One-Class Support Vector Machine for WiFi-based Device-free Indoor Presence Detection

Anatolij Zubow, Kim Petto, Falko Dressler
Technische Universität Berlin, Germany
{zubow, petto, dressler}@tkn.tu-berlin.de

***Abstract*—The utilization of existing radio signals such as 802.11 (WiFi) for device-free detection of human presence and movement indoors has garnered significant interest among researchers in academia and industry. Improving the efficiency of buildings, particularly in terms of heating and energy costs, relies on accurately detecting room occupancy. Our approach uses channel state information (CSI) obtained from commodity 802.11ac hardware as input to machine learning based on One Class Support Vector Machine (OC-SVM). Unlike other methods that necessitate extensive learning in environments with and without human presence, our approach treats human presence as a novelty. This simplifies the training process, as we only need to learn from environments without human presence, specifically empty rooms. Furthermore, since we focus solely on analyzing the magnitude information of the CSI data, there is no requirement for intricate sanitization of the phase information. Experimental results using standard WiFi hardware demonstrate exceptional performance, with accuracy, sensitivity, and specificity exceeding 97% in most cases. Furthermore, our proposed approach is practical, as it incurs minimal overhead in terms of radio resource usage. Simply capturing CSI data with a sampling rate of 5 Hz on only a few OFDM subcarriers from a 5 MHz channel is sufficient.**

***Index terms*—** wireless sensing, device-free detection

I. Introduction

The device-free detection of human presence finds applications in security, healthcare, and smart buildings. Traditional methods often rely on optical cameras and Passive Infra-Red (PIR) sensors. However, these technologies have certain limitations. Cameras require a direct line of sight and raise privacy concerns. PIR sensors, on the other hand, are unable to detect static individuals. An alternative approach is to passively analyze radio signals such as 802.11 WiFi, which are already extensively deployed in residential and enterprise areas. This alternative, termed as wireless sensing, offers several advantages, including accessibility, convenience, and cost-effectiveness, as existing WiFi infrastructure can be utilized at zero additional cost.

The wireless sensing methods function as follows. When a wireless signal like WiFi is transmitted from devices such as smartphones, it travels through various paths to reach the receiver, such as an access point. These paths can be direct when there is a line of sight (LOS) between the transmitter and receiver, or indirect when the signal reflects off walls and objects. As a result, the Channel State Information (CSI) extracted from the received WiFi signal contains valuable information about the surrounding environment. This information can be utilized to conduct device-free WiFi sensing.

There are a variety of wireless sensing applications ranging from activity detection, such as gesture recognition [1], [2], localization [3], and fall detection [4]. In these applications, the utilization of unique signatures that emerge during signal propagation on the CSI is crucial while activities are being performed. In the specific case of presence detection, previous studies have shown that even when a person remains stationary, there are alterations in the signal propagation caused by factors such as breathing [5] or prolonged slight movements.

While known solutions for device-free presence detection have shown to achieve high accuracy [6], there remains a need for low-complexity and robust solutions. Presence detection typically requires training data that encompasses both the presence and absence of humans in a well-defined environment. Collecting data for human absence can be relatively straightforward by transmitting radio signals through an empty room. However, gathering data for human presence can be challenging as it requires capturing various scenarios, including static individuals, people in motion, and individuals in different positions, ideally in all possible locations within the room. Therefore, by training a classifier solely with data collected in an empty room, it is possible to reduce the reliance on the calibration of human presence data collection.

Contributions: We propose an approach for presence detection that involves analyzing CSI data using One Class Support Vector Machine (OC-SVM). Specifically, we focus solely on analyzing the magnitude information of the CSI data. The system is trained using CSI data collected from empty rooms only. The actual presence detection is based on novelty detection. Experimental results using commodity WiFi hardware demonstrate exceptional performance, with accuracy, sensitivity, and specificity exceeding 97% in most cases. This high level of performance is achieved by simultaneously analyzing the temporal, frequency (OFDM), and spatial (MIMO) domains of the CSI data. Additionally, our proposed approach is practical as it imposes minimal overhead in terms of used radio resources. A sampling rate of just 5 Hz for CSI data on a few OFDM subcarriers from a 5 MHz channel is sufficient. Lastly, the use of multiple antennas enhances the detection performance.

