An Unsupervised Approach to Motion Detection Using WiFi Signals

Naveed Tahir[1], Yang Liu[2], Tiexing Wang[3], Garrett E. Katz[1], Biao Chen[1]

***Abstract*—WiFi signals have been demonstrated to facilitate non-intrusive detection of a range of activities and behaviors in the physical environments they permeate. Different activities affect both phase and magnitude of channel state information (CSI) in WiFi networks in a complex yet predictable way, and machine learning models can be trained to classify activities from such information. While constructing such WiFi-sensing systems is generally convenient and cost-effective, acquiring labeled data for a particular task can be time and labor-intensive. In this paper, we seek to remedy this issue in the context of human motion detection using deep unsupervised learning. Our proposed method uses a deep clustering model trained on appropriately-preprocessed CSI magnitude-only data to detect human motion with over 99% accuracy in the absence of any ground labels. Removing the need for labeled samples significantly reduces the training overhead, making it a promising alternative to existing methods for motion detection.**

***Index Terms*—RF sensing, WiFi sensing, channel state information, deep unsupervised learning, human activity recognition**

I. Introduction

WiFi-sensing re-purposes existing communication networks for detection, recognition, and estimation [1]. In doing so, it does away with the need for specialized embedded or vision sensors often relied upon in conventional sensing methods and instead uses ambient WiFi signals [2]. The advantages of a WiFi-sensing system are manifold. Not only are WiFi signals ubiquitous, but they also are privacy-preserving and more resilient than conventional sensors to changes in physical conditions such as temperature and lighting. Furthermore, the fine-grained nature of channel state information (CSI) in modern WiFi systems enables such systems to capture subtle changes in the physical environment. Together, these features facilitate low-cost, passive, accurate and real-time monitoring of various phenomena using WiFi sensing [3], [4].

Recent advances in WiFi sensing are generally rooted in data-driven approaches with neural networks or other machine learning models to learn specific patterns embedded in CSI variations resulting from a phenomenon under study [5], [6]. Such approaches require large amounts of quality data used to train neural networks. While modern WiFi systems generate large amounts of data, labeling that data for a deep supervised model is often a significantly time-consuming and error-prone task. For example, cameras or other sensing modalities are often required to assist data segmentation and labeling [7], [8]. Alternatively, continuous human motion has to be introduced for training data collection [9]–[11]. This presents a significant barrier to the deployment of any WiFi or RF (radio frequency) sensing systems for real-world applications.

A deep semi-supervised model could mitigate this labeling requirement but might be prone to overfitting in the presence of the noise that accompanies CSI samples or their corresponding labels. Given that CSI can register subtle changes in the environment, any labeling scheme will be affected by noisy labels. This effect is visible in human motion detection tasks where CSI data for still humans resembles that for human-free samples [12]. A motion classifier trained on such data usually has a high false-positive rate.

We propose a purely unsupervised deep clustering model to detect human motion from CSI magnitude-only data. This completely removes the need for any labeled samples, making it a promising alternative for motion detection in many practical applications. Using a compressive, digital signal processing-based pre-processing scheme, we process unlabeled motion-containing data collected in various human-free and human-present settings and use it to train a neural network model comprising an autoencoder and a clustering module. The model is then evaluated on held-out data with carefully curated motion and no-motion labels. We report both overall and class-wise accuracy, along with other evaluation metrics. To our knowledge, this is the first WiFi-sensing study on human motion detection with deep *unsupervised* learning. The code and data needed to reproduce our results are open-source and freely available.[1]

II. Related Work

Early work on radio frequency (RF) sensing relied on handcrafted features, such as those derived from the received signal strength (RSS) [13] measurement, or more fine-grained signatures extracted from CSI [14]. For example, RSS can be used to detect various events in the environment [15] including human gestures [16] and other activities [17]; whereas CSI has been used to detect and localize walking within an apartment [14] and even identify specific signs used in sign language [18]. Recent approaches combine preprocessed CSI input data with data-driven deep learning techniques to improve performance in various detection and classification tasks [12], [19]. However, the data-driven approach in RF

[1] Naveed Tahir, Garrett E. Katz and Biao Chen are with Dept. of Elec. Engr. and Comp. Sci. at Syracuse University, Syracuse, NY 13244. {ntahir, gkatz01, bichen}@syr.edu

[2] Yang Liu is with Micron Technology, San Jose, CA 95134. yliui@micron.com

[3] Tiexing Wang is with Samsung Research, Mountain View, CA 94043. tiexing.wang@samsung.com

