A WiFi-based Home Security System

Shaohu Zhang, Raghav H. Venkatnarayan, Muhammad Shahzad
North Carolina State University, USA. {szhang42, rhampap, mshahza}@ncsu.edu

***Abstract*—Typical home security systems monitor homes for intrusions by installing contact sensors on doors and windows and motion sensors inside the house. Unfortunately, due to the high deployment and operational costs of today's home security systems, only a small fraction of homes have security systems installed (*e.g.*, only 17% in the US and 15% in China). In this paper, we propose a WiFi based Home Security system (WiHS) that uses commodity WiFi devices, which most modern households already have, to perform the three primary tasks of typical home security systems: 1) detect when a door/window is opened/closed, 2) identify which door/window has been opened/closed, and 3) detect movements inside the house. The design of WiHS is based on our intuitive and theoretical understanding of the impacts of the movements of doors and windows on WiFi signals, which we will develop and present in this paper. We extensively evaluated WiHS using commodity WiFi devices in 3 different houses. WiHS detected intrusions with over 95% accuracy and identified the exact door/window that moved with just 4.5% average error.**

I. Introduction

Home security systems provide good deterrence against intrusions and burglaries [1], [2]. A typical home security system installs contact sensors on doors and windows and motion sensors inside the house, and raises an alarm if a door/window is opened or any motion is detected. Unfortunately, the deployment cost of today's home security systems is high. For example, in the US, the cost of a typical door, window, and motion sensor ranges from \$30 to \$80 [3], [4]. A home with front and back entrance doors, 10 windows, and 2 motion sensors incur about \$500 for just the senors; security monitoring hub and the labor cost are in addition. Due to such high costs, only a small percentage of homes have security systems [5] (*e.g.*, $<$ 17% in US [6] and $<$ 15% in China [7]).

While a large percentage of homes do not have security systems, most modern homes have WiFi. If one could replicate the monitoring functions of a conventional security system using existing WiFi devices, the resulting WiFi based security system will come at negligible/no monetary cost. Researchers have already shown that different human movements impact WiFi signals differently, and have leveraged this observation to develop human sensing systems [8]–[12]. We further observed that the movements of doors and windows situated at different locations in a house also impact WiFi signals differently. Thus, by measuring changes in WiFi signals, we should be able to detect whether and which door/window is being opened/closed and whether there is a movement inside the house.

Problem Statement: Our objective is to develop a WiFi based home security system that can perform the three primary monitoring tasks of the conventional security systems: 1) detect when a door/window is opened/closed, 2) identify which door/window has been opened/closed, and 3) detect whenever there is a movement inside the house.

Proposed Solution: In this paper, we propose WiHS, a WiFi based Home Security system that can accurately perform the three monitoring tasks using commodity WiFi devices. We have designed WiHS for two typical scenarios: 1) *away-mode*, where all occupants have gone outside (such as to their offices/schools), and 2) *stay-mode*, where one or more occupants are inside the house (such as when going to bed at night). When arming WiHS, an occupant selects which scenario he/she is arming it for.

Fig. 1 shows a block diagram that illustrates how WiHS uses the channel state information (CSI) reported by commodity WiFi devices to perform the three monitoring tasks. The *denoising* block continuously captures the CSI measurements from WiFi network interface cards (NIC), removes noise from them, and outputs a time series of denoised CSI values, which we will refer to as *denoised-stream*. As noise removal from CSI is a well studied topic, we simply adopted the principal component analysis based noise removal technique proposed in [13], [14], and will thus not discuss it further. The *segmentation* block takes the denoised-stream as input and continuously analyzes it to detect any movement. As soon as it detects a movement, it first determines whether the movement happened inside the house or outside, and if it is inside, it then determines whether the detected movement was due to a moving door/window or human. If the movement was due to a door/window, it raises an alarm and outputs the segmented portion of the denoised-stream containing the movement. If the movement was due to a human, it raises an alarm only if the system is armed for away-mode. The *D/W–identification* block takes the segmented denoised-stream as input and evaluates it against the classification model of each door and window in the house to determine which door/window moved. Finally, the *O/C–identification* block takes the same segmented denoised-stream along with the decision of the D/W–identification block as inputs, evaluates it against the classification model of open and close events of the identified door/window to determine whether the door/window opened or closed.

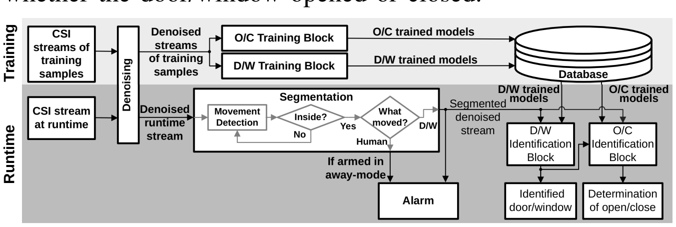


Fig. 1. Block Diagram of WiHS

To generate classification models that the D/W– and O/C–identification blocks use, for each door/window, WiHS requires an occupant to provide about 30 training samples each of opening and of closing that door/window. WiHS passes these samples through the denoising block and provides the denoised-streams as input to the *D/W–training* and the *O/C–training* blocks. From these denoised-streams, the D/W–training block extracts features appropriate for distinguishing between doors and windows, trains classifiers using the extracted features, and stores the trained classification models in a database from where the D/W–identification block retrieves them at runtime. The O/C–training block trains classifiers in a similar way for the O/C–identification to use at runtime.

Positioning the Proposed Solution: Compared to conventional security systems, which provide deterministic outputs, WiHS is probabilistic and can miss or misclassify some events. We emphasize that WiHS is not intended to replace conventional security systems. It is intended to provide a no/low-cost solution using existing WiFi equipment for homes that otherwise do not have a security system at all.

Key Contributions: In this paper, we make five key contributions. 1) We present WiHS, a WiFi based home security system that performs the three monitoring tasks of conventional security systems using commodity WiFi devices. 2) We characterize the impact of the movements of doors and windows on WiFi signals and explain our observations both intuitively and theoretically. 3) We present a method to distinguish between the movements of humans and of doors/windows inside any house. 4) We present our implementation and extensive evaluations of WiHS in three different houses. 5) We will open source the large data set of WiFi traces that we have collected and used to evaluate WiHS.