This work was supported in part by the Federal Ministry of Education and Research (BMBF, Germany) within the 6G Research and Innovation Cluster 6G-RIC under Grant 16KISK020K.

II. Related Work

An overview of wireless sensing for detection of human activity is given by Liu et al. [7]. Accordingly, the majority of proposed applications rely on processing RSSI and CSI data as techniques like Frequency Modulated Carrier Wave (FMCW) rely on customized hardware. In contrast to RSSI-based detection approaches which process CSI data are powerful as CSI allows to perform very fine-grained channel measurements (i.e., both amplitude and phase information for multiple subcarriers). Most approaches found in literature rely on analyzing amplitude, phase or phase difference to detect the presence of a human indoors. Nishimori et al. [8] analyzed the influence of antenna arrangement on the wireless signal propagation in typical indoor environments and proposed an intrusion detection system based on processing CSI from MIMO channels. DeMan [9] is based on the extraction of the maximum eigenvalues of the covariance matrix from successive CSI data, which is including both amplitude and phase. Ding et al. [10] proposed to analyze the phase difference between adjacent antenna pairs for passive device-free motion detection. According to Zou et al. [11] the shape similarity of multiple OFDM subcarriers could be used as a feature for input to supervised learning like random forest. Wu et al. [12] proposed the use of learning based on Support Vector Machine (SVM) for the detection of even a stationary human through its naturally breathing. They use subtle motion of the moving chest to register wave like patterns on the received OFDM signal. Additionally, the phase information of the CSI data can offer rich information about the variation of the channel. However due to phase noise in CSI obtained from commodity hardware, proper preprocessing is needed [13].

III. Background - Channel State Information

A radio signal propagating through the wireless channel to the receiver through multiple paths experiences several effects. Reflection on walls or scattering through obstacles will produce additional copies of the transmitted signal, so-called multipath components. These different components of the signal correspond to pulses at the receiver at different arrival times. The Channel Impulse Response (CIR) is used to describe these impulses. When assuming a time-invariant channel the CIR can be denoted as [14]:

$$h(t) = \sum_{n=0}^{N} a_n e^{-j\phi_n} \delta(t - t_n) \tag{1}$$

where a_n is the amplitude and ϕ_n is the phase of the nth multipath component at the time t and N is the total number of multipath components and $\delta(t)$ is the Dirac delta function. The sum of amplitude and phase of these multipath components can result in constructive or destructive interference, which result in larger or smaller pulses. In the time domain, the received signal $r(t)$ is the convolution of the transmitted signal $s(t)$ and the CIR $h(t)$:

$$r(t) = s(t) \otimes h(t) \tag{2}$$

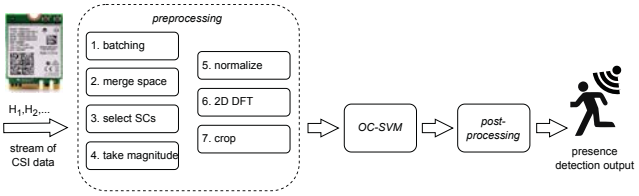

Fig. 1: Flowgraph of proposed device-free presence detection.

Hence in the frequency domain, the received signal spectrum $R(f)$ is the multiplication of the transmitted signal spectrum $S(f)$ and the Channel Frequency Response (CFR) $H(f)$:

$$R(f) = S(f) \times H(f) \tag{3}$$

Using eq. 2 and 3, the CIR can be derived from CFR [14]:

$$h(t) = \frac{1}{P_s} \mathfrak{F}^{-1}\{S^*(f)R(f)\} \tag{4}$$

where $\mathfrak{F}^{-1}$ denotes the inverse Fourier transform, $R(f)$ is the Fourier transform of the received signal $r(t)$ and $S^*(f)$ is the complex conjugate of the Fourier transform of the transmitted signal $s(t)$. Here P_s approximates the transmitted signal power. As each OFDM subcarrier experiences flat fading the received signal r of a subcarrier operating at the center frequency f can be described as:

$$r(f) = H(f)s + n \tag{5}$$

where s is the transmitted symbol vector and n is the additive white Gaussian noise. When using m receive and n transmit antennas, the MIMO-OFDM channel matrix H for an OFDM subcarrier centered at frequency f can be described as:

$$H(f) = \begin{pmatrix} H_{1,1} & \cdot & H_{1,n} \\ \vdots & \ddots & \vdots \\ H_{m,1} & \cdot & H_{m,n} \end{pmatrix} \tag{6}$$

whose complex entries H_{ij} are the sampled CFR from the jth transmit to the ith receive antenna. CSI samples can be estimated at the receiver by using pilot symbols scattered through the OFDM subcarriers.

IV. System Design

Our device-free presence detection scheme consists of four major steps (Fig. 1) which are described in this section.

A. Capturing CSI Data

Standard IEEE 802.11 devices are capable of calculating the CFR from received WiFi frames. Unfortunately, most chips do not provide this functionality as an API and changes must be made to the driver, e.g. [15], [16]. In case of Intel 9260 NIC which we used for our prototype (see §V) the complex CSI data captured from a single WiFi frame has the dimensions:

$$N_{tx} \times N_{rx} \times N_{sc} \tag{7}$$

where N_{rx} and N_{tx} is the number of receive and transmit antennas respectively. N_{sc} is the number of OFDM subcarriers which is dependent from the channel bandwidth, e.g. for a 40 MHz channel it is $N_{sc} = 114$.

B. Preprocessing CSI Data

The raw CSI data captured with a sample rate of S Hz must be preprocessed before it can be passed to ML algorithms. Our CSI pre-processing pipeline is similar to the one proposed by Wu et al. [9] except that we only analyze the magnitude of the CSI data and discard the phase information. This is beneficial as due to phase noise additional non-trivial preprocessing of CSI would be needed [13].

We describe the dimensions of CSI data used as input as:

$$N \times N_{tx} \times N_{rx} \times N_{sc} \tag{8}$$

where N is the Number of CSI samples, N_{rx} is the number of receive antennas, N_{tx} is the number of transmit antennas and N_{sc} is the number of OFDM subcarriers. As we use MIMO hardware we introduce an additional spatial domains represented through the dimensions of N_{rx} and N_{tx}. By using OFDM, the used subcarriers operate at distinct frequencies and open up sensing through the frequency domain. Lastly, through the time series of these CSI, temporal changes of the other two domains can be observed as well.

First, we group our N CSI samples into N_t batches of size B_t, with each batch resembling a time window of $\frac{B_t}{S}$ s. This results in the data dimension of our data array X, given by

$$X = N_t \times B_t \times N_{tx} \times N_{rx} \times N_{sc} \tag{9}$$

Within this time t, we expect a change in the temporal domain due to a person being present, even when this person is static, through small scale motion or even breathing [12]. Note, that B_t is dependent on the CSI sampling rate.

Then we reduce the dimensions by using only f subcarriers, leaving us with $N_{sc} \rightarrow N_f$ subcarriers. We select the f subcarriers evenly distributed across all N_{sc} subcarriers which gives us

$$X^{\text{reduced}} = N_t \times B_t \times N_{tx} \times N_{rx} \times N_f \tag{10}$$

This becomes possible as the subcarrier spacing in 802.11g/ac is 312.5 kHz which is much smaller than the coherence bandwidth in a typical indoor environment. Thus, adjacent subcarriers have a similar progression [6] and a downsampling is feasible.

Then we combine the dimensions of the receive (N_{rx}) and transmit (N_{tx}) antennas. This gives us

$$X^{\text{reshaped}} = N_t \times B_t \times (N_{rx}N_{tx}) \times N_f \tag{11}$$

reducing the dimensional complexity of the array by merging the spatial domain.

