrow bands, this is an out-of-band noise filtering technique. If a bandpass filter with a larger passband is used, the residual noise will have a higher contribution and a smaller passband cuts off vital signal components. Since CSI has noise in all bands, we chose Discrete Wavelet Transform (DWT) which is an in-band noise filtering technique. Through careful parameter selection, DWT filtering eliminates in-band noise, preserves high frequency components and introduces less distortions to the signal.

In DWT-based noise filtering, by transformation into the wavelet domain the signal is divided into multiple frequency levels called *wavelet levels* that consist of *approximate* and *detailed* coefficients. The highest wavelet level is considered noise. In DWT, first, noise and the threshold for that level is estimated, the threshold is adapted

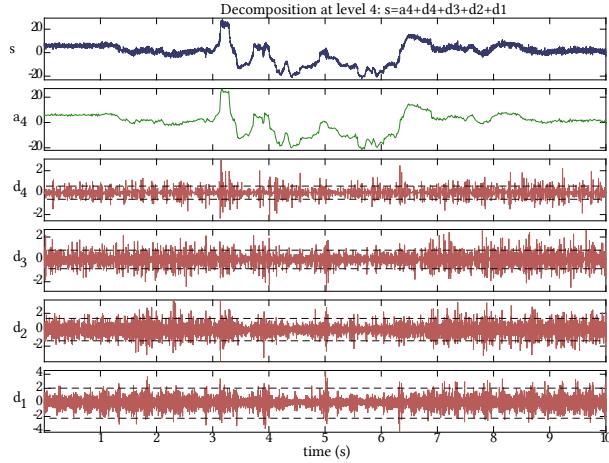


Fig. 3. DWT filtering used in this work. s is the noisy signal and a 4 level sym3 wavelet is applied to s . d_1, d_2, d_3 and d_4 are detailed coefficients and a_4 is the approximate coefficient obtained from wavelet transform. The thresholds of detailed coefficients are marked in the dashed blue line.

to lower wavelet levels, and then noise is cut off in all wavelet levels without significant distortions to the signal component. Finally, the de-noised signal in the wavelet domain is transformed back to the time domain. As an example, for a Nyquist sampling rate of 1000 Hz, assuming that human falls do not occur beyond 250 Hz, the highest wavelet level 250 Hz to 500 Hz is considered noise. Since the maximum velocity of a falling human just before hitting the ground is typically 5 ms^{-1} , applying the Doppler frequency formula [4],

$$\delta f_{max} = \frac{2v}{c} f_c \quad (1)$$

to a 5.2 GHz WiFi signal we get a maximum Doppler frequency of $\delta f_{max} \approx 175 \text{ Hz}$, which is well below 250 Hz. Thus all frequency components above 250Hz can be removed, however, to avoid significant distortions in other frequencies, not all high-frequencies components are removed but only those with limited influence on the signal. This is illustrated in Figure 3. The parameters that are initialized for wavelet denoising are discussed in § 7 under *System calibration*.

Figures 4a and 4b illustrate the transformation of a noisy sub-carrier amplitude to a noiseless signal through the selected strategies. The effect of noise filtering on the first principal component is illustrated in Figure 5. We further illustrate the effect of this noise filtering approach on fall classification accuracy in § 8.1.2.

4.2 Sub-carrier Decorrelation and Stream Selection

A single antenna pair provides 30 streams of CSI from 30 sub-carriers and they are mostly correlated, especially when there is an activity [14]. However, for time-frequency analysis, we require as few streams as possible to reduce the computational overhead in constructing the spectrogram for feature extraction. Methods like averaging a set of streams or selecting a subset of streams do not capture the variations in all the sub-carriers effectively. Therefore, we decorrelate the CSI streams using PCA and obtain the minimum number of resulting orthogonal streams, i.e, the principle components that capture $x\%$ of the variance contained in the 30 sub-carriers.

In PCA [33], a linear transformation ensures the projection of CSI sub-carrier amplitudes to a new orthogonal coordinate system such that the highest variance is present in the first coordinate, the next highest variance is present in the second coordinate, and so on. The linear transformation in PCA ensures that the cumulative variance in the resultant coordinates is equal to the cumulative variance in all the original sub-carriers. These

coordinates are called the principal components (PCs), and the amount of variance in a principle component is explained by the respective eigenvalue obtained through PCA. Due to the DWT-based noise filtering and PCA as per § 4.1, the first two to three PCs are, on average, sufficient to capture the majority of the variance in the original 30 sub-carriers. We select the number of PCs for a particular activity to satisfy the equation,

$$\text{Captured variance} = \frac{\text{Cumulative variance of selected PCs}}{\text{Cumulative variance of all 30 sub-carriers}} \quad (2)$$

The selection of PCs satisfying the value of *captured variance* as in Eq. 2 is empirical and it is described in § 8.1.3. As PCA is an unsupervised machine learning approach, selecting a fixed number of PCs can include noisy components for some activities and lesser information for others. Unlike previous research that use a fixed number of PCs [14, 27], our approach selects the required minimum number of PCs dynamically for each activity depending on the number of PCs that satisfy the predefined captured variance.

Figures 4 and 5 show how the noise filtering at sub-carrier level has a higher impact on the noise level of the first principal component than on the sub-carrier. As highlighted in § 4.1, the figure shows that the noise present in raw sub-carriers can result in high frequency noise in the first principal component, especially on occasions when there are no activities. Additionally, we observe that for noisy links, high frequency noise has a higher impact on the first principal component compared to the signal itself. Figure 6 compares the first three principal components for the same activity when de-noising is applied before PCA. It also shows that the majority of the human activity induced variation is concentrated in the first and second principal components, but in the third principal component (and onwards) the noise level begins to have a higher influence.

5 TIME FREQUENCY ANALYSIS

5.1 Spectrogram and its Parameter Selection

After CSI data collection and preprocessing, a transformation to the time-frequency domain is necessary to perform the feature extraction. For time-frequency analysis, various linear and non-linear techniques exist especially among radar fall detection [3] approaches. Among these, non-linear methods tend to distort the frequency components generated by falls. Additionally, among the various time-frequency feature extraction methods, STFT spectrogram features have outperformed DWT features in recent radar fall detection research [34]. Therefore, we selected STFT for our time frequency analysis.

In STFT, the frequency resolution is inversely proportional to the time resolution. Hence, this requires us to find an optimal window size to obtain satisfactory time and frequency resolutions for our application. With CSI measurements for human falls having frequencies up to a theoretical maximum of 175 Hz in the 5 GHz band, and as changes in time occur in the sub-second range, we opt for an FFT window size of 512 samples at a sampling rate of 1000 pkts/s. We chose the overlap size of two windows to be 256 samples for two reasons: (1) it is the optimum overlap size for the Hamming window used in FFT, (2) higher value increases the computational efforts as it introduces high interpolation. The selected parameters provide us with a frequency resolution of $\frac{\text{sample rate}}{\text{FFTsize}} \approx 2 \text{ Hz}$ and a time resolution of $\frac{\text{window-overlap}}{\text{sample rate}} \approx 0.1 \text{ s}$.

First, we obtain the optimum number of principal components that capture 95 % of total variance of all the sub-carriers. Then we apply STFT on the selected principal components individually and average the spectrograms to obtain a single spectrogram for each activity according to [14, 27]. By averaging the spectrograms we can obtain a spectrogram with a higher amount of information, as the principal components are orthogonal to each other and each of them consist of unique frequency components. The spectrograms obtained through this process for four types of falls (trip, slip, lose balance and lose consciousness) and other activities (sit, stand, bend and pick an object, jump, walk and no event) are illustrated in figure 7.