II. Movement Segmentation

Let N_{Tx} and N_{Rx} represent the number of Tx and Rx antennas, respectively, on any given WiFi device. Let S represent the number of OFDM subcarriers between each Tx-Rx pair. Each CSI measurement comprises $S \times N_{Tx} \times N_{Rx}$ channel frequency response (CFR) values, one for each subcarrier between each Tx-Rx pair. As WiFi NICs generate CSI measurements repeatedly, we obtain $S \times N_{Tx} \times N_{Rx}$ time-series of CFR values. Onward, we will call each time-series of CFR values a *CSI-stream*.

1) Segmentation: To detect the start and end of a movement, on any given denoised-stream, the segmentation block (shown in Fig. 1) slides a window of size W with a step size of S and calculates the variance of the values covered by the window in each step. We call the resulting stream of values a *variance-stream*. Fig. 2(a) shows a denoised-stream resulting from a door open followed by a close followed by another open and close. Fig. 2(b) shows the corresponding variance-stream. We observe from this figure that the values in the variance-stream in the presence of a door movement are much higher compared to in the absence. We made similar observations for window and human movements.

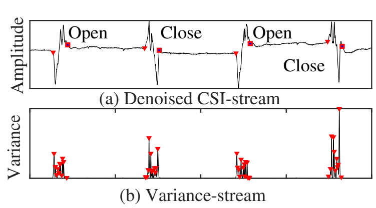


Fig. 2. Denoised and vairance-streams from door open and close events

To detect the start and end of a movement, the segmentation block first identifies all the local maxima in the variance-stream by comparing each value with a value before and after it. Next, it determines whether each local maximum represents a prominent peak or if it just appeared due to noise. For this, it compares the value of each maximum with a threshold T (we will describe shortly how WiHS sets the value of T). This simple approach works because the variance in the presence of a movement is much larger compared to in the absence. The segmentation block discards all the local maxima whose values are less than T. We refer to the remaining local maxima as peaks. The markers in Fig. 2(b) show the peaks detected by the segmentation block using this approach. Next, the segmentation block randomly chooses a peak and identifies all those peaks in its vicinity that are separated in time by less than a second from their respective adjacent peaks, and *obtains* a set of closely spaced peaks. We empirically observed that in the presence of a human movement or a door/window movement, the adjacent peaks are never separated by more than a second. The segmentation block uses the first point in the denoised-stream covered by the sliding window that generated the first peak in the *obtained* set of peaks as the start point of the movement. Similarly, it uses that point in the denoised-stream as the end point of the movement that comes W points after the last point in the denoised-stream covered by the sliding window that generated the last peak in the *obtained* set of peaks. The markers in Fig. 2(a) show the start and end points detected using this method.

To set the value of T, at the time of initial setup, WiHS requires the homeowner to perform a one-time calibration operation. In this operation, it requires the homeowner to first keep the environment inside the house static for 20 seconds by ensuring that no humans or doors/windows inside the house move, and then randomly walk around in the house for another 20 seconds. After this, WiHS calculates mean μ_S of the variance-stream corresponding to the 20 seconds of no movement and mean μ_W of the variance-stream corresponding to the 20 seconds of walk and sets $T = (1-\alpha) \times \mu_S + \alpha \times \mu_W$. Next, we determine the appropriate value of the weighting factor α.

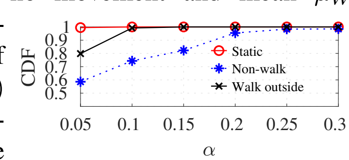
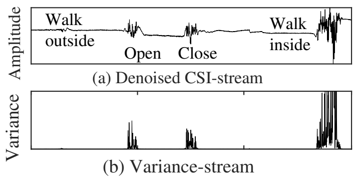


Fig. 3. %age of values $\leq T$ vs. α

Fig. 3 plots the percentage of values that are below T in the variance-streams in all our traces when the environment was static, where we varied α from 0.05 to 0.3. We observe from this figure that when $\alpha \geq 0.1$, 100% of the values in the variance streams are $< T$ when the environment was static. Fig. 3 also plots the percentage of values that are below T when one or more persons walked outside (but close to the outer walls of) the house. We observe that when $\alpha \geq 0.15$, 100% of the values in the variance streams are $< T$ when the people moved outside the house. Fig. 3 further plots the percentage of values that are below T when one or more persons performed non-walking movements (such as sitting down, eating, *etc.*) inside the house. We observe that when

$\alpha \geq 0.25$, 98.34% of the values in the variance streams are $< T$ when people performed non-walking movements inside the house. We further observed from our traces that when $\alpha = 0.25$, over 85.65% of the local maxima in the variance-stream were greater than T when a door/window moved or a human walked inside the house. Consequently, we set $\alpha = 0.25$. To visually demonstrate these observations, consider Figs. 4(a) and 4(b), where we plot the denoised-stream and corresponding variance stream, respectively, when a user performed four activities: 1) briefly walked outside the house, 2) opened the door, 3) entered and closed the door, and 4) walked inside the house. We can clearly see from this figure that walking outside the house results in significantly smaller (almost invisible) peaks in the variance-stream compared to walking inside the house or opening/closing a door. Thus, using $\alpha = 0.25$, the segmentation block is able to ignore movements that happen outside the house and any non-walking movements that happen inside, and detects the start and end of only those movements that are comprised of human walk or the door/window movement.



(a) Denoised CSI-stream
(b) Variance-stream

Fig. 4. Denoised and variance-streams resulting from various activities

2) Distinguishing Door/Window vs. Walking: The method described above already enables the segmentation block to detect start/end times of movements that are due to door/window or walk, and enables it to discard any non-door/window/walk movements inside the house or any movements outside. Next, we describe how it determines whether a segmented movement is due to a person walking (or significantly changing position) or due to a door/window.

WifiU [15] and WiStep [16] demonstrated that human movements introduce frequencies up to 20 Hz in CSI at 2.4 GHz. We will show (intuitively, theoretically, and experimentally) in Secs. III-A and III-B that the movements of doors and windows introduce frequencies in the range of 3 to 11 Hz in the denoised-streams. These observations are visually demonstrated in Figs. 5 and 6 as well, where we can see that the door/window movements do not introduce any significant frequencies beyond 11 Hz in the denoised-stream, whereas, human walk introduces frequencies beyond 11 Hz as well. The segmentation block leverages these observations to distinguish between door/window movements and walking movement.