The received signal from different antennas are fed into different RF chains. These RF chains are not synchronized and thus the measured CSI will be distorted by the phase offsets between these RF chains [17]. This would require sanitation of the phase information which would complicate the algorithm. We therefore discard the phase information and instead use the magnitude of our CSI only

$$X^{\text{abs}} = |X^{\text{reshaped}}| \tag{12}$$

Afterwards we normalize each grouped batch B_t in respect to the first CSI sample by element-wise division (Hadamard division). This gives us

$$X^{\text{norm}} = X^{\text{abs}} \div x^{\text{abs}}_{:,0,:,:} \tag{13}$$

Now the first entry of any batch B_t consists only of a matrix with the value *1*, while the remaining entries of B_t have a relative value. This is done because the absolute CSI magnitude is highly dependent on the distance between transmitter and receiver and the existence of LOS [6]. With this normalization, we try to remove this environmental information.

Then we apply a 2D DFT to each grouped, normalized packet on the temporal and frequency domain to obtain the Fourier coefficients X^{fft} from which we take the magnitude (abs). Moreover, we shift the zero-frequency component to the center of the spectrum:

$$X^{\text{norm}} \xrightarrow{\text{2D-DFT}} X^{\text{fft}} \xrightarrow{\text{fftshift}} X^{\text{fft+shift}} \xrightarrow{\text{magnitude}} |X^{\text{fft+shift}}| \tag{14}$$

Afterwards, we apply the periodic shift on the 2D DFT output, so the zero frequency component is in the center of the array.

The difference of the minimum and maximum of the Fourier coefficients can be too high to achieve good generalization. Therefore, we apply $\log_{10}(1 + x)$ to obtain

$$X^{\log} = \log_{10}(1 + |X^{\text{fft+shift}}|) \tag{15}$$

Finally, we cut off the left and right side of the matrix on the batch dimension B_t to obtain a further reduced data array B_{tw} with only entries in its central dimension:

$$X^{\text{crop}}_{:,i,:,:} = X^{\log}_{:,\frac{B_t - B_{tw}}{2}+i,:,:}, i = 0, \ldots, B_{tw} \tag{16}$$

This further reduces the data complexity and is done to increase the generalization. Because we used the periodic shift before, the remaining data still contain the most important information of the transformed Fourier coefficients. This results in the final dimension format:

$$X^{\text{preprocessed}} = N_t \times B_{tw} \times (N_{rx}N_{tx}) \times N_f \tag{17}$$

C. Presence Detection using OC-SVM

We denote presence detection as a binary classification problem with:

$$\text{presence} = \begin{cases} 1 & \approx \text{no} \\ -1 & \approx \text{yes} \end{cases}. \tag{18}$$

We assign presence with the class -1 and the absence of presence with the class 1. We therefore consider data corresponding to non-presence as regular and data corresponding to presence as irregular, a novelty.

One-class SVM is an unsupervised algorithm that learns a decision function for novelty detection by classifying new data as similar or different to the training set. The data we use to train this OC-SVM classifier is therefore collected from the regular class, a room without human presence. Then the

trained OC-SVM model can be tested and evaluated using unseen data from both classes.

We transform the input for the classifier in the *2-dimensional* format of $n_{samples} \times n_{features}$. Because we have multidimensional data after our preprocessing with the dimension format given in equation 17, we flatten our data dimensions of B_{tw}, $(N_{rx}N_{tx})$ and N_f into one dimension, which serves us as the features for the input. So, we denote our input data for the OC-SVM classifier as:

$$X = N_t \times (B_{tw}N_{tx}N_{rx}N_f) \tag{19}$$

D. Postprocessing

The OC-SVM could wrongly classify human absence when in reality the human is simply standing still. This is because according to [6] a completely static human has only a low effect on the CSI, making it sometimes indistinguishable from CSI of an empty room. We therefore added an additional postprocessing step as suggested by Liu et al. [6] in which we analyze the last k results from the OC-SVM in order to perform a majority vote. Hence, the appearance of a human in a previously empty room is detected after $\lceil \frac{k}{2} \rceil \times \frac{B_t}{S}$ seconds at the earliest. Note, that our approach might create false classifications during the transition of *presence* to *no presence* and vice versa, i.e. when a person leaves or enters the room. With $k = 5$ we would get at least two false detections during the transition period. If no such transitions happen, we expect to increase the overall detection rate.