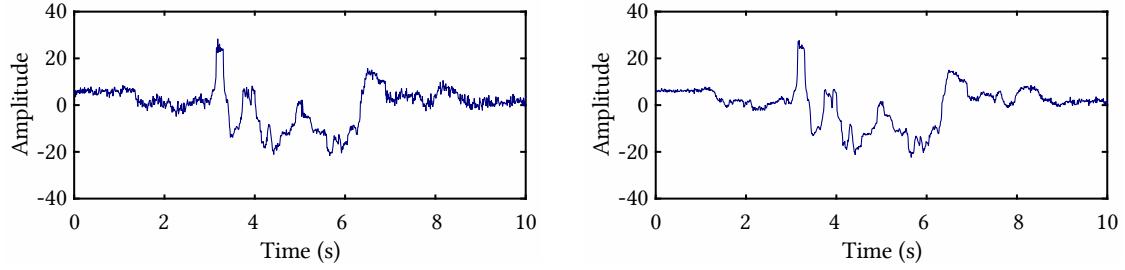


Fig. 4. Comparison of the amplitude variation of 1st subcarrier before and after wavelet denoising

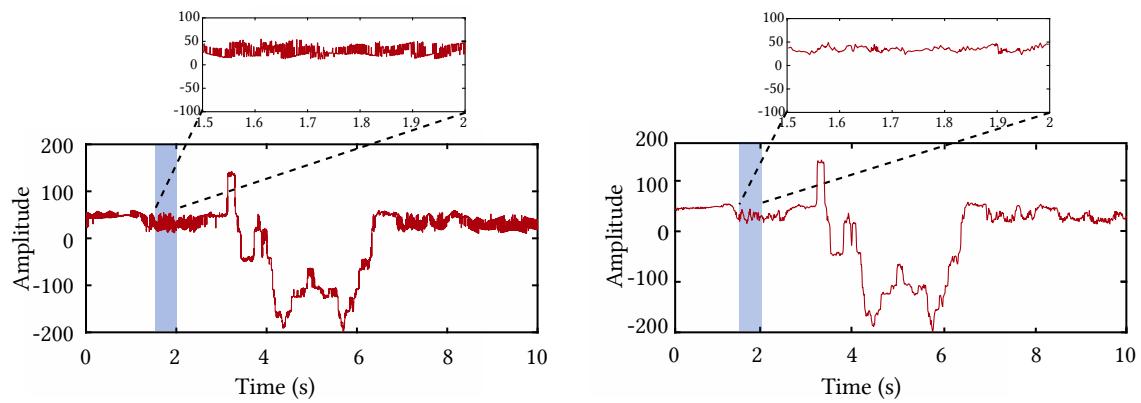


Fig. 5. Effects of de-noising are more prominent in PC 1 of Fig. 5b than in individual sub-carriers in Fig. 4b.

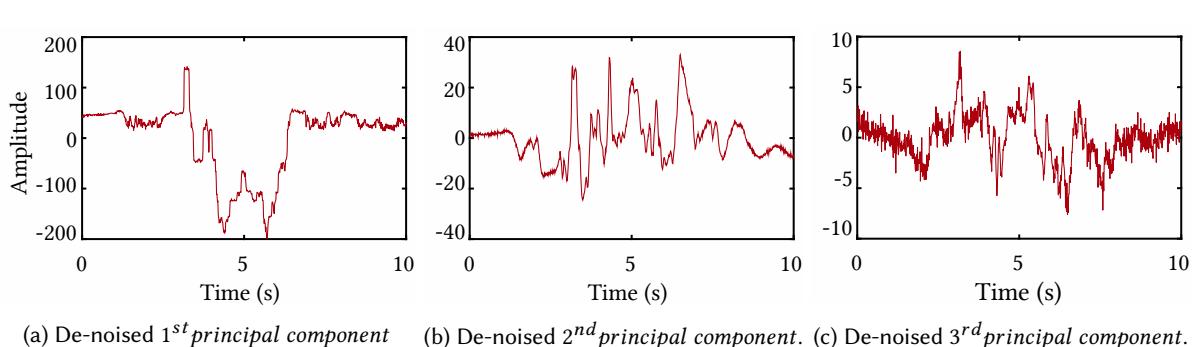


Fig. 6. Comparison of the resultant first three PCs after wavelet de-noising and then PCA. First and second PCs are less noisy, yet in the third PC (and seemingly onwards), the noise level has a higher influence.

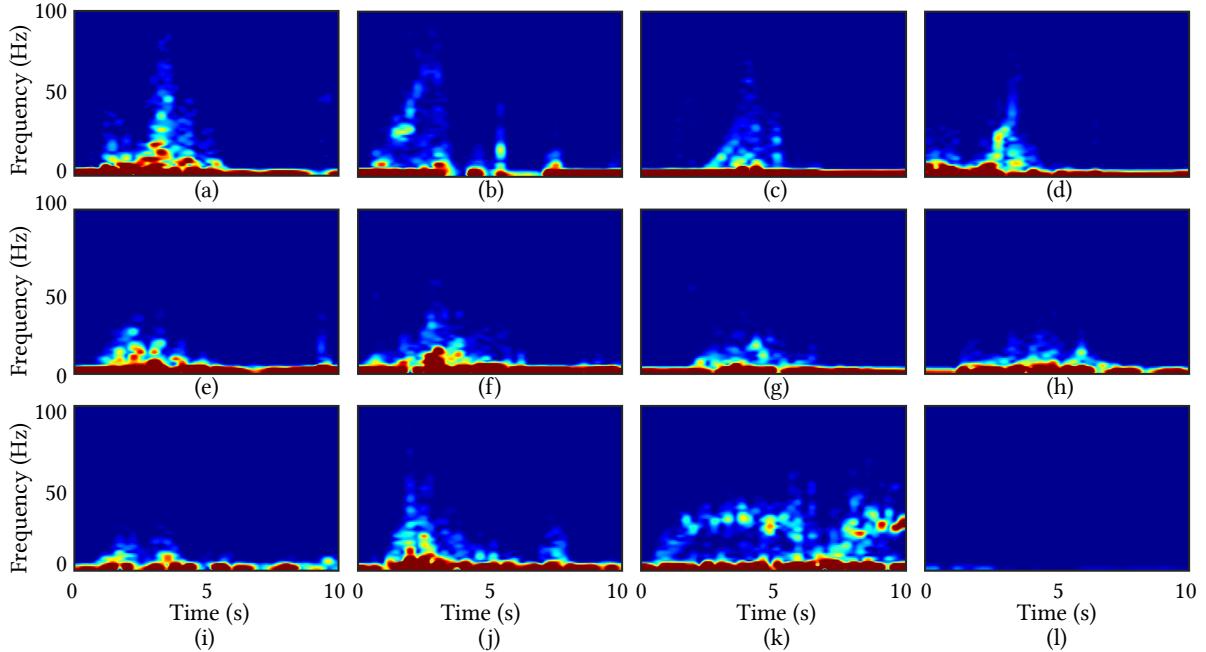


Fig. 7. Comparison of spectrograms for falls and fall like activities. (a) Trip, (b) Slip, (c) Lose balance, (d) Lose consciousness, (e) Sit down on the floor, (f) Stand up from the floor (g) Sit down on a chair, (h) Stand up from a chair, (i) Bend and pick up an object, (j) Jump, (k) Walk and (l) No event

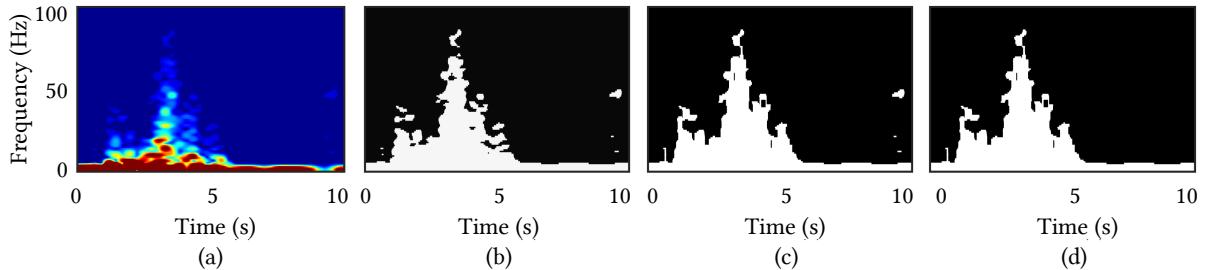


Fig. 8. (a) Raw spectrogram image, (b) Binary image of the raw spectrogram, (b) Binary image after Morphological processing, (d) Binary image after disconnected region removal

5.2 Spectrogram Processing

Before extracting the features corresponding to falls and fall-like activities, the main event has to be isolated to obtain a clear spectrogram that characterizes only the falls and fall-like events. Therefore, we treat time-frequency energies away from the main event as noise and remove them from the spectrogram.

In the literature, spectrogram thresholding is a common technique that is used to segment spectrograms [20, 24, 35, 36]. To determine the threshold, we estimate the noise at high frequencies of the spectrogram. When we construct the histogram of the amplitudes at high frequencies (>250 Hz), we observe that this can be approximated with a Gaussian distribution which is consistent with previous time-frequency analysis work [19]. This is due to the fact that events do not occur beyond 175 Hz and this region is dominated by the residual noise. If low

frequency amplitudes were also considered, the histogram would deviate from the Gaussian distribution and would become more skewed, due to the effects of human movement events at low frequencies. Hence, we only chose frequencies beyond 250 Hz to estimate the spectrogram segmentation threshold \hat{N}_{th} . Then, the spectrogram segmentation threshold can be defined as

$$\hat{N}_{th} = \mu + k\sigma \quad k \in R \quad (3)$$

where μ is the mean and σ is the standard deviation of the Gaussian approximation of the noise histogram from 250 Hz upwards. The value k is selected to tune the spectrogram segmentation threshold for different environments so that the weak energy components around the extreme frequencies of the main event are preserved and noise energy is filtered (Figure 8). We determined an average value for k by measuring noise in several spectrograms for a particular link.

$$S(n, k) = \begin{cases} S(n, k) & \text{if } S(n, k) \geq \hat{N}_{th} \\ 0 & \text{if } S(n, k) < \hat{N}_{th} \end{cases} \quad (4)$$

After spectrogram segmentation, disconnected regions may appear due to weak reflections off the limbs or other minor motions during a fall, which can cause weak energy components to be scattered around the extreme frequencies of the main event. For this reason, we performed morphological dilation [22, 37] to connect close yet broken spectrogram segments that are within a predefined frequency. The regions that were not connected are then removed from the main event region of the spectrogram, specified as disconnected region removal. This procedure is illustrated in Figure 8.