[1] https://github.com/vanishinggrad/unsupmotiondetection

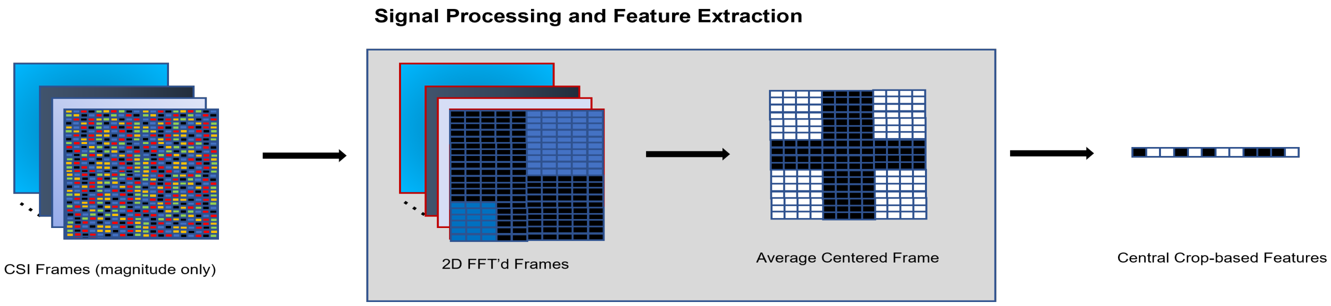
Signal Processing and Feature Extraction
CSI Frames (magnitude only)
2D FFT'd Frames
Average Centered Frame
Central Crop-based Features

Fig. 1. CSI feature extraction for motion detection. Consecutive CSI frames are stacked together and magnitude data is extracted from them. Next, 2D DFT is applied to the frequency (subcarrier) and temporal (CSI frame index) dimensions of the resulting array. The transformed array is then log-transformed and a rectangular crop is taken around zero frequency. Finally, the central crop is flattened into a feature vector.

sensing has been primarily limited to supervised learning, for which labeling can be expensive and error-prone. The result is highly noisy datasets in which some labels are missing or incorrect, leading to degradation in learning performance. Some recent works have employed a semi-supervised approach but the learning models still need to be fine-tuned extensively with labeled samples to exhibit competitive performance on activity recognition tasks [20], [21]. It is arguably desirable to explore learning paradigms and models that are either robust to noisy labels or require only unlabeled training samples.

Outside of RF sensing, there is a vast body of machine learning research on learning with missing labels (semi-supervised learning), e.g., [22]–[25], as well as noisy labels (e.g., [26], [27]). These methods reduce the reliance on completely or cleanly labeled data but still assume that a portion of labels are available and correct. In contrast, purely unsupervised methods can uncover regularities in completely unlabeled data and have the potential to identify distinct clusters that correspond directly to distinct classes. Recently, deep learning has been incorporated into more traditional unsupervised learning techniques such as K-Means clustering [28] and Gaussian Mixture Models [29]. For example, deep autoencoders trained with reconstruction loss do not require labels and can produce non-linearly transformed latent representations more amenable to clustering than the raw input data. However, deep unsupervised methods have mainly been tested using synthetic or carefully curated and cleaned datasets such as MNIST [30].

Our contribution is to adapt deep unsupervised learning approaches to the RF sensing domain, where comparatively sparse and noisy data require specialized preprocessing to enable effective deep clustering. To our knowledge, ours is the first approach capable of purely unsupervised motion detection using RF data.

III. Signal Pre-processing and Feature Extraction

CSI obtained from a WiFi system must be carefully pre-processed with due consideration for expected variations in its spatial, temporal, and frequency dimensions before being input into a learning model. We consider a WiFi system using multiple-input–multiple-output orthogonal frequency-division multiplexing (MIMO-OFDM) with N_t transmit antennas, N_r receiver antennas, and N_{sc} subcarriers. The CSI for the i-th dataframe at the receiver is a complex-valued 3D array given by $\mathbf{H}[i] \in \mathbb{C}^{N_{sc}\times N_r \times N_t}$. Consecutive CSI frames are needed to capture temporal variation in the propagation of wireless signals due to the movement of humans or other objects in the environment. If a motion is to be detected using I consecutive frames, a 4D array $\mathbf{H} \in \mathbb{C}^{I\times N_{sc}\times N_r\times N_t}$ is obtained by stacking consecutive frames along the temporal dimension. Given that typical carrier spacing in WiFi signals (312.5 kHz) is much smaller than the coherent bandwidth of typical indoor environments, we can reduce data dimensions by downsampling subcarriers with little to no effects on motion detection performance. Down-selecting N_f subcarriers in this way reduces data dimensions by a factor of $\frac{N_{sc}}{N_f}$, with the new array being $\mathbf{H} \in \mathbb{C}^{I\times N_f\times N_r\times N_t}$. Both CSI phase and magnitude information can then be extracted from $\mathbf{H}$.

Our feature extraction scheme (Fig. 1) preprocesses CSI magnitude data[2] to reduce its dependence on environment-specific effects. Suppose $\mathbf{X}^{\text{abs}}$ is the array of CSI magnitude-only data with the spatial (antenna) dimensions combined as $I \times N_f \times (N_r N_t)$. We normalize it to remove its dependence on absolute power level as follows:

$$\tilde{\mathbf{X}}^{\text{abs}}_{i,:,:} = \mathbf{X}^{\text{abs}}_{i,:,:}/\mathbf{X}^{\text{abs}}_{0,:,:}, \tag{1}$$

Next, we apply 2D Discrete Fourier Transform (DFT) to the first two dimensions of $\mathbf{X}^{\text{abs}}$.

$$\tilde{\mathbf{X}}^{\text{abs}-\text{fft}}_{:,:,j} = \mathcal{F}\left(\tilde{\mathbf{X}}^{\text{abs}}_{:,:,j}\right) \tag{2}$$

Finally, we take a 2D crop of size $T \times D$ from each $\tilde{\mathbf{X}}^{\text{abs-ftt}}_{:,:,j}$ centered around zero frequency, with $T < I$ and $D < N_f$:

$$\tilde{\mathbf{X}}^{\text{abs-fft-crop}}_{i,k,:} = \tilde{\mathbf{X}}^{\text{abs-fft}}_{i+t,k+d,:} \tag{3}$$

where $i = 0, \ldots, T-1$, $k = 0, \ldots, D-1$, and t and d are the index offsets to the corner of the crop.