V. Implementation

We implemented a prototype of our system using commodity hardware. As experimentation platform, we used standard notebooks equipped with Intel 9260 WiFi NICs. The Intel 9260 is an IEEE 802.11ac wave 2 compliant radio with 2x2 MIMO and channel width of up-to 160 MHz. During the experiments a pair of such nodes was used to create a point-to-point communication link. As operating system we used Ubuntu 18.04 with a patched Linux Kernel to enable the CSI functionality [18]. We run both the transmitter and receiver in *monitor* mode with packet filtering. We implement our presence detector fully in Python. Specifically, we used the implementation *OneClassSVM* of scikit-learn [19] which is based on the work of Schölkopf et al. [20]. During data collection the CSI was stored in a database. The actual presence detection was performed offline. Our code base can be downloaded from: https://github.com/zubow/ocsvm_pd.git.

VI. Evaluation

The proposed approach was evaluated by means of experiments using the prototype described in previous section.

A. Experimental Setup and Data Collection

The experiments were performed within three different apartment rooms. The location of the two WiFi nodes as well as furniture is shown in Fig. 2. The transmitter and receiver inside the rooms *a* and *c* didn't have LOS due to the different heights and position atop of the desk. In order to collect CSI data we transmitted unicast data packets with a transmission rate of 100 Hz using BPSK modulation with a coding rate (FEC) of 1/2 and channel bandwidth of 40 MHz. The experiment was performed in 2.4 GHz band on channels 1 and 5 GHz band on channel 36. Finally, we collected CSI under two scenarios: i) empty (human-free) room and ii) human-present-occupied room. For the latter scenario the human was walking, standing, and sitting, where applicable, randomly inside the room. The person alternates between the above-described actions and performs each action for an unspecified time duration. The collected CSI datasets were labeled accordingly. Each collection session lasts for about 10.5 min, resulting in 63k CSI samples.

Fig. 2: Environment: layout of rooms a, b and c.

TABLE I: Used parameters of our presence detection system.

Parameter	Value
CSI sampling rate S	100 Hz
No.of CSI samples N	63k
No. of transmit/receive antennas $N_{tx} \times N_{rx}$	2×2
No. of OFDM subcarriers N_{sc}	114
No. of time samples per batch B_t	128 ($\widehat{=}$ 1.28 s)
No. of batches N_t	490
No. of downsampled OFDM subcarriers N_f	15
No. of downsampled time samples B_{tw}	14
OC-SVM kernel	rbf
OC-SVM *nu* parameter	0.005
OC-SVM *gamma* parameter	0.01
No. of features $n_{features}$	840
Majority vote over k	5

B. System Parameters

Table I shows our used system parameters for the detector. In the preprocessing step we reduced the dimension of (complex) CSI data from $63k \times 2 \times 2 \times 114$ to just $490 \times 14 \times 4 \times 15$ of (real) data, which corresponds to a compression of around 99.3%. It is served as features for the input of the OC-SVM. With the selected parameters the detection of person entering a room takes place between 3.84 and 6.4 s.

For the training we used the data sets collected from empty rooms. Afterwards, we tested our trained system by using the datasets from both empty and occupied rooms. In order to avoid the repeated use of the same dataset for training and testing, we resample our training data. Data from each room (a, b, c) and used radio channel (1, 36) are forming a data set we use for training and testing. This results in six different training and test data sets and thus in six trained models. Although our presence detector was written entirely

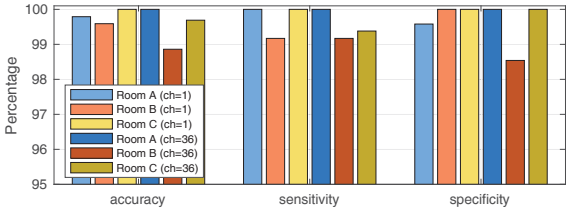

Fig. 3: Performance in 3 rooms (channel 1 and 36).