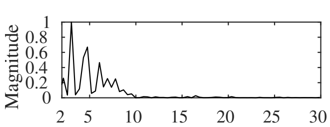


Fig. 5. Freqs. introduced by door

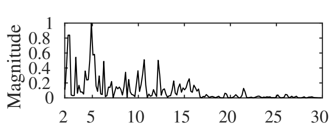


Fig. 6. Freqs. introduced by walk

Given any segmented denoised-stream, the segmentation block first applies fourier transform on it. Next, it performs max-min normalization on the magnitudes of the frequencies in the range of 3 to 20 Hz to bring the magnitude of each frequency in the range of 0 to 1. After this, it calculates the sum of the magnitudes of the frequencies in the range of 3 to 11 Hz and divides it with the number of frequencies in this range. We represent the resulting value with $P_{\text{D/W}}$. $P_{\text{D/W}}$, in essence, is the power spectral density over the range of 3 to 11 Hz. Similarly, the segmentation block also calculates P_{Walk}, which is the sum of the magnitudes of the frequencies in the range of 11 to 20 Hz divided by the number of frequencies in this range. If the movement in the given segmented denoised-stream is a door/window movement, then as per our observations in the last paragraph, $P_{\text{D/W}}$ must be much greater than P_{Walk} compared to if the the movement is a walking movement. Fig. 7 shows the box plots of the ratio $P_{\text{D/W}}/P_{\text{Walk}}$ for the samples in our traces. We can see that this ratio is much larger for door/window movements compared to human walk. To determine whether the movement in the given segmented denoised-stream is a human movement or a door/window movement, the segmentation block compares the ratio $P_{\text{D/W}}/P_{\text{Walk}}$ with a threshold R. If the ratio is greater than R, the segmentation block identifies the movement in the given segmented denoised-stream as a door/window movement and raises an alarm. If the ratio is less than R, it identifies the movement as a human walk and raises an alarm if WiHS has been armed in the *away-mode*. We have empirically determined the value of R to be 20, as we will show later in the evaluation section.

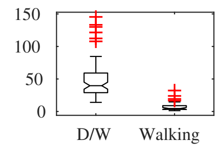


Fig. 7. $P_{\text{D/W}}/P_{\text{Walk}}$ for D/W and walk

III. Door/Window Identification

A. Intuition

As different doors and windows are situated at different angles from the WiFi receiver in any given house, the variations that their movements introduce in the CSI measurements are different. This intuition follows from the observation that Virmani *et al.* demonstrated in [10] that when a human performs any given gesture while facing in different directions, the variations introduced in the CSI measurements by the same gesture are different. As an example, Fig. 8(a) plots two denoised-streams from two samples of fully opening door 1 in House # 1 (floor plans will be described later in Fig. 12) and Fig. 8(b) plots two denoised-streams from two samples of fully opening door 2. Similarly, Figs. 8(c) and 8(d) plot two denoised-streams from two samples each of fully opening window 1 and window 2, respectively, in House # 1. We observe from Figs. 8(a) and 8(b) that the denoised-streams of the two doors are different from each other due to the difference in their relative positions from the WiFi receiver. We make similar observations from Figs. 8(c) and 8(d) for the two windows. We further observe that for any given door/window, the denoised-streams from the two samples of that door/window are fairly similar. Consequently,

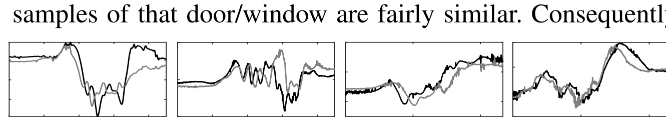

(a) Door 1 (b) Door 2 (c) Window 1 (d) Window 2

Fig. 8. Denoised-streams from two samples of opening

as the movement of any given door/window generates similar variations in the denoised-stream across multiple samples from the same door/window, and as these variations are very different for different doors/windows, by first learning the patterns of change in the denoised-stream introduced by the movement of any given door/window using some training samples, we can use an appropriate machine learning classifier to identify that door/window when it moves at runtime.

B. Feature Extraction

1) Quantifying the Impact of Door/Window Movements on CFR: A transmitted signal arrives at the receiver from multiple paths. Let $a_k(f,t)$ be the complex-valued representation of the initial phase and the attenuation of the k^{th} path at time t for a signal with carrier frequency f. Let $d_k(t)$ represent the length of the k^{th} path at time t. We can represent this aggregate CFR as the sum of a dynamic and a static component. The dynamic component changes as a door/window moves and is the sum of the CFRs of all those paths that arrive at the receiver after reflecting from the moving door/window in the environment. Let P_d represent the set of all those paths that reflect from the moving door/window and arrive at the receiver. The static component is not affected by the movement of any door/window and is the sum of the CFRs of all those paths that arrive at the receiver without reflecting from any moving objects. Let $H_s(f)$ represent the static component of the aggregate CFR. Let v_k represent the rate at which the length of the k^{th} path changes. We call v_k the *speed* of the k^{th} path. Thus, $d_k(t) = d_k(0) + v_k t$. The aggregate CFR power of the signal arriving at the receiver is given by the following well-known equation [14] [11]:

$$|H(f,t)|^2 = \sum_{k \in P_d} 2|H_s(f)a_k(f,t)| \cos\left(\frac{2\pi v_k t}{\lambda} + \frac{2\pi d_k(0)}{\lambda} + \phi_{sk}\right)$$
$$+ \sum_{k,l \in P_d;\ k \neq l} 2|a_k(f,t)a_l(f,t)| \cos\left(\frac{2\pi(v_k - v_l)t}{\lambda} + \frac{2\pi(d_k(0)-d_l(0))}{\lambda} + \phi_{kl}\right)$$
$$+ \sum_{k \in P_d} |a_k(f,t)|^2 + |H_s(f)|^2 \qquad (1)$$

where $2\pi d_k(0)/\lambda + \phi_{sk}$ and $2\pi\left(d_k(0) - d_l(0)\right)/\lambda + \phi_{kl}$ are constants representing initial phase offsets. We observe from Eq. (1) that the only sources of periodic variations in the total CFR power are the two terms $2\pi v_k t/\lambda$ and $2\pi(v_k - v_l)t/\lambda$, and the frequencies in these periodic variations are determined by v_k and v_l. Thus, the frequencies of the sinusoids in the total CFR power are functions of the speeds of path length changes.