5.3 Event Detection

A fall event occurs typically within a 2 to 3 s period [4, 9], comprising high frequency/energy components that are not so apparent in other activities, which can also be accompanied by low frequency/energy events prior to and after a fall. Therefore, it is important to localize the fall from other events to feed the fall specific features to simplify the fall classification. Thus, the event detection step should be able to effectively localize only the high energy/frequency fall and fall-like activities. For these reasons, the typical time domain event detection methods cannot be used. Instead, we use a power burst curve (PBC) [34] to detect and localize high frequency events, adapted from Doppler radar fall detection techniques. Unlike Doppler radars, our spectrogram $S(n, k)$ does not associate negative frequencies as we do not consider the motion direction. This enables us to define the PBC only for positive frequencies

$$PBC(n) = \sum_{k=k_l}^{k_u} |S(n, k)|, n = 1, 2, \dots, N \quad (5)$$

where k_l is the lower frequency bound and k_u is the upper frequency bound. Even though a fall may have frequency components in excess of 100 Hz in the 5.2 GHz WiFi band, the strongest signal energy of high frequency events including falling and other similar activities are concentrated between 5 Hz to 30 Hz. The frequency components within this range correspond to reflections from a large body of mass like the torso. Therefore, we set $k_l = 5$ Hz & $k_u = 25$ Hz, sum the signal powers between this range and when this exceeds a predefined threshold, PBC_{th} , it is considered as a fall-like event. The threshold, PBC_{th} is determined as

$$PBC_{th} = \sum_{k=k_l}^{k_u} \hat{N}_{th} \quad (6)$$

where \hat{N}_{th} is the estimated noise threshold in Eq. 3, which is dependent on the link. In this case we simply determine the power burst curve threshold PBC_{th} by summing the noise amplitude, \hat{N}_{th} , estimated in previous section within the 5 to 25 Hz frequency range. This process is illustrated in Figures 9a and 9b. From this

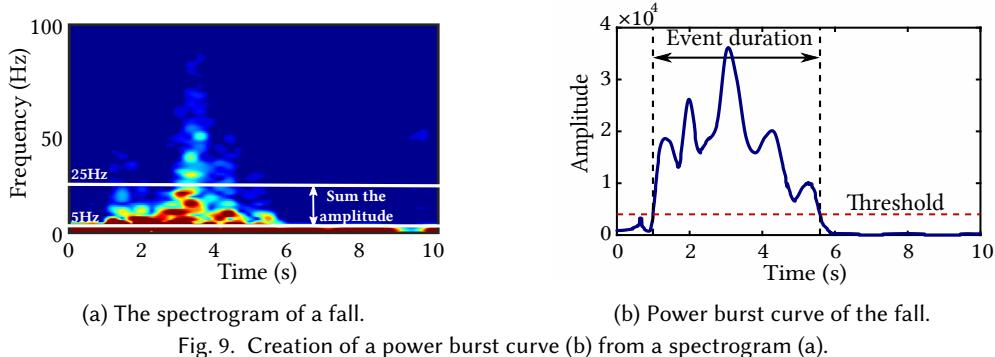


Fig. 9. Creation of a power burst curve (b) from a spectrogram (a).

prescreener, we are able to detect falls and other high frequency events with a 100% success rate. Unlike time domain prescreeners, the main advantage of this approach is the isolation and detection of events within a specific frequency range.

6 FEATURE EXTRACTION AND CLASSIFICATION

In this section we explain the set of features that were extracted from the spectrogram for classification. From previous research that involve time-frequency features, we chose a pool of features that have been successful and tested them on the traces we collected for both falls and non-falls. We tested them on our traces in two ways. First we identify which features correctly classify the falls from our traces, we call these features *Original Features* (*OFs*). From these features, we identify a feature subset which is more robust when environment changes occur and call them *Selected Features* (*SFs*). To select these two types of features, we use a sequential forward selection algorithm [2]. We use two criteria to select *OFs* and *SFs* in this algorithm. For *OFs*, the criterion is the highest accuracy when trained and tested on the same data. For *SFs*, the criterion is highest accuracy when trained and tested on two different data sets. Table 1 illustrates both sets of features and the classification performance of these two features are explained in § 8.2.

6.1 Original Features

As shown in Table 1, the extracted *spectral features* are as follows. (i) *Extreme frequency features* [22, 34]: mean, standard deviation and maximum of the extreme frequency curve. (ii) *Torso frequency features*: mean, standard deviation and maximum of the torso frequency curve. We use the percentile method to obtain the torso frequency

Table 1. Extracted features from sequential forward selection

	Spectral features	Power burst curve features
Original features	extreme frequency features, torso frequency features, ratio of max. extreme and max. torso freqs, spectral entropy, fractal dimension	event duration, ratio of energy above and below the PBC threshold
Selected features	spectral entropy, fractal dimension	event duration

curve which is the least affected method by noise and other limb motions according to [20]. (iii) The ratio between maximum extreme frequency and maximum extreme torso frequency. Falls have both higher extreme frequencies and higher torso frequencies. (iv) *Spectral entropy*. This is a normalized feature and measures the textural properties of a fall (randomness in the distribution of energy in a spectrogram) [38]. As an example, a fall has a higher entropy than other events at both lower and higher frequencies due to a high fluctuation in amplitudes. We compute the spectral entropy, H as: $H = -\sum_{i=k_l}^{k_u} p(n_i) \ln p(n_i)$ where k_l and k_u are upper and lower frequency bounds. $p(n_i)$ is the normalized power spectral density $p(n_i) = \frac{\hat{P}(k, n_i)}{\sum \hat{P}(k, n)}$ where $\hat{P}(k, n) = \frac{1}{N} |S(n, k)|^2$ of spectrogram amplitudes $S(n, k)$ mentioned in § 5.2. Even though we picked a range of frequencies initially, the frequency ranges that produced the best fall classification are the following: (a) 1 to 10 Hz, (b) 10 to 30 Hz and (c) 30 Hz to max frequency. (v) *Fractal dimension*: This feature, calculated using the Hausdorff dimension method, measures the roughness of the extreme frequency curve during the detected event interval. Since a fall includes whole body movements, we observe that the extreme frequency curve has a chaotic nature. Additionally, this is a normalized feature which can extract features robust to environment changes.

The extracted *power burst curve* features are as follows: i) *Event duration*: Typically a fall has a duration of 2 to 3 s period [4, 9], whereas other events have a range of durations depending on the activity. This is a characteristic unique to a fall which is independent of the environment and additionally, it can distinguish other events that have high frequency components yet the duration is either longer or shorter in length. ii) *Ratio of energy above and below the PBC threshold*: Some events have high frequencies but low energy in those frequencies, however, falls have high powers in high frequencies as well as a significant amount of energy in those frequencies. This helps to separate falls from other such activities.

6.2 Selected Features

One of the major goals of our fall detection scheme is to classify falls with high robustness to environment changes. As shown in Table 1, the feature subset that best satisfied this criterion are: *event duration*, *spectral entropy* (1 to 10 Hz & 10 to 30 Hz) and *fractal dimension*. Note that spectral entropies above 30 Hz are not being used for classification. For positions that are far from the link, especially when the link is weak due to non line of sight conditions, even when high frequency components are produced during falls, they are not captured by the system. As it is visible in Figure 7, low frequency components have higher energy than higher frequency components. When falls occur far from the link, amplitudes of higher frequencies are further weakened. Therefore, such falls tend to get misclassified with fast sitting down or standing up of stronger links if. This resulted in spectral entropies above 30 Hz being not robust to all environments. However, fractal dimension of the extreme frequency curve still produced a robust classification during environment changes. We explain the classification results of the selected features compared to originally extracted features in § 8.2.2.

7 EVALUATION SETUP

In this section we present the environments that we performed the experiments in, the hardware and software tools, the types of data sets that were gathered, the type of activities that were conducted to distinguish falls, and finally, the evaluation metrics we used to measure the performance of FallDeFi.