[2] It is conceivable that the performance can be further improved if CSI phase is incorporated. However, the performance of the trained model for motion detection is near perfect, suggesting the benefit of including CSI phase for this particular task is marginal at best.

As DFT can result in a high dynamic range, an element-wise logarithmic transform is further applied:

$$\tilde{\mathbf{X}}^{\text{abs-fft-crop-log}}_{i,k,j} = \log_{10}(\tilde{\mathbf{X}}^{\text{abs-fft-crop}}_{i,k,j} + 1) \tag{4}$$

We then aggregate over all transceiver pairs:

$$\tilde{\mathbf{X}}^{\text{agg}}_{i,k} = \frac{1}{N_t N_r} \sum_{j} \tilde{\mathbf{X}}^{\text{abs-fft-crop-log}}_{i,k,j} \tag{5}$$

To assess the generality of our method, two different datasets (discussed further below) from different environments were used for training and assessing the proposed learning model and with slightly different pre-processing: Specifically, one dataset used median rather than average for this aggregation step. Finally, $\tilde{\mathbf{X}}^{\text{agg}}$ is flattened and input to the learning model.

IV. Detection Model

We employ deep unsupervised learning for motion detection from CSI magnitude data. Deep unsupervised learning is a machine learning paradigm that extends backpropagation-based learning methods to problems where ground labels are not available for supervised learning. It is also a promising workaround for the problem of noisy labels. For instance, using a carefully designed training loss, a neural network can be forced to learn the underlying structure of inputs that is useful for separating one sample class from another. This is in contrast with supervised learning where the learning of input-output mapping is guided using class labels or true outputs.

For this paper, we sought to understand whether motion-containing samples were distinct from all other types of samples in the data and if an unsupervised neural network model could be trained to uncover the discriminating features of motion samples from other data. While unsupervised neural architectures such as autoencoders can learn powerful non-linear low-dimensional mappings, the optimization objectives generally do not explicitly focus on clustering [31]. As a result, representations learned by a neural network are not guaranteed to be clustering-friendly and may not reveal grouping structures underlying data. Motivated by these issues, many recent works on unsupervised learning explore joint dimensionality reduction and clustering.

Our motion detection model was a deep clustering network (DCN) comprising an autoencoder with an embedded clustering module based on [28] (Fig. 2). The autoencoder has a feedforward "encoder" block with multiple hidden layers that compute low-dimensional encodings of input data. A mirror "decoder" block then reconstructs input data from these encodings. The encoder block is further attached to a K-Means clustering block which calculates cluster assignments for encoded samples. Learnable weights for the three blocks are optimized jointly using a composite of reconstruction loss and clustering loss.

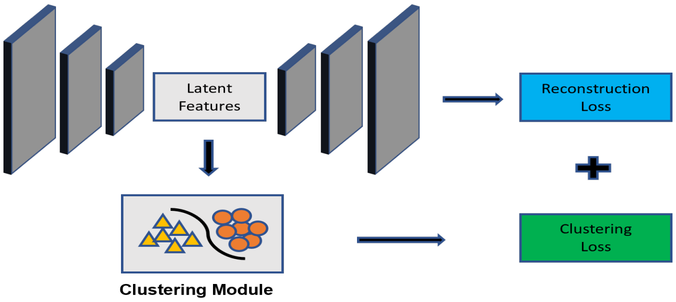

Fig. 2. Deep clustering model used for CSI-based motion detection. A feedforward autoencoder neural network is embedded with a clustering module that outputs clusters or classes for given samples. The network is trained on feature vectors extracted from CSI magnitude data with a joint clustering and reconstruction loss.

Formally, training loss for the network was calculated as follows:

$$\begin{aligned} &\min_{\mathcal{W},\mathcal{Z},\boldsymbol{M},\{s_i\}} \sum_{i=1}^{N} \left(\ell\left(\boldsymbol{g}\left(\boldsymbol{f}\left(\boldsymbol{x}_i\right)\right), \boldsymbol{x}_i\right) + \frac{\lambda}{2} \left\|\boldsymbol{f}\left(\boldsymbol{x}_i\right) - \boldsymbol{M}\boldsymbol{s}_i\right\|_2^2 \right) \\ &\quad \text{s.t.} \quad s_{j,i} \in \{0,1\}, \mathbf{1}^T s_i = 1 \quad \forall i, j \end{aligned} \tag{6}$$

where $\boldsymbol{f}(\boldsymbol{x}_i)$ and $\boldsymbol{g}(\boldsymbol{h}_i)$ are simplified equivalents for $f(\boldsymbol{x}_i; \mathcal{W})$, the encoder representation, and $\boldsymbol{g}(\boldsymbol{h}_i; \mathcal{Z})$, the decoder reconstruction, respectively (with $\mathcal{W}$ and $\mathcal{Z}$ being encoder and decoder weights respectively). Similarly, $\boldsymbol{M}$ is the centroid matrix with cluster centroids as its columns, $\boldsymbol{s}_i$ is the (cluster) assignment vector for $\boldsymbol{x}_i$ with only one $\boldsymbol{s}_{j,i} = 1$, and ℓ is taken to be the squared-error loss. The parameter λ controls the weight of the clustering objective relative to the reconstruction objective and is determined empirically.