To illustrate this, consider a typical 0.9 meters ($\sim$ 3 ft) wide door. From our data sets (which we will describe later in Sec. V), we observed that typically it takes people 2 to 6 seconds to open a door by $90°$. Thus, the speed v_k of the signal path from the door to the WiFi receiver generally lies in the range of about $2 \times \frac{2\times\pi/4\times0.9}{6}$ to $2 \times \frac{2\times\pi/4\times0.9}{2} \sim 0.47$ to 1.41 m/s. As per the first cosine term in Eq. (1), the frequencies that should appear in the CFR power as the door moves should be v_k/λ, *i.e.*, $\approx$ 3 to 11 Hz (using $\lambda \approx 0.125$ m for the 2.4 GHz WiFi band). Figs. 9 and 10 show spectrograms obtained from the denoised-streams resulting from the opening and closing of a door and of a window, respectively. To obtain a spectrogram from any given denoised-stream, we apply short time fourier transform (STFT) on it. STFT slides a window over the denoised-stream, where at each sliding step, it applies fast fourier transform (FFT) on the values covered by the window in that step. As our sampling rate is 300 samples/sec, we chose the window size of 300 samples and step size of 15 samples, which resulted in a good frequency resolution of 1 Hz and time resolution of 50 ms. An FFT at any given sliding window step results in a vector of magnitudes of all frequencies in the portion of the denoised-stream covered by the window in that step. We will refer to this vector as FFT vector. As door movements give rise to frequencies less than 11 Hz, we only use FFT values of the first 11 Hz. A spectrogram of any given denoised-stream is essentially a concatenation of all FFT vectors obtained from sliding the window over that denoised-stream. We indeed observe from Figs. 9 and 10 that the frequencies that appear with high magnitude in the spectrogram lie under approximately 11 Hz. We do see some frequencies > 11 Hz but they have low magnitude, and appear due to faster moving body parts of the person open/closing the door/window. We also see frequencies < 3 Hz, which appear permanently due to the DC or slowly varying components in the denoised-stream. These spectrograms empirically validate the impact of door movement on denoised-stream that we theoretically predicted above using Eq. (1).

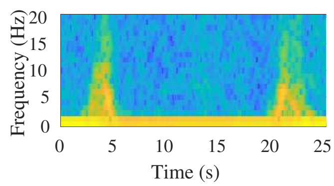


Fig. 9. Spectrogram resulting from the opening and closing of a door

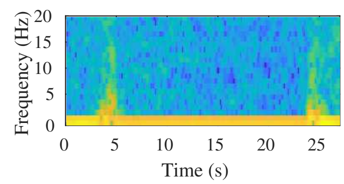


Fig. 10. Spectrogram resulting from the opening and closing of a window

Takeaway: As different doors are situated at different distances and angles from the receiver, different doors have different relative speeds seen by the receiver as they move, due to which, they introduce different frequencies in denoised-streams, which makes denoised-streams from different doors/windows different. Thus, for the purpose of distinguishing across different doors and windows, the best features to use are the magnitudes of different frequencies that appear in the denoised-streams as any door/window moves.

2) Extracting the Features: From Fig. 9, we observe that both door open and close give rise to the same set of frequencies in the CFR power, just that the FFT vectors appear in the reverse order due to the opposite directions of motion. Thus, if we take the average of the magnitudes of any given frequency during a door open/close event across all the FFT vectors, that average should be the same whether the door was opened or closed. The black lines in Fig. 11 show the average magnitude of each frequency during the open and close events shown in Fig. 9. The gray lines show the average magnitudes during an open and a close event of another door. The figure shows that for any given door, regardless of whether the door opens or closes, the average magnitude of each frequency is fairly equal. However, across different doors, the average magnitudes of any given frequency differ. This is inline with our take-away. We made similar observations when using median values instead of the mean values. Thus, the D/W–training block uses the mean and median of each frequency in the range of 3∼11 Hz as features. More specifically, for each door/window open/close event, it generates the spectrogram using the values in the denoised-stream from the start till the end of the event. As a spectrogram has multiple rows (one per frequency), it calculates a mean and a median for each row starting from row # 3 till row # 11, and uses these values as features.

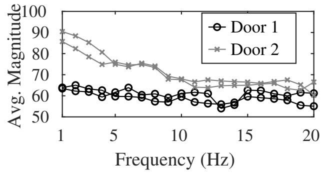


Fig. 11. The mean and median of frequency magnitudes for opening and closing of two doors

Note that as the frequencies that appear when a door opens/closes depend on the speed with which the door opens, we must handle the changes in speed. To make the duration of all door/window movements consistent, the D/W–training block virtually expands/contracts the given sample. Let $\mathcal{T}$ represent the duration of the sample. If $\mathcal{T} > 1$ sec, the D/W–training block virtually contracts the sample by $\mathcal{T} - 1$ seconds to normalize its duration to one second, and vice versa. Expanding/contracting a sample implies that the door was moved at a slower/faster rate, which in turn implies that the frequencies in the aggregate CFR power must be decreased/increased depending on the extent of expansion/contraction of the given sample. To achieve this, the D/W–training block re-scales the y-axis of the spectrogram by simply multiplying it with $\mathcal{T}$. To see why, consider a door that moves by distance d in time $\mathcal{T}$. Thus the speed v of the change in path length is $\frac{2d}{\mathcal{T}}$, and the frequency F that appears in the CFR power is $F = \frac{v}{\lambda} = \frac{2d}{\mathcal{T}\lambda}$. Rearranging this, we get $d = \frac{\mathcal{T}\lambda F}{2}$. If the same door movement was to happen in 1 sec, the distance moved by the door is same d. Let us represent the frequency resulting from this door movement by f. Through the same derivation as above, we get $d = \frac{\lambda f}{2}$ for time 1 sec and frequency f. Equating the two equations, we get $f = F \times \mathcal{T}$.

C. Classifier Training

The D/W–training block generates an independent single class classification model for each door/window. The motivation behind using single class classifiers is that due to the differences in the physical positions of the doors and windows, different doors and windows give rise to different sets of frequencies in the range of 3 to 11 Hz that are more prominent. If we use frequencies that are more tailored for each door, naturally, we get higher accuracy. To select the appropriate frequencies for any given door/window, the D/W–training block takes the mean and median values for each frequency, trains a classier (classifier training will be discussed shortly) using the two-valued feature vector from each training sample of that door/window and calculates the true positive rate of the resulting classifier through 10-fold cross validation. Next, the D/W–training block discards all those frequencies for this door/window that attain less than 80% true positive rate. After selecting all frequencies for the given door/window, the D/W–training block uses their medians and means as features to train the classifier for this door/window. We call the set of mean and median values of the selected frequencies as feature vector. In our implementation, we chose Support Vector Distribution Estimation (SVDE) with the Radial Basis Function (RBF) kernel as our single class classification model.

To generate an SVDE classification model for any given door/window, the D/W–training block first normalizes the values of each feature in the feature vectors of that door/window from all training samples of that door/window to bring them between 0 and 1. The normalization keeps the features with larger values from dominating the classifier training and hurting the accuracy. The D/W–training block finds the optimal values of the tunable parameters of SVDE by performing grid search along with 10-fold cross validation and selecting those values for the tunable parameters that give highest accuracy. Finally, the D/W–training block trains an SVDE classifier and saves this classification model in a database for use at runtime.