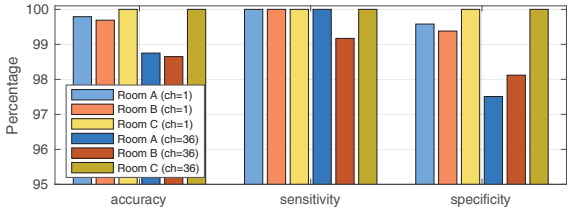

Fig. 4: Performance in 3 rooms (training performed with data from all rooms).

in Python, training took only $\approx 6\,\mathrm{s}$ on a modern PC. This confirms the low complexity of the proposed approach.

C. Performance Metrics

Our metrics for evaluation are the accuracy, sensitivity and specificity. Accuracy is the percentage of correctly classified results. It shows us the overall performance of a model. Sensitivity is the percentage of correctly classified presence. Specificity is the percentage of correctly classified non-presence. These two metrics will be used to get further insight on the performance of a model.

D. Detection Accuracy, Sensitivity, Specificity

In our first experiment we analyzed the performance of our detector in the three different rooms using channel 1 and 36 respectively. Therefore, our detector was learned for each room and channel independently. As can be seen from Fig. 3 we achieved very good performance. The accuracy is ranging from 98.86% up to 100%. The average accuracy is 99.79% for channel 1 and 99.52% for channel 36. Regarding the capability of our system to detect human absence channel 1 achieved an average sensitivity of 99.72%. This is higher than the average sensitivity on channel 36 with a score of 99.51%. On channel 1 and in environment *b* we only achieved a sensitivity of 99.17%. A lower human absence rate of 99.17% and 99.38% was achieved in the environments *b* and *c* on channel 36 respectively. Our system detected human presence with a rate of 100% on multiple environment-channel combinations. Only in environment *a* on channel 1 and *b* on channel 36 we got a lower specificity of 99.58% and 98.54%. We see that in the smaller environment *b* our system achieved the least percentage of correct classifications.

So far, we trained and tested our model independently for each room and channel configuration. Next, we train a single model with data from all rooms at once and test with testing data from each room. Fig. 4 shows the results. The performance loss compared to Fig. 3 is very small, i.e., at most a drop of one percentage point. This confirms that training a single model is sufficient.

E. Impact of Spatial Domain

Next, we want to analyze the gain from having multiple antennas at the transmitter and receiver, i.e., MIMO. The impact of that spatial domain is shown in Fig. 5. The best performance is achieved when using all two transmit and receive antennas resulting in 4 spatial paths from which the CSI can be obtained. By lowering the number of antennas,

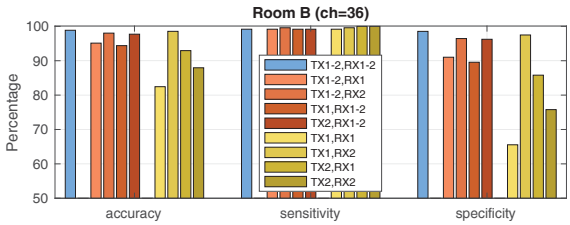

Fig. 5: Impact of spatial domain (antennas).

we see a dramatic drop in performance. With only a single spatial domain, i.e., SISO with single TX and RX antenna, we see for some combination a drop in specificity to only 65%. Also, the accuracy can be low as 82.5%. This confirms the need to use MIMO systems with 2 or more antennas.

F. Impact of CSI Sampling Rate

Estimating the CSI creates overhead as packets need to be transmitted which are consuming valuable radio resources which could be otherwise used for ordinary communication. Therefore, we want to analyze the impact of CSI sampling rate of the detection performance. Fig. 6 shows the results. Both accuracy and specificity are affected by the used CSI sampling rate whereas the sensitivity stays at the maximum. Hence there is a tradeoff between overhead and performance of the detection which has to be taken into account.