Similar to the alternating construction of the K-Means algorithm, the network was trained using an alternating variation of stochastic gradient descent (SGD) where $(\mathcal{W}, \mathcal{Z})$, $\boldsymbol{M}$, and $\boldsymbol{s}_i$ were updated alternately while keeping others constant. To be more precise, for a constant $\boldsymbol{M}$ and $\boldsymbol{s}_i$, the training loss given in Eq. (6) for input $\boldsymbol{x}_i$ was decomposed as:

$$\min_{\mathcal{W},\mathcal{Z}} \ell\left(\boldsymbol{g}\left(\boldsymbol{f}\left(\boldsymbol{x}_i\right)\right), \boldsymbol{x}_i\right) + \frac{\lambda}{2} \left\|\boldsymbol{f}\left(\boldsymbol{x}_i\right) - \boldsymbol{M}\boldsymbol{s}_i\right\|_2^2 \tag{7}$$

which was optimized for autoencoder parameters $(\mathcal{W}, \mathcal{Z})$ using gradient descent methods. Similarly, for fixed centroids and autoencoder parameters, the assignment vector $\boldsymbol{s}_i$ was updated in an online manner as follows:

$$s_{j,i} \leftarrow \begin{cases} 1, & \text{if } j = \underset{k=\{1,\ldots,K\}}{\arg\min} \left\|\boldsymbol{f}\left(\boldsymbol{x}_i\right) - \boldsymbol{m}_k\right\|_2 \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

Finally, each centroid in $\boldsymbol{M}$ was updated by using a gradient descent-like rule as follows:

$$\boldsymbol{m}_k \leftarrow \boldsymbol{m}_k - \left(\frac{1}{c_k^i}\right) \left(\boldsymbol{m}_k - \boldsymbol{f}\left(\boldsymbol{x}_i\right)\right) s_{k,i} \tag{9}$$

where c_k^i counts the number of samples assigned to cluster k before handling the i-th input.

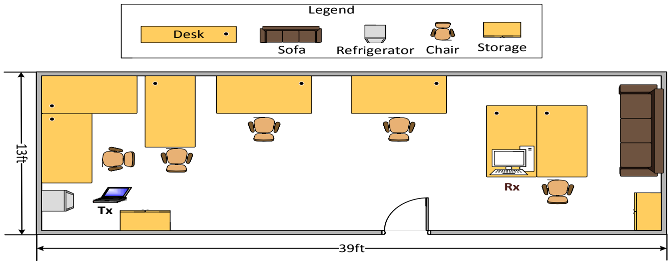

Fig. 3. Floor plan of the lab where data was collected.

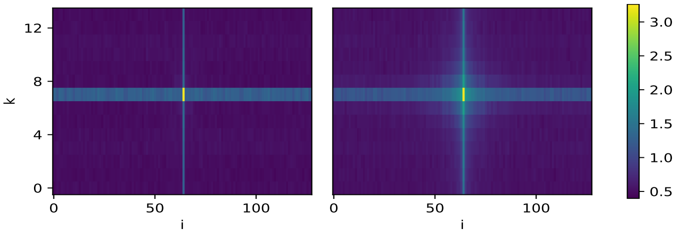

Fig. 4. Average Fourier-transformed CSI frame for Human-free (left) and Human-presence (right) data from the lab dataset. The sparsity of the plots indicates that a central crop might be useful for further dimensionality reduction.

Unsupervised models such as the one above can be evaluated using various intrinsic and extrinsic metrics. While normalized mutual information (NMI) [32] and adjusted Rand index (ARI) [33] are common choices, a more direct comparison with a supervised learning method can be facilitated using clustering accuracy (ACC) [32].

V. Experiments

For experiments, we used two datasets collected over a period of 18 months. The first data was collected in a typical lab inside a university building. Motions, when present, come exclusively from occupant(s) in the lab. The second dataset was collected in a residential house. In addition to residents, their pet dogs also contributed to the motion data.