If there are D doors and W windows, the D/W–training block generates $D + W$ classification modes. To enroll any new door/window, all WiHS has to do is acquire its training samples, train a classifier, and store it in the database. This enables incremental deployment of WiHS instead of having to enroll all doors/windows before WiHS becomes usable.

D. Runtime Identification of Door/Window

When the segmentation block reports that a door/window event has happened, the D/W–Identification block takes the denoised-stream of the event, generates a spectrogram, multiplies its y-axis with the duration of the event, and calculates the mean and median of each frequency. To evaluate the detected movement against the classification model of any given door/window, the D/W–identification block selects the mean and median magnitude values of the same frequencies that the D/W–training block selected for the given door/window, and obtains a feature vector of the same length that D/W–training block used in generating the classification model of the given door/window. Next, the D/W–identification block scales the values in this feature vector using the same scaling factors that the D/W–training block used during training, and evaluates this normalized vector against the classification model of the given door/window to calculate a likelihood value. It calculates a likelihood value from all $D + W$ classification models, and declares the movement to be originating from that door/window whose model returned the highest likelihood.

IV. Open/Close Identification

1) Intuition: Referring back to Fig. 2, we observe that the variations introduced by door open in the denoised-stream are opposite to those by door close, which is intuitive as door open is the opposite movement of door close. Based on this, we chose to employ the simple, yet very effective, approach of dynamic time warping (DTW) to determine whether the movement was an open event or a close event. DTW calculates a score between any given pair of time-series: *the more similar the time-series, the smaller the DTW score*.

2) Runtime Identification of Open/Close: As soon as the D/W–identification block identifies the door/window that moved, the O/C–identification block takes the denoised-stream for that event and compares it with all training samples of the opening and closing of the identified door/window and calculates a DTW score from each training sample. If the average DTW score is smaller with the opening training samples, the O/C–identification block declares the event to be door open, otherwise door close. Note that the door/window does not have to be completely opened or closed. The O/C–identification block is able to distinguish between open and close movements even when the door is partially moved because in case of opening event, the DTW score with the training samples of the opening of the identified door/window is still smaller than that with the training samples of the closing of that door/window. The only effect that the partial door/window movement has is that the magnitude of the difference between the average DTW score with opening training samples and the average DTW score with closing samples decreases, which impacts the final decision only minimally.

V. Performance Evaluation

We implemented WiHS using commodity devices that included TL-WR7500 WiFi access point (AP) and Lenovo X200 laptops, each with an Intel 5300 WiFi NIC and 3 omnidirectional antennas. We used the tool presented in [17] to acquire CSI measurements at a sampling rate of 300 samples/sec.

A. Test Houses

Fig. 12 shows the floor plans of our three test houses along with the locations where we placed the WiFi AP (TX) and the receiver laptops (RX). In each house we used one AP, but a different number of receivers due to different sizes of the houses. House (H) # 1 is a 1-bedroom apartment with 12 inch thick exterior walls made of brick and mortar and 6 inch thick interior walls made of sheet rock. The doors are made of 1.75 inch thick wood. The widths of doors D1 and D2 and sliding windows W1 and W2 are 36, 28, 32, and 32 inches, respectively. H # 2 is a 2-bedroom apartment with similar types of walls as H # 1. D1, D2, and D3 are each 28 inch wide and 1.75 inch thick made of wood, and W1 to W4 are all 32 inch wide. H # 3 is a 4-bedroom condo with 16 inch thick wooden exterior walls and 6 inch thick interior sheet rock walls. D1 to D6 are all 1.375 inch thick wooden with widths between 28 and 32 inches. W1 through W4 are all 48 inch wide. While collecting data, in addition to our own AP, H # 1, 2, and 3 received signals from 10, 6, and 12 other APs, respectively, of neighboring homes.

B. Overall Accuracy

1) Identification of the Door/Window that Moved: To study the overall accuracy of WiHS in identifying which door/window moved, we collected 60 open and 60 close events for each door and window in each house in 4 sessions (15 open and 15 close events per session). For each door, we asked our volunteers to open and close the door from outside to imitate an intrusion. Most volunteers took 2 to 4 seconds to open or close a door by 90°. We collected 480, 840, and 1200 samples for H # 1, 2, and 3, respectively, over five weeks.

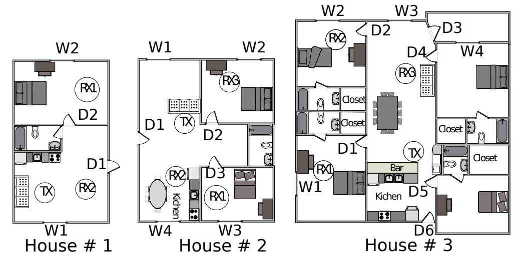


Fig. 12. Floor plans of the three test houses

For this evaluation, we assigned the same label to the open and close samples of any given door/window because we are just identifying which door/window *moved*. For each house, we took the 120 samples of each door/window and performed 10-fold cross validation. In each of the 10 classification rounds, we used the method described in Secs. III-B and III-C for training and Sec. III-D to evaluate the samples of test folds.

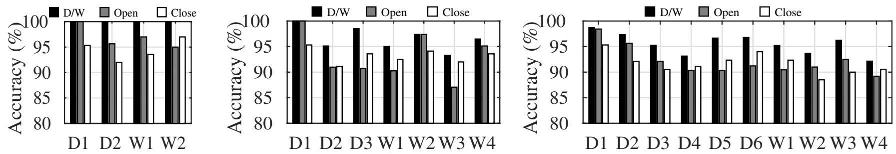
The black bars in Figs. 13(a) – 13(c) show the percentage of 120 samples of each door and window in the three houses that WiHS identified correctly. We see from Fig. 13(a) that WiHS identified all doors and windows with 100% accuracy in H # 1 due to the relatively smaller number of doors and windows. The accuracies for H # 2 and 3 are slightly lower, but still over 95% across all doors and windows. Table I shows the confusion matrix for house number 2. We have not included the confusion matrix for H # 1 as it is an identity matrix and for H # 3 due to the similarity of observations as for H # 2. We see in Tab. I that WiHS made some mistakes for the doors and/or windows that were very close to each other. This is intuitive because different doors and windows affect CSI measurements differently due to the differences in their positions. This slight error does not have significant practical implications because as long as the system points to a door/window that is close to the actual door/window through which an intruder may be breaking in, the home-occupants/law-enforcement still get the correct information about where the intrusion started from.