Experimental Environments. To evaluate the performance of our fall detection system over diverse conditions, we performed falls and other activities in several environments. Figure 10 illustrates our experimentation environments. Marked in *yellow* crosses is the area that both falls and other activities are performed. To avoid injuries to the volunteers due to the limited space, in the area marked in *blue* crosses, other activities are performed. As shown in the figure, these environments include an apartment, a laboratory, and a bathroom and a toilet. The experimentation area of the apartment consists of two bedrooms, a corridor and the kitchen. The



Fig. 10. Experimental environments. [©David Rojas]

experiments were conducted in typical indoor environments to encompass cluttered, uncluttered, Line of Sight (LoS), non-LoS and through-wall scenarios. The link types of all environments are as follows: (i) kitchen–4 m, LoS cluttered environment, (ii) corridor–9 m, LoS uncluttered environment, (iii) lab–7 m, non-LoS cluttered, (iv) two bedrooms–5 m, through wall (single wall with decoupled and insulated plaster partition), and (v) bathroom and toilet–5 m, through wall (two walls, each having a plaster partition).

Hardware and Software Setup. We implemented FallDeFi using commercial WiFi devices, the transmitter and receiver in each link consist of two Linux laptops. Each of them is equipped with two external omnidirectional antennas. In each laptop, we installed a Wi-Fi network interface card with the IEEE 802.11n Intel WiFi Link 5300 chipset, modified the driver and firmware in the laptops to function as the transmitter and receiver. The Wi-Fi chipsets in the laptops operate in monitor mode which is one of the IEEE 802.11n Wi-Fi modes to sniff packets in a particular channel. We used the CSI Tool [39] to analyze the data collected from the chipsets. We make the source code and data sets available in [40].

Table 2. Data collection in two different days. Group A has five data sets for Corridor, Bedroom, Kitchen, Lab and Bathroom for falls and non-falls, while Group B has only four.

Group	Measurements	Corridor	Bedroom	Kitchen	Bathroom	Lab	Total
A	Falls	33	21	39	35	49	177
	Others	58	70	99	81	86	394
B	Falls	49	30	30	40	-	149
	Others	77	91	90	92	-	350
Changes from A to B		Diff. in Days	7	33	33	10	
		Diff. in environment	Tx. moved by .5m	+1 person	+1 person, furniture moved nLoS	+1 person, Tx moved by .5m	

Execution. During the data collection period for each activity, the transmitter sends 100 B packets in a $t_{int}=10$ s duration resulting in a data rate of 800 kb/s. Therefore, we collected 10 000 packets for each activity, corresponding to 30×10000 CSI values for each antenna pair. We perform all experiments in the 5.2 GHz band because it has less interference than 2.4 GHz band and the bandwidth of activities are higher than in 2.4 GHz. Table 2 illustrates the amount of repetitions of falls and other activities that were conducted in each environment. In each environment except the lab, we collected two data sets A and B on two different days with changes in the environment. We used the lab data set as the training environment for robustness studies in section 8.2.2. The table further illustrates the changes from data sets A to B. Collection of data sets A and B for bedrooms and kitchen followed a gap of 33 days while corridor and bathroom data sets had gaps of 7 and 10 days between them. Additionally, from A to B the corridor and bathroom transmitters were moved by 0.5 m from the original positions to synthetically generate environment changes as these two places are not cluttered by objects. In the kitchen, non line of sight furniture was moved. In the bedroom, kitchen and bathroom, during the collection of data set B, another person was present in the monitoring area besides the person who falls.

System Calibration. Our system depends on input parameters at four different stages: *denoising*, *stream selection after PCA*, *spectrogram processing* and at *threshold selection for event detection*. Denoising and PCA stream selection parameters are selected only once and these parameters are not changed across environments.

There are four parameters that must be initialized in wavelet de-noising: *the wavelet function* and its *coefficients*, the *number of wavelet levels*, and the *thresholding strategy*. As there are several strategies to select these four parameters, we studied them heuristically to select the best strategy for CSI signals. The amount of noise in CSI is the main characteristic that changes from one environment to another. Once the strategies are selected, DWT adjusts its thresholds to lower wavelet levels by estimating the noise in the highest wavelet. Therefore, this is a one time selection and initialization for all experiment environments. The criteria we set for tuning those parameters are: high correlation of the synthesized signal to the original signal after de-noising, preservation of high frequency components, and computational complexity. By comparing the synthesized signals obtained from different parameter combinations, the parameters that satisfied the selected criteria yielded a ‘symlet’ *wavelet function* with three *coefficients* and 10 *wavelet levels*. The *thresholding strategy* we used is: fixed-form threshold with threshold rescaling for individual wavelet levels and soft thresholding [41].

PCA stream selection requires only one parameter (captured variance) to be initialized and it is fixed for all environments. The main goal of captured variance initialization is to avoid selecting noisy PCs for the spectrogram. However, depending on the noise levels in each environment the number of selected PCs will vary dynamically.

Table 3. Performed activities in the experiments.

	Main categories	Sub variations
Fall Types	trip, slip lose consciousness lose balance	forward, backward, lateral, on-position
Other Activities	walk, jump, no event bend and pickup	
	random event	open, close door, raise hands, walk and pause
	sit down, stand up	floor, chair

The threshold \hat{N}_{th} used for spectrogram segmentation is estimated by considering several spectrograms corresponding to different activities for each environment. During the calibration phase we obtain the \hat{N}_{th} values for each spectrogram from each environment and compute a median segmentation threshold for that environment. During the testing phase, we use this threshold to segment all the spectrograms to isolate the main activity's time-frequency components. Since the power burst curve threshold (PBC_{th}) is dependent on this threshold, event detection step is also impacted by this value.

Activity Types. Table 3 specifies the types of activities that were considered in the experiments. Some of the activities in the main category have other variations, i.e., trip, slip, lose consciousness and lose balance can occur either as a forward, backward, lateral or as an on-position motion. These variations were mainly considered to avoid over-fitting to a particular falling motion. As each person may imitate the falls uniquely, to avoid over-fitting to a particular person, three volunteers aged between 27 to 30 years with different physiques performed the activities. In all experimentation environments, we evaluated two combinations of those volunteers: only one person is in the area of interest, two persons are in the area (when one person performs falling activities, the other is either involved in regular activities or is motionless). We do not use more than two actors in the experiments due to the application scenario, elderly falls go unnoticed when the inhabitants live independently. Hence, the system should be properly tuned to the worst case scenario where a single or couple of occupants are in the area of interest.

Evaluation Metrics. The evaluation metrics used in this paper are as follows:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \quad \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \quad \text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad \text{F1 score} = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \quad (7)$$

where TP = true positives, FP = false positives, TN = true negatives and FN = false negatives. Note that true positives are manually annotated falls that are correctly detected, false positives are non-falls that are detected as falls, true negatives are non-falls that are detected as non-falls and false negatives are falls that are detected as non-falls. Out of these metrics, recall and precision are the most important measures for our system. Recall provides the fraction of falls that were correctly identified among actual falls by our system. Precision identifies the fraction of relevant falls among all the detected falls, therefore it is a measure of false alarms. F1 score is the harmonic mean of these two metrics and accuracy identifies the fraction of correctly identified events out of all the events. For a good fall detection scheme all four metrics should be high, ideally close to 100 %.

8 PERFORMANCE EVALUATION

In this section we evaluate the performance of FallDeFi in different conditions, i.e., different rooms, link types, persons and activities as mentioned in § 7. First we evaluate the performance of individual signal processing

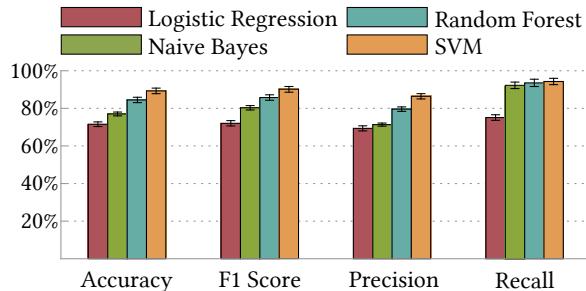


Fig. 11. Classifier comparison. Performance of SVM is higher than others in all the metrics.