A. Setup

The WiFi data collection system consisted of a laptop (Thinkpad T410) as the transmitter and a desktop (Dell OptiPlex 7010) as the receiver. Both had Atheros 802.11n WiFi chipset AR9580 installed, and Atheros-CSI-Tool [34] was used to extract CSI frames at the receiver. The WiFi system had a 3×3 MIMO architecture and was operating in a 20MHz channel at channel 6 in the 2.45GHz band. With $N_t = 3$ transmit antennas, $N_r = 3$ receiver antennas, and $N_{sc} = 56$ subcarriers, a single CSI frame was a complex-valued array of size $\mathbf{H}[i] \in \mathbb{C}^{56 \times 3 \times 3}$. Down-sampling the number of subcarriers to 14 evenly-spaced subcarriers resulted in smaller dimensions of $14 \times 3 \times 3$ instead. The duration of each WiFi packet is roughly 10ms, which is also the time interval between two CSI measurements. For motion detection, a key feature is the temporal variation of the CSI induced by the movement of humans/objects. For this reason, we stacked 128 consecutive CSI frames together which resulted in a detection window of size $\approx$ 1.28s. A typical CSI sample in our experiments was therefore given by the array $\mathbf{H} \in \mathbb{C}^{128 \times 14 \times 3 \times 3}$.

B. Data Collection

For the lab dataset, CSI data were collected on 10 days spaced over a period of 1.5 months in a lab for engineering graduate students. Training data were collected on days 9 and 10 and were broadly classified as "human-free" or "human-presence" data. No one was present inside the lab when data was collected in a "human-free" session. On the other hand, there was at least one person present in the lab and going about their usual activities when data was collected for a "human-presence" session. The lab layout is visualized in Fig. 3. The arrangement of workstations and other objects shown is illustrative only and varied over time. Human-free data was collected for about 40 minutes each day whereas human-free data was recorded for about 1 hour each day. Together, the two days yielded $N^{(0)} = 5000$ "no-motion" and $N^{(1)} = 10000$ "motion" samples for training on this data. It is worth noting that with "human-presence" data, a majority of those CSI samples contain little or no movements as the occupant(s) were largely still. Those CSI samples resemble that of a "human-free" environment and, if used directly with a supervised learning approach, can lead to an excessive false-positive rate.

The rest of the data, from days 1 through 8, were used to evaluate the motion detection model. These data include both "human-free" data when the lab was completely empty and "human-motion" data when occupant(s) engaged in deliberate and continuous motion throughout the entire measurement period. Those data provide correctly labeled samples used only for evaluating the performance of the unsupervised learning model trained on data from days 9 and 10, and were not involved in the training process itself.

Within the "human-motion" test data, on days 4-8, subjects were asked to perform large-scale motions including walking, sitting down, and standing up; whereas on days 1-3, only small-scale motion (e.g., turning in chairs, arm waving, etc.) was present. Each of these days yielded about 4500 samples for each label from 35-minute data collection sessions. Fig. 4 visualizes 2D DFT for two random CSI frames sampled from both human-free and human-presence data.

For the house dataset, CSI data were collected for 7 different days over a period of 2 weeks in a typical four-bedroom colonial-style house accommodating two human participants and their three pet dogs. The WiFi transceiver pair were both placed on the first floor, one in the kitchen and the other in the living room, with no line-of-sight between the transmitter and the receiver. Given the mercurial nature of the pets, data collection during the 19 sessions lasted from 5 to 30 minutes and varied in the count of samples obtained. Since we focused only on motion detection, we combined human motion and pet/mixed motion samples under the same label. Doing so, we obtained $N^{(0)} = 1746$ and $N^{(1)} = 873$ samples from a training set spanning 4 days. Similarly, the other 3 days generated a test set of 10869 samples with a 45:55% split between motion and no-motion samples.

C. Model Training

For unsupervised motion detection from CSI data, both datasets were pre-processed using the scheme described in Section III, and 1D feature vectors were obtained, except that median instead of averaging was used for the lab data in the aggregation step (Eq. (5)). Next, we trained an instance of the deep clustering network (DCN) described in Section IV on each dataset. Since we were interested in finding a bi-clustering that separated motion samples from no-motion samples, the number of clusters was set to be 2 for the K-Means clustering block. The encoder network was a feed-forward network comprising three hidden layers with 100, 500, and 10 units respectively. Layer sizes were the same for both datasets and were determined to be performing well empirically. The decoder network was a “mirror” version of the encoder and had the same layers but in reverse order. Each hidden layer consisted of a learnable linear transformation followed by rectified linear unit (ReLU) activation.

The network was trained to minimize the composite loss of Eq. (6) using the Adam optimizer [35] with default hyper-parameters and a batch size of 30. Before training the DCN with this loss, it was pre-trained for 100 epochs using a reconstruction-only loss. For training with joint loss, the number of epochs was set to be 50, and clustering loss weight to $\lambda = 0.1$. These hyper-parameters were chosen by hand and found to work well empirically, although performance could potentially be further improved with more systematic hyper-parameter search and cross-validation. Finally, the model was evaluated on test data, and the performance was reported for multiple metrics, including overall accuracy (ACC), accuracy on class 1 motion samples (ACC^1), accuracy on class 0 no-motion samples (ACC^0), normalized mutual information (NMI), and adjusted Rand index (ARI).

D. Results

For the lab dataset, our model detected motion with very high accuracy even with its unsupervised design. When trained on data from days 9 and 10, the accuracy of the model on held-out data from days 1-8 was 99.09% when averaged over 10 trials where the model weights were randomly initialized. Class-wise performance was similarly good with the average accuracy for motion and no-motion classes being 98.60% and 99.53% respectively. For the house dataset, we trained a separate model with the same architecture for which the overall accuracy was 98.84%, also averaged over 10 random model initializations. Table I reports performance on both datasets across all metrics.