G. Impact of Channel Bandwidth

Like the CSI sampling rate the bandwidth used by the probing packets creates overhead in terms of used radio resources. Therefore, in Fig. 7 we analyzed its impact. Again, we see no impact on sensitivity whereas the impact of accuracy and specificity is visible but small.

H. Impact of Downsampling in Frequency Domain

Here, we analyze the impact of downsampling in frequency domain N_f, cf. eq. 10. From Fig. 8 we see that optimal

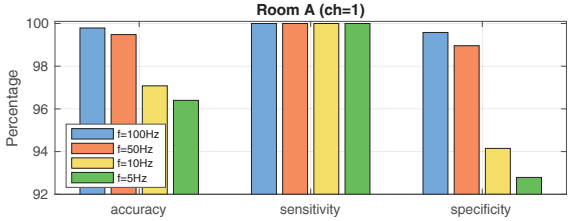

Fig. 6: Impact of CSI sample rate (trained with data from all rooms while tested with data from room A, channel=1).

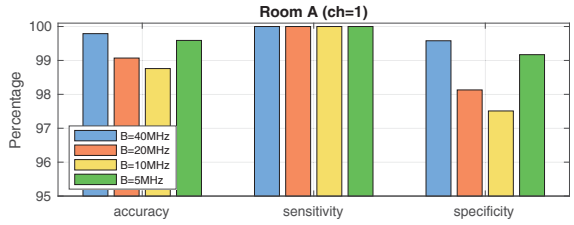

Fig. 7: Impact of CSI bandwidth (trained with data from all rooms while tested with data from room A, channel=1).

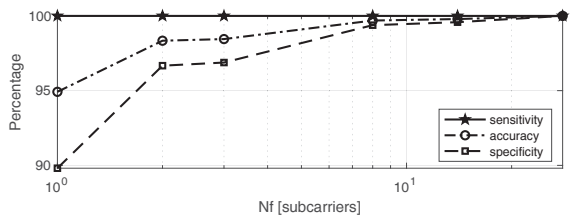

Fig. 8: Impact of downsampling N_f (trained with data from all rooms while tested with data from room A, channel=1).

results can be achieved for $N_f = 28$. However, to keep the computational overhead low the value can be reduced to $N_f = 8$ without significantly affecting the performance of the detector. However, analyzing just a single subcarrier, i.e., $N_f = 1$, is not sufficient.

I. Impact of Batch Size

Finally, we want to study the impact of the selected batch size B_t, cf. eq. 9, on the performance. From Fig. 9 we see that values below 1.2 s for B_t lead to performance degradation especially in terms of specificity which can drop to just 88% when using $B_t = 16$ ms. Hence, we can conclude to have a tradeoff between the accuracy and latency.

VII. Conclusions

In this paper, we proposed a low-complexity device-free human presence detection method which utilizes the CSI provided by commodity 802.11 (WiFi) devices. The key idea is to utilize OC-SVM for novelty detection, hence requiring training to be performed in empty rooms only. It is of low-complexity both in terms of required radio resources, i.e., CSI sampling rate and bandwidth, as well as computational complexity. The proposed approach was prototypically implemented and experimentally evaluated.

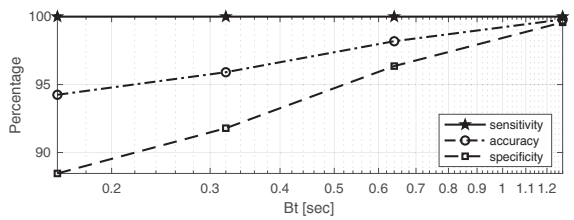

Fig. 9: Impact of batch size B_t (trained with data from all rooms while tested with data from room A, channel=1).

References

[1] J. Shang and J. Wu, "A Robust Sign Language Recognition System with Sparsely Labeled Instances Using Wi-Fi Signals," in *IEEE MASS 2017*. Orlando, FL: IEEE, Oct. 2017.