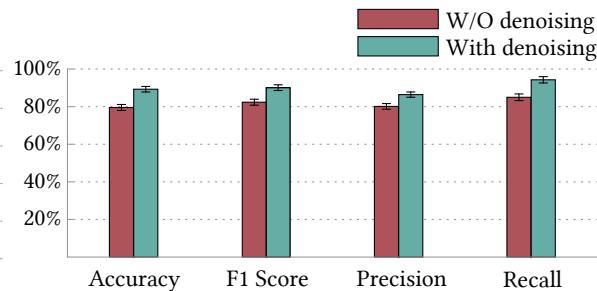


Fig. 12. Performance with and without de-noising as mentioned in § 4.1.

components of our system and identify their optimum operating points. Then we analyse the performance of the extracted features in classifying falls, i.e., OFs and SFs as defined in § 6. Unless otherwise mentioned, for the rest of the evaluations in this section we use only the SFs for classification, *event duration*, *spectral entropy* (1 to 10 Hz & 10 to 30 Hz) and *fractal dimension* as detailed in § 6.2. Finally we compare the performance of our system with two state of the art schemes, CARM [14] and RTFall [9].

We noticed a similar pattern of high variance in the classification accuracy in all experiments as a consequence of two reasons: (1) low sample size of the data sets and (2) class imbalance between falls and other fall like activities. As a solution, we generated synthetic samples from the data using the synthetic minority over sampling technique (SMOTE) [42].

8.1 Optimization and Validation of Different Components of FallDeFi

8.1.1 Classifier Selection. We performed binary classification to separate a fall from other activities because of two reasons. Firstly, we only require to distinguish a fall from another similar event. Secondly, our features are specially suitable for distinguishing falls from other events and not the individual activities. As performance of a classifier is mostly dependant on the data set, we compared several classifiers (logistic regression, naive Bayes, SVM and random forest) before choosing one. SVM was implemented using the LIBSVM [43] tool set, and the hyper-parameters σ and c were chosen through five-fold cross validation. For random forest, the number of trees was set to 25 as it provides high classification accuracy and avoids significant over-fitting. Then we compared the performance of each classifier as shown in Figure 11. Based on these results, SVM with RBF kernel provided the best performance out of the four classifiers and we used this classifier for the rest of the analysis.

8.1.2 Validation of Noise Filtering. We introduced a wavelet based de-noising technique in § 4.1. Recall that this approach enabled us the first principal component (together with other principal components) in constructing the spectrogram even for noisy links. In this section we compare the overall classification performance if the first principal component is used with and without noise filtering. Figure 12 compares the average results for all the noisy and noiseless data sets and illustrates that noise filtering improves the performance metrics by 5 % to 10 %.

8.1.3 Selection of the Optimum Number of Principal Components. As explained in § 4.2, the optimum number of principal components to be used in the spectrogram is heuristic and this number is determined by the captured variance metric. Here we compare the classification results with the captured variance of CSI which ranges from 85 % to 99 % corresponding to the required maximum number of principal components. We chose not to plot the 100 % value for captured variance as it does not reduce the original number of CSI streams and requires a significant amount of computational time as all the principal components have to be used in the spectrogram. As depicted in Figure 13 the classification performance (y axis in the figure) increases up to 95 % of

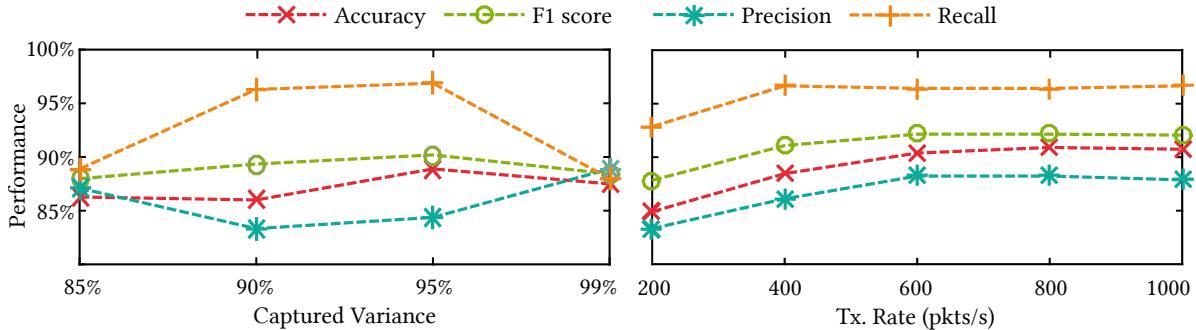


Fig. 13. Performance comparison of the cumulative variance of selected PCs as a percentage of cumulative variance of all the subcarriers.

Fig. 14. Performance comparison for different transmission rates.

captured variance (x axis) and starts to reduce from thereon. This shows that using a lower number of principal components corresponding to low captured variance may not provide enough information. Moreover, using a higher number includes noisy principal components which again reduces the performance. Therefore, in our scheme we selected 95 % of captured variance as the optimum value for principal component selection. This amount usually corresponds to 2 to 3 principal components on average.

8.1.4 Performance of FallDeFi under Different Transmission Rates. Here we evaluate the range of transmission rates for which FallDeFi performs well. We tested for rates ranging from 200 pkts/s to 1000 pkts/s with increments of 200 pkts/s. Different rates were achieved by down-sampling the transmission rate of 1000 pkt/s used in all the experiments so that parameters such as activities, positions and environments are fixed and the only variable is the sampling rate. The performance, as shown in Figure 14, is constant up to a transmission rate of 600 pkts/s and starts deteriorating from 400 pkts/s onward. This behavior is due to the fact that when the transmission rate is reduced, the bandwidth of the measured activity is also reduced. Thus, low frequency activities are perceived to be similar to high frequency activities resulting in lower classification accuracies. These results indicate that the performance of our system is consistent up to transmission rates of 600 pkts/s. Nonetheless, all the analysis in this paper was conducted using transmission rates of 1000 pkts/s.

8.2 Evaluations on the Original and Selected Features

As explained in § 6 we extracted a set of STFT-based features which we termed as Original Features (OF). Then, in § 6.2 we used sequential forward selection to select a feature set from the OFs that is robust to environment changes and we termed them as Selected Features (SFs). In this section we evaluate the performance of OFs and SFs across all the defined performance metrics and analyse the classification performance of our system in the two groups A and B listed in Table 2 in three different ways:

- (i) First we assess the classification performance in training and testing on the same data set ($A, A \& B, B$). For this we train 8 SVM models for Corridor, Bedroom, Kitchen and Bathroom in groups A and B and test on the same data set in each group by dividing them as 70% training and 30% testing.
- (ii) Then we analyse the classification performance of training on group A and testing on group B (denoted as A, B). We train four SVM models for the data (corridor, bedroom, kitchen, bathroom) in group A and test on the corresponding data in group B. Here we test whether the models are resilient to changes that occur over time in the same environment since the data in group B were collected several days after group A's data.

- (iii) Finally, we train an SVM model for the Lab data set in group A and test on each data set in groups A and B of the other four environments (denoted as *Lab, A+B*). Here we tested whether a model can be trained in a completely different environment and still apply this to other environments to eliminate the need for training each time we deploy a new system.

8.2.1 Performance of Original Features. Here we evaluate the performance of 8 models on the eight testing data sets consisting of OFs. The average results are illustrated in Figure 15a as *OF: A,A&B,B*. The figure shows that for OFs, all the performance metrics have values above 93 %, especially, with recall reaching as high as 97.5 %. This translates to a highly reliable fall detection system when the system is trained to the prevalent environment conditions.

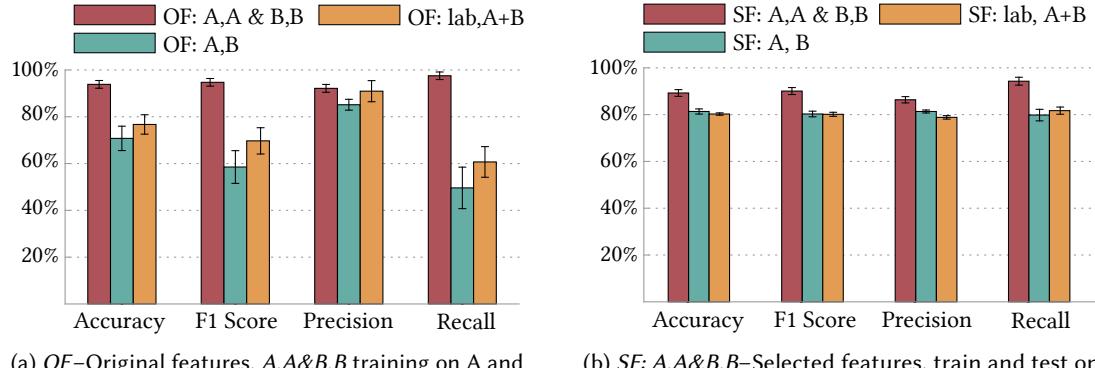
Then we tested the performance of OFs for A,B and Lab,A+B as mentioned above in § 8.2. The results of these two scenarios are indicated in Figure 15a as *OF: A,B* and *OF: Lab,A+B* respectively. As evident from the figure, all the performance indication metrics degrade considerably to a range between 55 % to 80 % for both the scenarios. This indicates that for OFs, the system performs poorly in detecting falls after environment changes occur or if the model is trained in a different environment.