Per-day evaluation for the lab dataset is visualized in Fig. 5. Notice that we did not make any assumptions on the relative distribution of motion samples within the dataset (or within human-presence data). Such assumptions, while helpful in tuning clustering algorithms in general, were not needed for the model to detect motion on all test days with high accuracy. Moreover, the model performs well for instances of both small and large-scale motions. As discussed in Section V-B, days 1-3 and days 4-8 differ in the range of motion induced in their samples. The model shows only slightly lower accuracy for days 1-3, which is intuitive if we consider that a small number of samples with small-scale, subtle motion (e.g., turning in chairs, arm waving, etc.) may not get classified correctly by a completely unsupervised model. The model also showed good robustness to environmental changes in that the performance on different test days was consistently high even though some days were weeks apart when data was collected.

We also conducted a small cross-data experiment to explore whether the model could generalize and detect motion in an environment different from the one for which it was trained. In other words, we were interested in understanding if learning could be transferred between two environments with different motion characteristics. To test for such transferability, we trained an instance of the proposed motion detection model on the house dataset and then evaluated it using the lab dataset. The model could still generalize very well to this new data with 87.54% overall accuracy and 99.95% motion class accuracy, indicating that it had largely retained its motion-detecting properties. By re-training the (unsupervised) model on the lab dataset for 10 epochs, we could boost class-wise accuracy for the no-motion class from 75.14% to 97.70%, with

TABLE I
RESULTS (AVERAGE ± STANDARD DEVIATION) FOR MODEL EVALUATION ON BOTH DATASETS, AVERAGED OVER 10 RANDOMLY INITIALIZED RUNS. ALL METRICS ARE ON A 0-100% SCALE.

	Lab	**House**
ACC	99.09 ± 0.37	98.84 ± 0.07
ACC^1	98.60 ± 0.77	98.96 ± 0.25
ACC^0	99.53 ± 0.23	98.74 ± 0.09
NMI	92.81 ± 2.27	90.82 ± 0.49
ARI	96.38 ± 1.44	95.40 ± 0.27

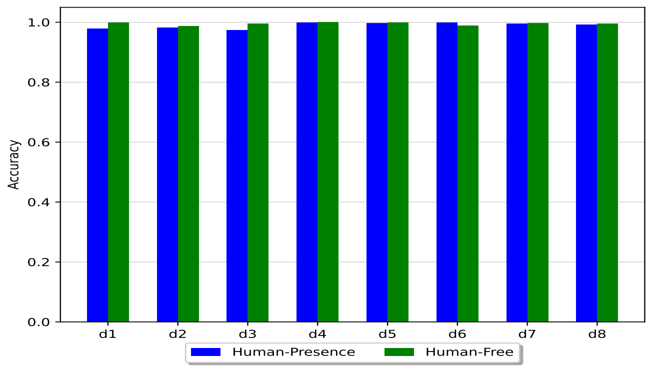

Fig. 5. Class-wise accuracy for the lab dataset on test days. Human-motion data from these test days contains deliberately-induced small (days 1-3) and large-scale (days 4-8) motion.

the overall accuracy also improving to 98.70%. The results have been visualized in Fig. 6.

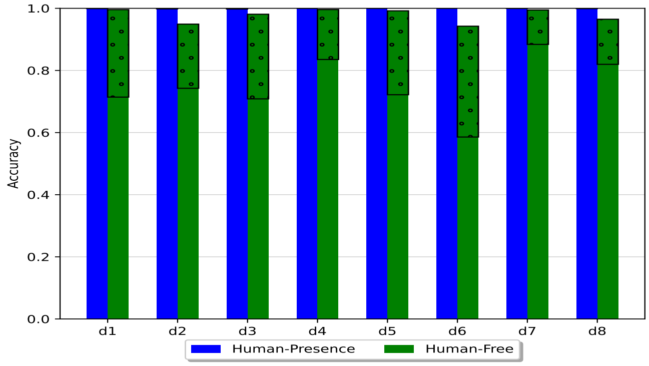

Fig. 6. Transfer learning for motion detection across environments. Dotted stacked bars in the no-motion class correspond to the accuracy level restored by fine-tuning the transferred model trained on house data with re-training on unseen lab data for 10 epochs.

As baselines for comparison, we also tested standard K-Means clustering (without deep autoencoding) and Gaussian Mixture Models (GMMs), as implemented in Scikit-learn [36]. We combined GMMs with random projection (RP) into lower-dimensional sub-spaces, following [37], which can improve performance when clusters have high eccentricity.

TABLE II
TEST ACCURACY FOR BASELINE METHODS, IN FORMAT: AVERAGE ± STANDARD DEVIATION (BASE-10 LOGARITHM OF P-VALUE).