	D1	D2	D3	W1	W2	W3	W4
D1	1	0	0	0	0	0	0
D2	0	.95	.04	0	.01	0	0
D3	0	.01	.99	0	0	0	0
W1	0	0	.01	.95	.01	.03	0
W2	0	0	0	0	.97	.03	0
W3	0	0	0	.03	.04	.93	0
W4	0	0	0	.01	0	.02	.97

TABLE I
Confusion matrix for House # 2

2) Distinguishing between Open/Close Events: To study this, we used the same data set as in Sec. V-B1 and performed 10 fold cross validation, this time for each window/door separately. Figs. 13(a) to 13(c) plot the percentage of 60 open and 60 close samples of each door and window identified correctly in the three houses. We observe from these figures that WiHS identified the open and close events of the majority of doors/windows with at least 90% accuracy. The average accuracy of WiHS in identifying the door open/close events was 95.7%, 93.1%, and 91.9% for H # 1, 2, and 3, respectively.

3) Distinguishing Between Inside and Outside Movements and Between Human and Door/Window Movements: To study this, we collected CSI measurements for 60 instances of



(a) House # 1 (b) House # 2 (c) House # 3

Fig. 13. Percentage of samples of each door/window from which WiHS identified that door/window correctly and from which WiHS correctly detected whether the door/window was opened or closed

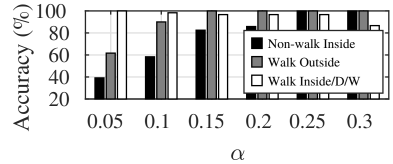


Fig. 14. Segmentation accuracies of WiHS with different values of α

walking outside H # 2, where in each instance a volunteer walked outside along the path from W1 to D1 to W4 at a distance of about 2 ft from the wall. We also collected CSI measurements for 60 instances of walking inside H # 2, where in each instance, the volunteer walked a random path either between D1 and D2 or between W1 and W2. We further collected CSI measurements for 30 open and 30 close events of D1, 30 sitting down and 30 standing up events on the sofa and the dining chair, 30 waving hand events, where the volunteer randomly chose a different position in the house for each waving hand event, and 30 eating/drinking events sitting on the dining table chair.

To distinguish between the movement outside the house and inside, as well as between the non-walking movements and the walking/door/window movement inside the house, we used the method in Sec. II-1. We used $\alpha = 0.25$ to ignore any movements that happen outside the house as well as any non-walking movements that happen inside. We experimented with various values of α ranging from 0.05 to 0.30. To distinguish between walking inside the house and door window movements, we used the method described in Sec. II-2.

Fig. 14 plots the percentage of instances of walking outside the house that WiHS ignored correctly using different values of α. It also shows the percentage of non-walking instances inside the house (*i.e.*, sitting down, standing up, waving hand, and eating/drinking) that WiHS ignored correctly. It further shows the percentage calculated collectively over the 60 instances of walking inside and the 30 open and the 30 close events of D1 that WiHS detected correctly. We observe from this figure that, WiHS is able to accurately ignore 100% of the walking instances outside the house when $\alpha \geq 0.15$ and 100% of the non-walking instances inside the house when $\alpha \geq 0.25$. We also observe that WiHS is able to correctly detect the walking and door/window movement 100% of the times at $\alpha = 0.05$. However, the accuracy decreases as α increases. Therefore, to keep a balance between correctly ignoring non-walking movements and accurately detecting walking and door/window movements, we have chosen $\alpha = 0.25$ in WiHS, where WiHS correctly ignores non-walking movements inside the house with an accuracy of 100% and correctly detects the walking and door/window movements with 96.7% accuracy.

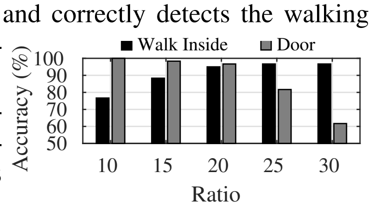


Fig. 15. Accuracy of distinguishing between walk & door/window movt.

Fig. 15 plots the percentage of walking instances inside the house that WiHS correctly identified as walking and of 30 door open and 30 close instances that it correctly identified as door movements for different values of R. We observe from this figure that as R increases, the percentage of walking instances that WiHS correctly recognizes increases. However, increasing R too much increases incorrect identification of door open/close instances. WiHS achieves highest accuracy when $R = 20$, where it correctly identifies walking and door open/close instances 95% and 96.7% of the times, respectively.

C. Impact of Real-World Characteristics

1) Number of WiFi Receivers: To evaluate the impact of the number of WiFi receivers on the accuracy of WiHS, we used the same data set that we used in Sec. V-B1. Just like in Sec. V-B1, we measured the overall accuracy of WiHS in identifying the door/window that moved, this time using CSI measurements from 1 receiver, from 2 receiver, and from 3 receivers separately. Fig. 16 plots the accuracy of WiHS using 1, 2, and 3 receivers in each house. For H # 1, we do not show accuracy with 3 receivers as we only used 2 receivers in it. We observe from this figure that in each house, the accuracy increases with the increase in the number of receivers. We further observe that the houses that are bigger in size, naturally, need more receivers to achieve a target accuracy.

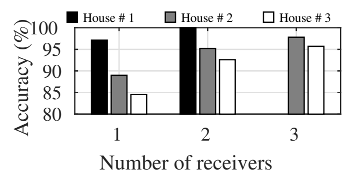


Fig. 16. Accuracy vs. # of receivers

2) Speed of Opening/Closing the Door/Window: We collected additional samples where we asked a volunteer to open and close D1 and D2 in H # 2, where each door open and close took approximately two seconds. We collected 15 samples of door open and 15 of door close for each of the two doors. Next, we collected same data for 4 second and then for 6 second durations of door open and close. This way, we collected 180 new samples. To evaluate these samples, we used samples for all 7 doors and windows that we collected in Sec. V-B1 and trained the classifiers using the method described in Secs. III-B and III-C. Next, we evaluated each of the 180 newly collected samples using the trained classifiers and obtained a decision for each sample. Our results showed that WiHS achieved accuracies of 98.7%, 100%, and 95.1% in identifying the 60 samples of D1 and D2 with 2, 4, and 6 second opening/closing times, respectively. This shows that WiHS is not significantly impacted by the speed with which a door/window moves. The high accuracy is the result of the approach described at the end of Sec. III-B2 to normalize the speed of the moving door/window before classification.