8.2.2 Performance of Selected Features. To evaluate the performance of SFs, we follow the same procedure as mentioned above in §8.2.1. Figure 15b shows the average classification results for SFs for the three scenarios, *SF: A,A&B,B*, *SF: A,B*, and (*SF: Lab,A+B*). Average results indicate that *SF: A,A&B,B* performs only 3 % worse than *OF: A,A&B,B*, yet, all metrics exceed 90 %. This is a clear indication that when trained to the prevalent conditions, the difference in performance of SFs and OFs is small. However, for the latter two scenarios, *SF: A,B* and *SF: Lab,A+B*, we notice an improvement in accuracy, f1 score and recall for SFs. This indicates that the SFs have especially improved the ability to detect falls of our system albeit a decrease (12 %) in precision for *SF: A,B* from *OF: A,B*. Yet, all the average performance metrics are closer to 80 % for SFs. More importantly, this illustrates that the SFs have not only good performance when trained to the existing environment, they are resilient in fall detection when environment changes occur or when the training is done in a completely different environment.

The reason for this behavior can be attributed to the type of features that were selected, i.e., event duration, spectral entropy and fractal dimension. Typically, an event duration of 2 s to 3 s is a unique characteristic of falls irrespective of the environment. Spectral entropy and fractal dimension are normalized features, the former quantifies the variations in the amplitudes in a specific frequency range while the latter quantifies the roughness of the extreme frequency curve.

8.2.3 Robustness of Selected Features on each Environment. Average performance results indicate the general tendencies of our system, however, they do not yield insights into the impact on each environment. Here we provide this view using only the selected features because they have the best performance in all the conditions. In Figures 16a, 16b, 16c and 16d we compare the accuracy, f1 score, precision and recall of corridor, bedroom, kitchen, lab and bathroom. First we show the results for training and testing on the same data sets of A (*SF: A,A*) and B (*SF: B,B*), then compare the results of training on data set A and testing on the data set B (*SF: A,B*). Finally we compare the results for training on Lab data and testing on both A and B data sets (*SF: Lab,A+B*).

The results for individual environments confirm the average results in § 8.2.2. Even though training and testing on the same data set performs slightly better than the other two scenarios as expected, all the metrics perform above 70 %, especially, the accuracy, f1 score, and precision are either above or closer to 80 %. For recall, the lowest performance is $\approx 70\%$ in corridor and bathroom data sets for *SF: A,B*. Generally, for corridor bedroom and bathroom, other metrics also perform slightly lower than for kitchen. The reason is that for the former three environments, the link is not as strong as in the kitchen (corridor: 9m LoS, bedroom: 5m through wall and bathroom: 5m through wall vs kitchen: 4m LoS). The link in the bathroom penetrates two walls while the link in the bedroom penetrates one wall. We notice that for falling positions that are far from the receiver or the link,



(a) *OF*-Original features, *A,A&B,B* training on *A* and *B* and testing on *A* and *B*. *A,B*-training on *A* and testing on *B*. *Lab,A+B*-Training on *Lab* data set and testing on *A* and *B*.

(b) *SF*: *A,A&B,B*-Selected features, train and test on same data set.*SF*: *A,B*-Selected features, train on *A* and test on *B*. *SF*: *Lab,A+B*-Selected features, train on *Lab* data set and test on *A* and *B*.

Fig. 15. Accuracies of original (a) and selected (b) features.

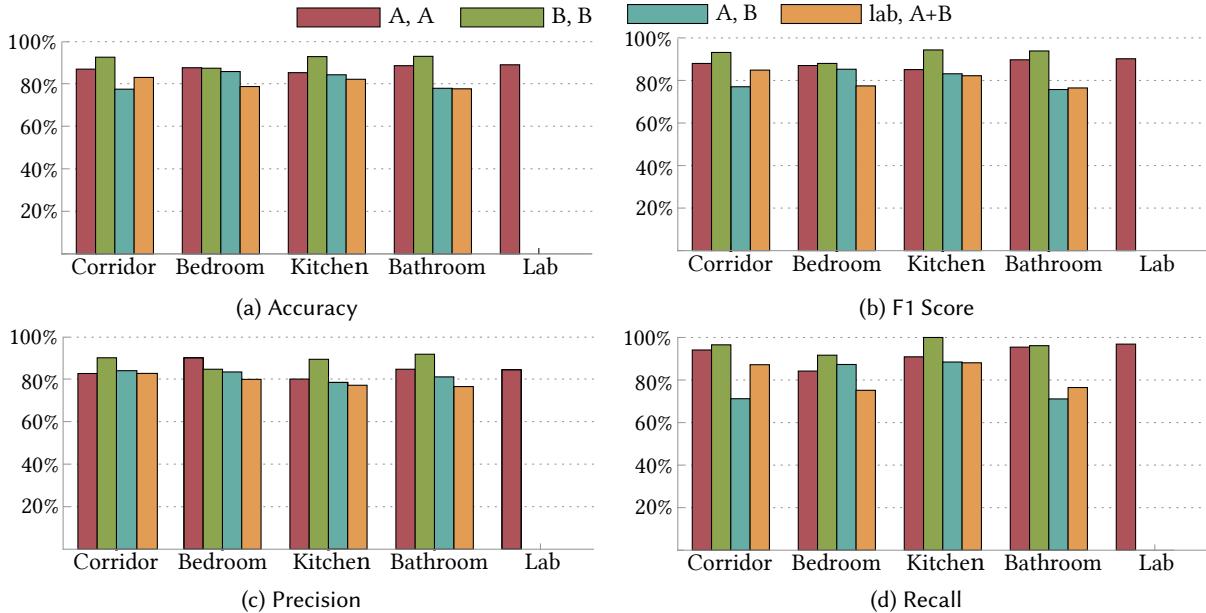


Fig. 16. Comparison of FallDeFi performance across different environments. For these comparisons we only used SFs. The notation used in all the figures are as follows. *A,A*-Training on *A* and testing on *A*. *B,B*-Training on *B* and testing on *B*. *A,B*-Training on *A* and testing on *B*. *Lab,A+B*-Training on *Lab* data set and testing on *A* and *B*.

especially when the link is weak as in above three environments, high frequencies/energies are not effectively captured from an activity. Additionally, when a fall or a similar activity e.g. sit down is closer to the link or the receiver, we notice the amplitudes in high frequencies increasing, causing an adverse effect on distinguishing falls from other activities.

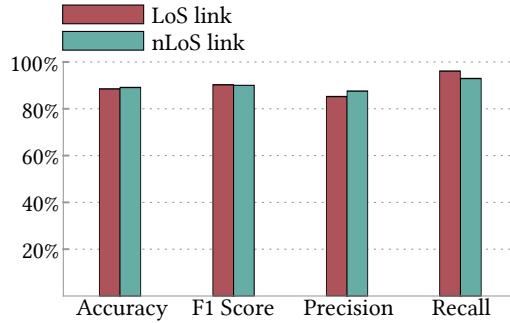


Fig. 17. Performance comparison for falls and other activities in LoS and nLoS links.

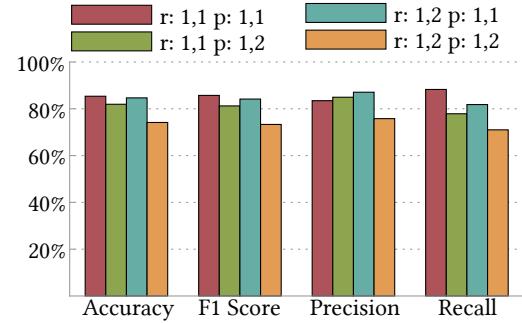


Fig. 18. Performance comparison for training and testing in the same room for the same person (r:1,1 p:1,1), two different rooms but same person (r:1,2 p:1,1), same room but different persons (r:1,1 p:1,2), and two different rooms each having a different person (r:1,2 p:1,2).

8.3 Impact of the Link type and the Person who does the Activities

We then divided the results of all the environments according to the link type, LoS and nLoS links. The results are shown in Figure 17. According to the results, there's no clear difference between the system's performance for LoS and nLoS links and these results are consistent with the average results of SF:*A,A&B,B* in Figure 15b.

Recall that we used three persons (p1, p2 & p3) to train our system. Here we analyze how much of an impact each person's body type and unique falling characteristics would have on the overall performance of the system. First we trained and tested FallDeFi for falls and other activities in a single room for the same person (r:1,1 p:1,1). Then we analyze for the same person trained and tested in two different rooms (r:1,2 p:1,1), trained and tested on two different persons in the same room (r:1,1 p:1,2), and trained and tested for two different persons each of them placed in two different rooms (r:1,2 p:1,2). The results are illustrated in figure 18 and as expected they indicate that training and testing for the same person yields higher performance than training on one person and testing on another person. The lowest performance is yielded for r:1,2 p1:2 where both the person and the environment change from the training phase to the testing phase (the accuracy is $\approx 10\%$ lower than r:1,1 p:1,1).