	Lab	**House**
K-Means	$95.56 \pm 0.04\,(-12)$	$97.44 \pm 0.00\,(-12)$
GMM	$94.45 \pm 0.03\,(-13)$	$98.34 \pm 0.00\,(-7)$
GMM+RP	$94.48 \pm 3.34\,(-4)$	$97.94 \pm 2.78\,(-3)$

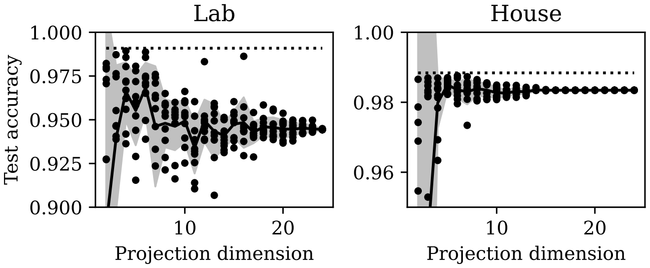

Fig. 7. Test performance of Gaussian mixture models after random projection of the feature vectors, for varying projection dimension. The average (solid black line) and standard deviation (gray envelope) are aggregated over 30 independent repetitions (black dots). For reference, the horizontal dashed black line shows average test accuracy for DCN.

Compared to the DCN, the baselines also performed quite well, indicating the suitability of the proposed feature extraction scheme, but there was a statistically significant reduction in test accuracy ("**ACC**"), as shown in Table II (cf. Table I, first row). We measured statistical significance with Welch's t-test as implemented in SciPy [38], with the null hypothesis that a given baseline has the same average accuracy as the DCN on a respective dataset, and report the rounded, base-10 logarithms of the resulting p-values. The statistics in this table are aggregated over 30 independent repetitions of each baseline. For the random-projection baselines, we tested every possible sub-space dimension ranging from 2 to 23 (one less than the preprocessed feature vector dimension), resulting in the data shown in Fig. 7. The statistics in Table II (bottom row) are only for the projection dimensions where test accuracy was highest (6 for lab and 5 for house).

VI. Conclusion

We have presented a machine-learning model for motion detection using WiFi CSI data. Even without comprehensive hyper-parameter tuning, the performance of our model is comparable to existing approaches to this problem. However, unlike previous approaches, our method is completely unsupervised. This work represents a major departure from existing approaches in WiFi sensing reported in the literature that invariably rely on the availability of labeled samples in the training process. By removing the need for expensive and labor-intensive labeling of training data, the proposed learning model constitutes a significant step towards effective WiFi sensing deployable in the real world.

The work involves two separate datasets for validation. One important open research question is whether a model trained on one environment (e.g., the house data) can transfer to another environment (e.g., the lab data). Our initial experiments suggest this is possible, although our current model incurs a modest accuracy reduction in direct transfer without any additional fine-tuning.

Another important open question is to understand the potential and limitations of an unsupervised learning approach for WiFi sensing. By nesting the K-Means clustering with an autoencoder, the learning model is capable of extracting latent features that are useful to separate motion samples from static samples. Extending the current approach to classify different motion types can make such an unsupervised approach even more appealing. For example, there is a clear incentive to distinguish motion induced by humans and pets, or detect human falls within the human motion data. This is conceivably straightforward in a supervised learning framework involving carefully curated samples. Whether an unsupervised model such as the one used in this paper can extract subtle differences in WiFi CSI remains an open question.

Acknowledgment

This work was supported in part by the National Science Foundation under grant ECCS-2036236 and a gift fund from Syracuse Research Corporation.

References

[1] Y. Ma, G. Zhou, and S. Wang, "Wifi sensing with channel state information: A survey," *ACM Computing Surveys (CSUR)*, vol. 52, no. 3, pp. 1–36, 2019.

[2] L. M. Dang, K. Min, H. Wang, M. J. Piran, C. H. Lee, and H. Moon, "Sensor-based and vision-based human activity recognition: A comprehensive survey," *Pattern Recognition*, vol. 108, p. 107561, 2020.

[3] S. Tan, Y. Ren, J. Yang, and Y. Chen, "Commodity wifi sensing in 10 years: Status, challenges, and opportunities," *IEEE Internet of Things Journal*, 2022.

[4] H. Jiang, C. Cai, X. Ma, Y. Yang, and J. Liu, "Smart home based on wifi sensing: A survey," *IEEE Access*, vol. 6, pp. 13 317–13 325, 2018.

[5] F. Gu, M.-H. Chung, M. Chignell, S. Valaee, B. Zhou, and X. Liu, "A survey on deep learning for human activity recognition," *ACM Computing Surveys (CSUR)*, vol. 54, no. 8, pp. 1–34, 2021.

[6] Z. Wang, Z. Huang, C. Zhang, W. Dou, Y. Guo, and D. Chen, "Csi-based human sensing using model-based approaches: a survey," *Journal of Computational Design and Engineering*, vol. 8, no. 2, pp. 510–523, 2021.

[7] H. Huang and S. Lin, "Widet: Wi-fi based device-free passive person detection with deep convolutional neural networks," *Computer Communications*, vol. 150, pp. 357–366, 2020.

[8] F. Wang, S. Zhou, S. Panev, J. Han, and D. Huang, "Person-in-wifi: Fine-grained person perception using wifi," in *Proc. IEEE Int. Conf. Comput. Vision*, Seoul, Korea, Nov. 2019, pp. 5452–5461.