3) Status of the Other Doors and Windows in the House: We collected 30 open and 30 close samples of D3 in H # 2. Before collecting each sample, we randomly changed the state of the other doors and windows. We first evaluated the accuracy in identifying that the moving door was D3 from each of the 60 samples. For this, we used all samples for all 7 doors and windows described in Sec. V-B1 and trained 7 classifiers. When we were collecting samples for Sec. V-B1, we kept all doors open and windows closed. Next, we evaluated the 60 newly collected samples against these 7 classifiers one by one and obtained a decision for each sample as to which class it belongs to. Second, we evaluated WiHS's accuracy in distinguishing between the open and close events of D3 from each of the 30 open and close samples. For this, we used all open and close samples for D3 in H # 2, described in Sec. V-B1, and evaluated each of the 30 open and 30 close samples.

WiHS achieved 97.6% accuracy in identifying that D3 was moving, which is similar to the 98.5% accuracy for D3 in Fig. 13(b). Our results further showed that WiHS achieved 91.5% accuracy in correctly identifying whether D3 was opened or closed, which is similar to the 92.2% accuracy in Fig. 13(b) . These observations show that the open/close status of other doors and windows does not have any noticeable impact on the accuracy of WiHS. This is because WiHS relies on measuring the changes in the denoised-streams caused by the *movement* of a door/window. The multi-paths reflected from doors that are already open or close are part of $H_s(f)$ in Eq. (1) and thus do not contribute any frequencies to the denoised-stream.

4) Partial Opening/Closing of Door/Window: To evaluate how partial opening/closing of a door (*i.e.* by $>45°$ and $<90°$) or window (*i.e.* by greater than half way and less than all the way) impact the accuracy, we asked a volunteer to partially open and close D1 and W2 in H # 2. We collected 30 samples each for door open, door close, window open, and close and evaluated them against the models trained in Sec. V-B1. WiHS achieved accuracies of 100% and 96.7% in identifying that the moving entity was D1 and W2, respectively. This accuracy is very similar to what we saw in Tab. I. This shows that WiHS can accurately identify the moving door/window even when the door/window is not fully opened/closed.

5) Presence and Movements of Other People: To study how the presence and movements of occupants impact WiHS, we collected samples from D1 in H # 2 in 5 scenarios (S1 to S5), where in each scenario, we collected 30 open and 30 close samples. In S1, an occupant was sleeping in room 1 while we collected samples. In S2, an occupant used his phone sitting on the bed in room 1 while we collected samples. In S3, an occupant practiced yoga in room 2, in S4, an occupant walked in the vicinity of D1 but outside the house, and in S5, an occupant cooked in the kitchen while we collected samples.

We evaluated the 60 samples of D1 in each scenario in the same way as we evaluated the 60 samples of D3 in Sec. V-C3. Fig. 17 plots the accuracy of WiHS in identifying that D1 was moving and if it was opened or closed. We see that for S1 to S4, WiHS achieved the same 100% accuracy in identifying D1 as we saw in Sec. V-B1. However, in S5, the accuracy dropped to 58% due to a lot of dynamic activities by the occupant such as opening and closing refrigerator, cutting vegetable, sitting and standing, *etc.* Similarly, in determining whether the door was opened or closed, the accuracies dropped in the scenarios where movements of the occupant were more prominent. We emphasize that while the accuracy of WiHS dropped in S5, *its accuracy in detecting that a movement has occurred remained unaffected. WiHS still raises an alarm;* its just that in such a scenario, it does not accurately determine which door moved. Thus, WiHS is best useful when the occupants have all gone outside, *i.e.*, the away-mode, or when they are not performing any major activities (such as going to bed at night), *i.e.*, the stay-mode, and need the home to be monitored for intrusions. These two times, *i.e.*, when occupants are not home or at night time, are usually the two times where users of conventional security systems arm them to perform security monitoring.

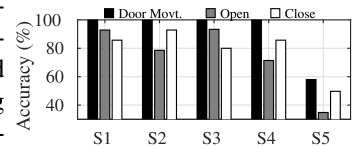


Fig. 17. Impact of presence & movement of people on WiHS's accuracy

VI. Related Work

Wireless signal based. Several prior works use received signal strength to determine the presence of humans in the given environment (mostly at a room level) [18], [19]. Recently, Wu *et al.* proposed DeMan that utilized temporal variations in CSI measurements to detect a moving person and breathing-induced periodic patterns in CSI measurements to detect a stationary person [20]. We emphasize that this line of work does not overlap with the movement segmentation component of WiHS because the focus of prior work is only to determine whether one or more humans are present in the coverage area of a WiFi transceiver, whereas the focus of the movement segmentation component of WiHS is not only to detect movements in a given house but also to determine whether the movement was inside the house or outside, or whether the movement was from a human or doors/windows.

Another, relatively more recent, line of work uses CSI measurements to recognize human gestures and activities [11], [14], [15]. Wang *et al.* modeled the impact of human activities on CSI measurements and used this model to develop a single user activity recognition system [14]. Venkatnarayan *et al.* extended the work in [14] and modeled the impact of activities of multiple simultaneously moving humans on CSI measurements and used this model to develop a multi-user activity recognition system [11]. Xi *et al.* analyzed the relationship between the CSI variation and the number of moving people to count people in a crowd [21]. E-eyes used distributions of the CSI measurements to recognize activities such as cooking, bathing *etc.* [22]. SIED detected human intrusion using the variance in CFR [23]. Li *et al.* proposed AR-Alarm that distinguishes human movements from movements of everyday objects, such as curtains [24]. While SIED and AR-Alarm detect human motion inside the house, they can not distinguish between movements inside and outside the house and further do not detect whether a door/window was opened/closed.

Sensor based. Many other sensors such as accelerometers, passive infrared (PIR) motion sensors, vibration sensors, barometers, and even cameras have been used to detect door/window/human movements. Commercial products such as [25], [26] can detect an intruder by sensing vibration when a door or window is opening. In [27], an energy-harvesting vibration sensor was proposed that detects door open events when attached to the door. Patel *et al.* utilized pressure sensors installed in the air conditioning ducts to detect pressure variation caused by door open/close events [28]. Wu *et al.* [29] used barometer in a smartphone to detect door open/close events anywhere inside a building. These pressure sensor and barometer based approaches usually do not distinguish across different doors, especially if the doors are of the same size.