8.4 Comparison of FallDeFi with CARM [14] and RTFall [9]

We compared the performance of our approach using SFs with two other state of the art CSI-based schemes CARM [14], an activity recognition scheme and RTFall [9], a fall detection system. We modelled the feature space of CARM using a Gaussian mixture model and the state space using a Hidden Markov model (HMM). For fair comparison with FallDeFi, we used our event detection method to detect events. We trained two HMMs for falls and non-falls for binary classification. The number of Gaussian components in a mixture and the number of states for falls and non-falls were determined through cross validation. We initialized the mean vector and covariance matrix of the Gaussian mixture, state probabilities and transition probabilities of the HMMs as mentioned in [14].

We implemented RT-Fall exactly as described in the literature using CSI phase difference from two antennas to extract features and compared with our results. Even though there are two other CSI-based fall detection schemes in the literature, we selected RTFall specifically because it builds upon the features used by those approaches. The CSI packet transmission rate is set to 100 pkts/s which is achieved by down-sampling the 1000 pkt/s rate. The band-pass filter had a range of 5 Hz to 10 Hz. We implemented RTFall's event detection method as the feature,

Table 4. Average classification comparison of FallDeFi, CARM and RTFall for Corridor, Bedroom, Kitchen, Bathroom and Lab. Used notation: $A,A\&B,B$ -Training and testing on same data of A and B; A,B -Training on A and testing on B; $Lab,A+B$ -Training on Lab data and testing on A and B. The results of the two best schemes for each data set are highlighted in bold.

% Accuracy	A,A & B,B			A,B			Lab, A+B		
	FallDeFi	CARM	RTFall	FallDeFi	CARM	RTFall	FallDeFi	CARM	RTFall
Accuracy	88.90	77.85	80.96	80.90	75.68	59.35	80.10	64.26	51.14
F1 Score	90.24	76.41	82.45	82.34	77.15	57.72	80.15	58.34	57.00
Precision	86.65	76.73	80.01	76.12	73.60	56.63	78.95	67.90	51.40
Recall/TPR	94.33	80.32	85.20	90.48	81.70	62.81	81.56	53.07	67.82
FPR	14.92	24.63	24.00	27.87	30.34	45.52	21.56	24.55	65.91

time lag relies on this particular event detection. We used SVM for binary classification and the two thresholds for event detection were determined from cross validation.

In Table 4, we first compare the average performance of FallDeFi, CARM and RTFall when the systems are trained for the same data set and tested for the same ($A,A\&B,B$). Then we train a separate model for each environment in data set A and test each model for the corresponding environment in data set B (A,B). Then a model for the Lab data set is trained and tested for both the data sets A and B in corridor, bedroom, kitchen and bathroom ($Lab,A+B$). We measure the performances using accuracy, F1 score, precision, recall/true positive rate (TPR) and false positive rate (FPR). Recall/TPR characterizes the number of false alarms generated in relation to missed fall events. $FPR = \frac{FP_s}{FP_s + TN_s}$ characterizes the false alarms at zero-fault level. These metrics are average values for a total of 326 falls and 744 non-falls. In the same table we compare these values with the state of the art. It is evident from the results that FallDeFi has a better accuracy (11% >CARM and 8% >RTFall), a better true positive rate (14% >CARM and 9% >RTFall) and a better false positive rate (10% <CARM and 15% <RTFall) for training and testing in same environment. FallDeFi is also more robust across different environments than CARM and RTFall as indicated by the A,B and $Lab,A+B$ results. The reasons for this performance enhancement can be attributed to the denoising technique we introduced and the resultant activation of the 1st PC (1st PC contains the majority of information in the subcarriers) and the rigorous feature selection procedure that was followed to extract environment resilient features. However, this performance comes at the price of computational time. The data in Table 5 illustrates this. FallDeFi takes twice the computational time (0.6s) than RTFall (0.34s), especially due to feature extraction from the spectrogram. These computation times are for 1 s of measured data. Therefore, the important fact is that even with the added computation efforts, FallDeFi is still able to produce results before the next second of data measurements, offering real-time execution.

Table 5. Comparison of FallDeFi, CARM and RTFall processing times for one second of CSI.

	CPU time					
	FallDeFi		CARM		RTFall	
	(s)	(%)	(s)	(%)	(s)	(%)
Amplitude/phase extraction	0.282	45.30%	0.282	46.70%	0.290	85.00%
Denoise	0.018	3.00%	0	0.00%	0.001	0.29%
PCA	0.002	0.30%	0.002	0.33%	0	0.00%
Spectrogram	0.020	3.20%	0.020	3.30%	0	0.00%
Feature Extraction & classification	0.300	48.20%	0.300	49.67%	0.050	14.66%
Total	0.622		0.604		0.341	

9 LIMITATIONS & FUTURE WORK

Although FallDeFi improves on the current state of the art, our system nevertheless is a research prototype implementation and requires further improvements before it can finally be deployed for the intended application of saving lives of elderly people. More specifically, we need to assess the optimum density of WiFi access points for a successful deployment in domestic environments. Currently, we have chosen 5 access points for a 60 m^2 area (Fig. 10a) and placed them in positions that were perceived to achieve maximum coverage. Furthermore, we achieve an overall TPR of 80 % and an FPR of 20 % across all scenarios which translates to 2 in 10 falls are undiscovered and 2 in 10 non-falls are detected as falls. Additionally, in situations where falls occur at locations with low WiFi coverage or in weak links, the TPRs further decrease (Fig. 16d). As the majority of events that occur during a day in real life are non-falls, FPRs in excess of 20 % are still not adequate which may result in annoying the users involved in daily activities. Therefore, we plan to mitigate these two shortcomings by the fusion of data from other ubiquitous fall detection technologies.

FallDeFi can be further enhanced by the fusion of location information of the targets. As mentioned in § 8.2.3, the target position has an impact on distinguishing falls that occur far away from the link. This is also confirmed by radar based fall detection research [23], where occupant range information has increased the detection of falls. This could be obtained using the same FallDeFi infrastructure. We can further distinguish falls from other activities by capturing the movement direction obtained through Doppler signatures. For this, we require precise phase information of CSI with further sanitization of the signal. Currently, our system uses CSI measurements from one transceiver pair dedicated to monitor falls in a particular area.

As mentioned in § 7, we used a data rate of 1000 pkt/s, which is equivalent to 800 kbps. As the results in § 8.1.4 show, this can be further reduced to 480 kbps without a degradation in accuracy. We believe that this amount of data rate is easily achievable in existing WiFi networks, especially 802.11n devices without degrading the network performance, when mixed with other services. However, we envisage that this fall detection system can also be installed as an independent system from existing home wireless networks as the cost of WiFi access points and receiver cards is quite low compared to other non-wearable fall detection systems such as Doppler radars.

Although CSI is available in all 802.11n compatible chipsets, it can be accessed through software only in a limited number of chipset ranges with modified drivers and firmware e.g. Intel 5300 and Atheros 9k series. However, this issue is not just limited to our system. With the current expansion of research using CSI as the signal strength descriptor, we expect chipset manufacturers to expose CSI externally in a wider range of products.

As it is not possible to subject elderly people to simulate the falls, our data were collected from young volunteers (aged between 27-30 years). This implies that the system parameters require tuning for elderly that fall in real world situations. This issue is also common to all fall detection systems. Nonetheless, we require further analysis on adaptation mechanisms of the system to such scenarios.

10 CONCLUSIONS

In this paper we presented a fall detection system comprising of commodity WiFi devices. We provided a novel methodology for time-frequency analysis of CSI for WiFi fall detection including improved CSI preprocessing techniques. Using these techniques we were able to obtain high performance under transmission rates as low as 600 pkts/s. We also developed a pre-screener with 100% detection rate for fall like events using the spectrogram and used it for feature extraction. Through feature effective extraction in time-frequency domains, we detect and classify falls with low dependence on the training environment. We achieve above 93 % average accuracy for a pre-trained system while the average accuracy is closer to 80 % with changes in the environment or when the system is trained in a different environment.

11 ACKNOWLEDGMENTS

This work has been funded by the Irish Research Council and United Technologies Research Centre, Cork, Ireland. We thank Ramona Marfievici of the Nimbus Centre, Cork Institute of Technology, Ireland and Pablo Corbalán of University of Trento, Italy for providing insightful comments.