[9] H. Zhu, F. Xiao, L. Sun, R. Wang, and P. Yang, "R-ttwd: Robust device-free through-the-wall detection of moving human with wifi," *IEEE J. Select. Areas in Commun.*, vol. 35, no. 5, pp. 1090–1103, 2017.

[10] K. Qian, C. Wu, Z. Yang, Y. Liu, F. He, and T. Xing, "Enabling contactless detection of moving humans with dynamic speeds using csi," *ACM Trans. Embedded Computing Syst. (TECS)*, vol. 17, no. 2, pp. 1–18, 2018.

[11] Y. Liu, T. Wang, Y. Jiang, and B. Chen, "Harvesting ambient rf for presence detection through deep learning," *IEEE Trans. Neural Networks and Learning Systems*, vol. 33, no. 4, pp. 1571–1583, 2022.

[12] ——, "Harvesting ambient rf for presence detection through deep learning," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[13] A. E. Kosba, A. Saeed, and M. Youssef, "Rasid: A robust wlan device-free passive motion detection system," in *2012 IEEE International Conference on Pervasive Computing and Communications*. IEEE, 2012, pp. 180–189.

[14] 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*, 2014, pp. 617–628.

[15] M. Moussa and M. Youssef, "Smart cevices for smart environments: Device-free passive detection in real environments," in *2009 IEEE International Conference on Pervasive Computing and Communications*. IEEE, 2009, pp. 1–6.

[16] H. Abdelnasser, M. Youssef, and K. A. Harras, "Wigest: A ubiquitous wifi-based gesture recognition system," in *2015 IEEE conference on computer communications (INFOCOM)*. IEEE, 2015, pp. 1472–1480.

[17] Y. Gu, F. Ren, and J. Li, "Paws: Passive human activity recognition based on wifi ambient signals," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 796–805, 2015.

[18] 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, 2018.

[19] S.-H. Fang, C.-C. Li, W.-C. Lu, Z. Xu, and Y.-R. Chien, "Enhanced device-free human detection: Efficient learning from phase and amplitude of channel state information," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 3, pp. 3048–3051, 2019.

[20] J. Yang, X. Chen, H. Zou, D. Wang, and L. Xie, "Autofi: Toward automatic wi-fi human sensing via geometric self-supervised learning," *IEEE Internet of Things Journal*, vol. 10, no. 8, pp. 7416–7425, 2022.

[21] J. Yang, X. Chen, H. Zou, C. X. Lu, D. Wang, S. Sun, and L. Xie, "Sensefi: A library and benchmark on deep-learning-empowered wifi human sensing," *Patterns*, vol. 4, no. 3, 2023.

[22] J. E. Van Engelen and H. H. Hoos, "A survey on semi-supervised learning," *Machine Learning*, vol. 109, no. 2, pp. 373–440, 2020.

[23] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. A. Raffel, "Mixmatch: A holistic approach to semi-supervised learning," *Advances in neural information processing systems*, vol. 32, 2019.

[24] F. Zheng, N. Chen, and L. Li, "Semi-supervised laplacian eigenmaps for dimensionality reduction," in *2008 International Conference on Wavelet Analysis and Pattern Recognition*, vol. 2. IEEE, 2008, pp. 843–849.

[25] M.-F. Balcan and A. Blum, "A pac-style model for learning from labeled and unlabeled data," in *International Conference on Computational Learning Theory*. Springer, 2005, pp. 111–126.

[26] N. Natarajan, I. S. Dhillon, P. K. Ravikumar, and A. Tewari, "Learning with noisy labels," *Advances in neural information processing systems*, vol. 26, 2013.

[27] H. Song, M. Kim, D. Park, Y. Shin, and J.-G. Lee, "Learning from noisy labels with deep neural networks: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[28] B. Yang, X. Fu, N. D. Sidiropoulos, and M. Hong, "Towards k-means-friendly spaces: Simultaneous deep learning and clustering," in *international conference on machine learning*. PMLR, 2017, pp. 3861–3870.

[29] J. Wang and J. Jiang, "Unsupervised deep clustering via adaptive gmm modeling and optimization," *Neurocomputing*, vol. 433, pp. 199–211, 2021.

[30] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[31] P. Baldi, "Autoencoders, unsupervised learning, and deep architectures," in *Proceedings of ICML workshop on unsupervised and transfer learning*. JMLR Workshop and Conference Proceedings, 2012, pp. 37–49.

[32] D. Cai, X. He, and J. Han, "Locally consistent concept factorization for document clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 6, pp. 902–913, 2010.

[33] K. Y. Yeung and W. L. Ruzzo, "Details of the adjusted rand index and clustering algorithms, supplement to the paper an empirical study on principal component analysis for clustering gene expression data," *Bioinformatics*, vol. 17, no. 9, pp. 763–774, 2001.

[34] Y. Xie, Z. Li, and M. Li, "Precise power delay profiling with commodity wifi," in *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '15. New York, NY, USA: ACM, 2015, p. 53–64. [Online]. Available: http://doi.acm.org/10.1145/2789168.2790124

[35] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR (Poster)*, 2015.

[36] 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 E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[37] S. DASGUPTA, "Experiments with random projection," in *Uncertainty in Artificial Intelligence*, 2000, pp. 143–151.

[38] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020.