Today's commercial home security systems use magnetic contact sensors to detect door/window open/close events and PIR sensors to detect human motion. As me mentioned in the introduction, while these provide deterministic intrusion detection, WiHS is not competing with them, rather is providing a solution for homes that otherwise do not have a security system at all. In many ways, WiHS is actually complimentary to these commercial systems. For example, PIR sensors detect human motion only in the line of sight, while WiHS can detect any motion in non line of sight as well. Similarly, due to their cost, conventional systems deploy contact sensors only at the main entrance. While they monitor the main entrance of a house, WiHS can monitor the door movements inside the house, which is important for post-incident analysis (*e.g.*, to determine what locations inside the house an intruder visited).

VII. Conclusion and Future Work

In this paper, we proposed WiHS, a WiFi based home security system that accurately performs the three primary monitoring tasks of the typical home security systems. The key technical novelty of WiHS lies in developing the theoretical understanding of the impact of the movements of doors and windows on CSI measurements. The key technical depth of WiHS lies in the techniques that it uses to 1) distinguish between the movements inside and outside the house, 2) distinguish between walking, non-walking, and door/window movements, 3) identify the door/window that has moved, and 4) determine whether the moving door/window opened or closed. The results from an extensive evaluation of WiHS on commodity WiFI devices in three different houses show that WiHS *detects* intrusions with over 95% accuracy.

WiHS also has its share of limitations. For example, WiHS works best in a single story house. For a multi-story house, each floor will need its own instance of WiHS deployed using the hardware on that floor. WiHS also requires that the locations of transceivers stay fixed. This means that to deploy it on existing equipment, the house will need WiFi receivers that do not change their position, such as a desktop computer. One could also achieve this by, for example, using a few raspberry Pis and attaching them to various wall-sockets in the house. This, however, entails some monetary cost. We plan to address these aspects in our future work.

References

[1] S. Lee, "The impact of home burglar alarm systems on residential burglaries."

[2] A. Tseloni, R. Thompson, and N. Tilley, *Reducing Burglary*. 2018.

[3] www.samsung.com/us/smart-home/smartthings/sensors

[4] www.adt.com/door-window-alarms

[5] J. J. Roth, "The role of perceived effectiveness in home security choices," *Security Journal*, vol. 31, no. 3, pp. 708–725, 2018.

[6] Burglary crime statistics and facts. www.alarms.org/burglary-statistics/

[7] www.chinairr.org/report/R03/R0304/201808/27-271833.html

[8] E. Soltanaghaei, A. Kalyanaraman, and K. Whitehouse, "Multipath triangulation: Decimeter-level wifi localization and orientation with a single unaided receiver," in *MobiSys*, 2018, pp. 376–388.

[9] K. Qian, C. Wu, Y. Zhang, G. Zhang, Z. Yang, and Y. Liu, "Widar2. 0: Passive human tracking with a single wi-fi link," *MobiSys*, 2018.

[10] A. Virmani and M. Shahzad, "Position and orientation agnostic gesture recognition using wifi," in *MobiSys* 2017.

[11] R. H. Venkatnarayan, G. Page, and M. Shahzad, "Multi-user gesture recognition using wifi," in *MobiSys* 2018.

[12] M. Shahzad and S. Zhang, "Augmenting user identification with wifi based gesture recognition," *IMWUT*, vol. 2, no. 3, 2018.

[13] K. Ali, A. X. Liu, W. Wang, and M. Shahzad, "Keystroke recognition using wifi signals," in *MobiCom*, 2015.

[14] W. Wang, A. X. Liu, M. Shahzad, K. Ling, and S. Lu, "Understanding and modeling of wifi signal based human activity recognition," in *MobiCom*, 2015.

[15] W. Wang, A. X. Liu, and M. Shahzad, "Gait recognition using wifi signals," in *IMWUT*, 2016.

[16] Y. Xu, W. Yang, J. Wang, X. Zhou, H. Li, and L. Huang, "Wistep: Device-free step counting with wifi signals," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 1, no. 4, pp. 172:1–172:23, Jan. 2018.

[17] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, "Tool release: Gathering 802.11 n traces with channel state information," *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 1, pp. 53–53, 2011.

[18] M. Moussa and M. Youssef, "Device-free passive detection in real environments," in *Pervasive Computing and Communications, 2009. PerCom 2009. IEEE International Conference on*. IEEE, 2009, pp. 1–6.

[19] J. Yang, Y. Ge, H. Xiong, Y. Chen, and H. Liu, "Performing joint learning for passive intrusion detection in pervasive wireless environments," in *INFOCOM, 2010 Proceedings IEEE*. IEEE, 2010, pp. 1–9.

[20] C. Wu, Z. Yang, Z. Zhou, X. Liu, Y. Liu, and J. Cao, "Non-invasive detection of moving and stationary human with wifi," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 11, pp. 2329–2342, 2015.

[21] W. Xi, J. Zhao, X.-Y. Li, K. Zhao, S. Tang, X. Liu, and Z. Jiang, "Electronic frog eye: Counting crowd using wifi," in *Infocom, 2014 proceedings IEEE*. IEEE, 2014, pp. 361–369.

[22] Y. Wang, J. Liu, Y. Chen, M. Gruteser, J. Yang, and H. Liu, "E-eyes: device-free location-oriented activity identification using fine-grained wifi signatures," in *Proceedings of the 20th annual international conference on Mobile computing and networking*. ACM, 2014, pp. 617–628.

[23] J. Lv, W. Yang, L. Gong, D. Man, and X. Du, "Robust wlan-based indoor fine-grained intrusion detection," in *2016 IEEE Global Communications Conference (GLOBECOM)*, Dec 2016, pp. 1–6.

[24] S. Li, X. Li, K. Niu, H. Wang, Y. Zhang, and D. Zhang, "Ar-alarm: An adaptive and robust intrusion detection system leveraging csi from commodity wi-fi," in *Enhanced Quality of Life and Smart Living*, 2017.

[25] www.zwaveproducts.com/shop/z-wave-security/ z-wave-door-and-window-sensors/door-window-sensor-gen-5

[26] www.vigilantpps.com/products/mace-slimline-wireless-door-window-sensor

[27] B. Campbell and P. Dutta, "An energy-harvesting sensor architecture and toolkit for building monitoring and event detection," in *Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings*. 2014.

[28] S. N. Patel, M. S. Reynolds, and G. D. Abowd, "Detecting human movement by differential air pressure sensing in hvac system ductwork: An exploration in infrastructure mediated sensing," in *International Conference on Pervasive Computing*. 2008.

[29] M. Wu, P. H. Pathak, and P. Mohapatra, "Monitoring building door events using barometer sensor in smartphones," in *UbiComp*, 2015.