[2] Y. Ma, G. Zhou, S. Wang, H. Zhao, and W. Jung, "SignFi: Sign Language Recognition Using WiFi," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, no. 1, pp. 1–21, Mar. 2018.

[3] X. Wang, X. Wang, and S. Mao, "Deep Convolutional Neural Networks for Indoor Localization with CSI Images," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 1, pp. 316–327, Jan. 2020.

[4] Y. Wang, K. Wu, and L. M. Ni, "WiFall: Device-Free Fall Detection by Wireless Networks," *IEEE Transactions on Mobile Computing*, vol. 16, no. 2, pp. 581–594, Feb. 2017.

[5] Y. Zeng, D. Wu, J. Xiong, E. Yi, R. Gao, and D. Zhang, "FarSense: Pushing the Range Limit of WiFi-based Respiration Sensing with CSI Ratio of Two Antennas," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 3, no. 3, pp. 1–26, Sep. 2019.

[6] Y. Liu, T. Wang, Y. Jiang, and B. Chen, "Harvesting ambient rf for presence detection through deep learning," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–13, 2020.

[7] J. Liu, H. Liu, Y. Chen, Y. Wang, and C. Wang, "Wireless sensing for human activity: A survey," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 1629–1645, 2019.

[8] K. Nishimori, Y. Koide, D. Kuwahara, N. Honmay, H. Yamada, and M. Hideo, "MIMO Sensor Evaluation on Antenna Arrangement," in *EUCAP 2011*. Rome, Italy: IEEE, Apr. 2011, pp. 2771–2775.

[9] 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, Nov. 2015.

[10] E. Ding, X. Li, T. Zhao, L. Zhang, and Y. Hu, "A Robust Passive Intrusion Detection System with Commodity WiFi Devices," *Journal of Sensors*, vol. 2018, pp. 1–12, Jun. 2018.

[11] H. Zou, Y. Zhou, J. Yang, W. Gu, L. Xie, and C. Spanos, "FreeDetector: Device-Free Occupancy Detection with Commodity WiFi," in *IEEE SECON 2017, SCIE Workshop*. San Diego, CA: IEEE, Jun. 2017.

[12] 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, Nov. 2015.

[13] E. Soltanaghaei, A. Kalyanaraman, and K. Whitehouse, "Peripheral WiFi Vision: Exploiting Multipath Reflections for More Sensitive Human Sensing," in *ACM MobiSys 2017, WPA Workshop*. Niagara Falls, NY: ACM, Jun. 2017.

[14] Z. Yang, Z. Zhou, and Y. Liu, "From RSSI to CSI: Indoor localization via channel response," *ACM Computing Surveys*, vol. 46, no. 2, pp. 1–32, Nov. 2013.

[15] Y. Xie, Z. Li, and M. Li, "Precise Power Delay Profiling with Commodity WiFi," in *ACM MobiCom 2015*. Paris, France: ACM, Sep. 2015, pp. 53–64.

[16] M. Atif, S. Muralidharan, H. Ko, and B. Yoo, "Wi-ESP - A tool for CSI-based Device-Free Wi-Fi Sensing (DFWS)," *Journal of Computational Design and Engineering*, vol. 7, no. 5, pp. 644–656, May 2020.

[17] D. Zhang, Y. Hu, Y. Chen, and B. Zeng, "Calibrating Phase Offsets for Commodity WiFi," *IEEE Systems Journal*, vol. 14, no. 1, pp. 661–664, Mar. 2020.

[18] A. Zubow, P. Gawłowicz, and F. Dressler, "On Phase Offsets of 802.11ac Commodity WiFi," in *IEEE/IFIP WONS 2021*. Virtual Conference: IEEE, Mar. 2021.

[19] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay, "Scikit-Learn: Machine Learning in Python," *The Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, Nov. 2011.

[20] B. Schölkopf, R. Williamson, A. Smola, J. Shawe-Taylor, and J. Platt, "Support Vector Method for Novelty Detection," in *12th International Conference on Neural Information Processing Systems (NIPS 1999)*. Denver, CO: MIT Press, Nov. 1999, pp. 582–588.