REFERENCES

- [1] United Nations, Department of Economic and Social Affairs, Population Division (2015). World Population Prospects: The 2015 Revision, custom data acquired via website. <https://esa.un.org/unpd/wpp/>. Accessed: 2017-04-24.
- [2] B. Erol, M. G. Amin, B. Boashash, F. Ahmad, and Y. D. Zhang. Wideband radar based fall motion detection for a generic elderly. In *2016 50th Asilomar Conference on Signals, Systems and Computers*, pages 1768–1772, 2016.
- [3] Moeness G Amin, Yimin D Zhang, Fauzia Ahmad, and KC Dominic Ho. Radar signal processing for elderly fall detection: The future for in-home monitoring. *IEEE Signal Processing Magazine*, 33(2):71–80, 2016.
- [4] Bo Yu Su, KC Ho, Marilyn J Rantz, and Marjorie Skubic. Doppler radar fall activity detection using the wavelet transform. *IEEE Transactions on Biomedical Engineering*, 62(3):865–875, 2015.
- [5] E. Cippitelli et al. Radar and RGB-Depth Sensors for Fall Detection: A Review. *IEEE Sensors Journal*, 17(12):3585 – 3604, 2017.
- [6] L Day. Falls in Older People: Risk Factors and Strategies for Prevention. *Injury Prevention*, 9(1):93–94, 2003.
- [7] Leila Takayama, Caroline Pantofaru, David Robson, Bianca Soto, and Michael Barry. Making technology homey: finding sources of satisfaction and meaning in home automation. In *Proc. of the 2012 ACM Conf. on Ubiquitous Computing*, pages 511–520. ACM, 2012.
- [8] Ossi Kaltiokallio, Maurizio Bocca, and Neal Patwari. Enhancing the accuracy of radio tomographic imaging using channel diversity. In *Mobile Adhoc and Sensor Systems (MASS), 2012 IEEE 9th International Conference on*, pages 254–262. IEEE, 2012.
- [9] H. Wang, 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*, 16(2):511–526, 2017.
- [10] Muhammad Mubashir et al. A survey on fall detection: Principles and approaches. *Neurocomputing*, 100:144–152, 2013.
- [11] Raul Igual et al. Challenges, issues and trends in fall detection systems. *Biomedical engineering online*, 12(1):66, 2013.
- [12] Natthapon Pannurat et al. Automatic fall monitoring: a review. *Sensors*, 14(7):12900–12936, 2014.
- [13] Xuefeng Liu, Jiannong Cao, Shaojie Tang, Jiaqi Wen, and Peng Guo. Contactless Respiration Monitoring Via Off-the-Shelf WiFi Devices. *IEEE Transactions on Mobile Computing*, 15(10):2466–2479, 2016.
- [14] Wei Wang, Alex X. Liu, Muhammad Shahzad, Kang Ling, and Sanglu Lu. Understanding and Modeling of WiFi Signal Based Human Activity Recognition. In *Proc. of the 21st Annual Int. Conf. on Mobile Computing and Networking, MobiCom ’15*.
- [15] Brad Mager, Neal Patwari, and Maurizio Bocca. Fall detection using RF sensor networks. In *24th Int. Symp. on Personal Indoor and Mobile Radio Communications (PIMRC)*, pages 3472–3476. IEEE, 2013.
- [16] Heba Abdelnasser, Moustafa Youssef, and Khaled A Harras. Wigest: A ubiquitous wifi-based gesture recognition system. In *IEEE Conference on Computer Communications (INFOCOM)*, pages 1472–1480. IEEE, 2015.
- [17] Zicheng Chi, Y. Yao, Tiantian Xie, Zhichuan Huang, M. Hammond, and Ting Zhu. Harmony: Exploiting coarse-grained received signal strength from IoT devices for human activity recognition. In *IEEE 24th Int. Conf. on Network Protocols (ICNP)*, pages 1–10, 2016.
- [18] Anh Luong, Alemayehu Solomon Abrar, Thomas Schmid, and Neal Patwari. RSS step size: 1 dB is not enough! In *Proc. of the 3rd Workshop on Hot Topics in Wireless*, pages 17–21. ACM, 2016.
- [19] Youngwook Kim and Hao Ling. Human activity classification based on micro-doppler signatures using a support vector machine. *IEEE Transactions on Geoscience and Remote Sensing*, 47(5):1328–1337, 2009.
- [20] Ph Van Dorp and FCA Groen. Feature-based human motion parameter estimation with radar. *IET Radar, Sonar & Navigation*, 2(2):135–145, 2008.
- [21] Ajay Gadde, Moeness G Amin, Yimin D Zhang, and Fauzia Ahmad. Fall detection and classifications based on time-scale radar signal characteristics. In *SPIE Defense+ Security*, pages 907712–907712. International Society for Optics and Photonics, 2014.
- [22] Luis Ramirez Rivera et al. Radar-based fall detection exploiting time-frequency features. In *Signal and Information Processing (ChinaSIP), 2014 IEEE China Summit & International Conference on*, pages 713–717. IEEE, 2014.
- [23] Baris Erol and Moeness G Amin. Fall motion detection using combined range and doppler features. In *Signal Processing Conference (EUSIPCO), 2016 24th European*, pages 2075–2080. IEEE, 2016.
- [24] Qisong Wu, Yimin D Zhang, Wenbing Tao, and Moeness G Amin. Radar-based fall detection based on Doppler time–frequency signatures for assisted living. *IET Radar, Sonar & Navigation*, 9(2):164–172, 2015.
- [25] J. Hong, S. Tomii, and T. Ohtsuki. Cooperative fall detection using Doppler radar and array sensor. In *IEEE 24th Annual Int. Symp. on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pages 3492–3496, 2013.
- [26] Yan Wang et al. E-eyes: device-free location-oriented activity identification using fine-grained wifi signatures. In *Proc. of the 20th annual int. conf. on Mobile computing and networking (Mobicom)*, pages 617–628. ACM, 2014.

- [27] Wei Wang, Alex X. Liu, and Muhammad Shahzad. Gait Recognition Using WiFi Signals. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '16, 2016.
- [28] Kun Qian, Chenshu Wu, Zimu Zhou, Yue Zheng, Zheng Yang, and Yunhao Liu. Inferring Motion Direction using Commodity Wi-Fi for Interactive Exergames. In *Proc. of the 2017 CHI Conference on Human Factors in Computing Systems*, May 6-11 2017.
- [29] C. Han et al. Wifall: Device-free fall detection by wireless networks. In *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, 2014.
- [30] Yuxi Wang et al. Wifall: Device-free fall detection by wireless networks. *IEEE Transactions on Mobile Computing*, 16(2):581–594, 2017.
- [31] Daniel Halperin, Wenjun Hu, Anmol Sheth, and David Wetherall. Predictable 802.11 Packet Delivery from Wireless Channel Measurements. *SIGCOMM Comput. Commun. Rev.*, 40(4):159–170, August 2010.
- [32] Sameera Palipana, Piyush Agrawal, and Dirk Pesch. Channel State Information Based Human Presence Detection Using Non-linear Techniques. In *Proc. of the 3rd ACM Int. Conf. on Systems for Energy-Efficient Built Environments*, BuildSys '16, 2016.
- [33] Jonathon Shlens. A tutorial on principal component analysis. *CoRR*, abs/1404.1100, 2014.
- [34] Baris Erol, Moeness Amin, Fauzia Ahmad, and Boualem Boashash. Radar fall detectors: A comparison. In *SPIE Defense+ Security*, pages 982918–982918. International Society for Optics and Photonics, 2016.
- [35] W. Tao et al. Color Image Segmentation Based on Mean Shift and Normalized Cuts. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 37(5):1382–1389, 2007.
- [36] Wenbing Tao, Hai Jin, Yimin Zhang, Liman Liu, and Desheng Wang. Image thresholding using graph cuts. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 38(5):1181–1195, 2008.
- [37] Philipp Heidenreich, Luke A Cirillo, and Abdelhak M Zoubir. Morphological image processing for FM source detection and localization. *Signal Processing*, 89(6):1070–1080, 2009.
- [38] Aihua Zhang, Bin Yang, and Ling Huang. Feature Extraction of EEG Signals Using Power Spectral Entropy. In *Proc. of the 2008 Int. Conf. on BioMedical Engineering and Informatics - Volume 02*, BMEI '08, pages 435–439, 2008.
- [39] Daniel Halperin et al. Tool release: gathering 802.11n traces with channel state information. *ACM SIGCOMM Comput. Commun. Rev.*, 41(1):53–53, 2011.
- [40] Sameera Palipana. Falldefi source code and data. <https://github.com/dmusp123/FallDeFi>, 2017.
- [41] David L Donoho. De-noising by soft-thresholding. *IEEE transactions on information theory*, 41(3):613–627, 1995.
- [42] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.
- [43] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.

Received May 2017; revised August 2017; accepted October 2017