mmLock: User Leaving Detection against Data Theft via High-Quality mmWave Radar Imaging

Jiawei Xu*§, Ziqian Bi*§, Amit Singha*, Tao Li*, Yimin Chen†, Yanchao Zhang‡
* Indiana University-Purdue University Indianapolis, † University of Massachusetts Lowell , ‡ Arizona State University
{jxu4, bizi, singamit}@iu.edu, tli6@iupui.edu, ian_chen@uml.edu, yczhang@asu.edu
§ Co-first Authors

2023 32nd International Conference on Computer Communications and Networks (ICCCN) | 979-8-3503-3618-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICCCN58024.2023.10230151

***Abstract*—The use of smart devices such as smartphones, tablets, and laptops skyrocketed in the last decade. These devices enable ubiquitous applications for entertainment, communication, productivity, and healthcare but also introduce big concern about user privacy and data security. In addition to various authentication techniques, automatic and immediate device locking based on user leaving detection is an indispensable way to secure the devices. Current user leaving detection techniques mainly rely on acoustic ranging and do not work well in environments with multiple moving objects. In this paper, we present *mmLock*, a system that enables faster and more accurate user leaving detection in dynamic environments. mmLock uses a mmWave FMCW radar to capture the user's 3D mesh and detects the leaving gesture from the 3D human mesh data with a hybrid PointNet-LSTM model. Based on explainable user point clouds, mmLock is more robust than existing gesture recognition systems which can only identify the raw signal patterns. We implement and evaluate mmLock with a commercial off-the-shelf (COTS) TI mmWave radar in multiple environments and scenarios. We train the PointNet-LSTM model out of over 1 TB mmWave signal data and achieve 100% true-positive rate in most scenarios.**

I. Introduction

Smart devices, such as smartphones, tablets, and laptops, flooded into the market in the past decade. The number of mobile devices operating worldwide is expected to reach 18.22 billion in 2025, twice over the world population [1]. Smart devices greatly improved our life quality but also introduced big concern regarding user privacy and data security. According to Asurion's report, 8.7 million smartphones were lost or stolen in 2021—that's more than 24,000 phones each day [2]. The 2021 Data Breach Report from Verizon shows that personal data of 80% of the victims and bank data of 7% of the victims were compromised because of device loss and theft [3]. Therefore, it is critical to secure smart devices and prevent illegal access to the data and system operations therein.

The most common defense against device losses/thefts and the related data theft is to set a password on the smart device. However, the time window for a password-protected device going from the unlocked mode to the locked mode may be long enough for a capable attacker to access all the sensitive information on the lost/stolen device. For example, the auto-lock options on iOS 15 include time windows from 30 s to NEVER. Many users choose a longer time period or even NEVER for convenience. Other one-time authentication techniques based on physical or behavior biometric information such as fingerprints and faces have similar issues. The second method is to authenticate users continuously when they are using the device based on passively collected sensor data from the device [4], [5]. However, continuous authentication can only detect the attackers after they have used the device for a while, providing the attackers opportunities to access private data.

The third mechanism is more appealing, i.e., to lock the device immediately once the user has left. Existing systems [6], [7] in this category rely on acoustic ranging techniques to detect the user leaving gesture. However, these techniques require the line-of-sight (LOS) channel between the user and device, which may not be available all the time because the LOS channel could be easily blocked by surrounding objects. In addition, it has been shown that such systems cannot differentiate the correct leaving user from other moving objects simply due to the limited number of on-device microphones.

New-generation networking systems, such as 5G/6G cellular and future Wi-Fi networks, are envisioned to connect billions of heterogeneous smart devices and enable high-speed and low-latency communications using millimeter wave (mmWave) technologies. Qualcomm already has mmWave communication modules on their Snapdragon 5G chipset [8] and 802.11ad Wi-Fi chipset [9]. With the frequency of up to 100 GHz, the wavelength of mmWave signals can be as short as 3 mm about $\frac{1}{20}$ of the traditional 5 GHz Wi-Fi wavelength. The high frequency and short wavelength bring extraordinary sensing resolution for promising applications on autonomous driving [10], non-contact health monitoring [11], [12], material detection [13], vibration sensing [14], etc. However, the research on security applications of mmWave sensing is still very much in its infancy.

In this paper, we propose to detect the user leaving gesture by combining the mmWave and MIMO techniques. We aim to identify the leaving gesture immediately at the beginning of the leaving process to reduce any data leak risk. Meanwhile, we want to apply our method to more scenarios even when LOS channels might be blocked during the leaving process. A naive idea is to follow existing gesture recognition techniques and extract the user leaving gesture/pattern by analyzing RSSI [15]–[17] or CSI [18]–[20] data. However, pattern extraction in such techniques strongly depends on subject (i.e., the user) orientation therefore posing high training overhead to users.

979-8-3503-3618-4/23/$31.00 ©2023 IEEE

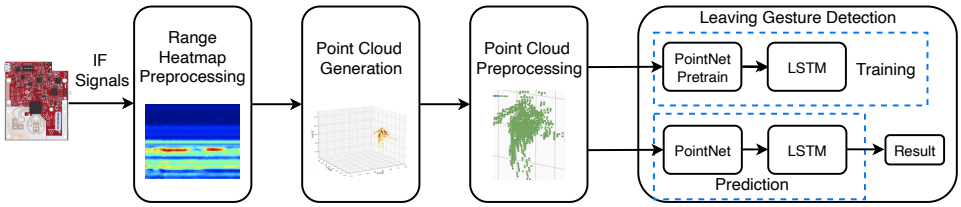


Fig. 1: System overview of mmLock. IF: Intermediate Frequency.

Similarly, models trained from such techniques are highly user- or scenario-dependent thus are unfriendly to new users and domains. Not to mention that the models lack explainability hindering further development and improvement.

To conquer the above challenges, we design *mmLock* (illustrated in Figure 1) which uses a mmWave radar to capture the user leaving gesture. One key novelty is that we explore new radar imaging techniques to generate high-quality 3D mesh of the leaving user, which is much more intuitive and explainable than the signal patterns from other sources (e.g., RSSI and CSI). Specifically, the radar in mmLock senses the user activity using the frequency-modulated continuous wave (FMCW) technique which achieves high range resolution and has no "blind spot" problem at short range. Our radar emits FMCW chirps and extracts range, velocity, and angle information of the points on the user body which reflect received signals. Utilizing the above information, mmLock is able to construct point clouds in the 3D space where our target user can be clearly represented by the human meshes.

After building the 3D human meshes for a user, the next step for mmLock is to detect the user leaving gesture from the time-series point clouds. mmLock first determines the pose in each point cloud and detects a potential leaving gesture by looking at all poses extracted from point clouds. Note that existing CNN models for pose estimation still suffer from the same drawbacks of gesture recognition techniques: orientation-, user-, and domain-dependence. Therefore, we opt for *PointNet* [21], a point set classification backbone that can generate invariant spatial features from point clouds of the same body pose but with different transformations like permutations, rotations, and translations. Having obtained the pose in each point cloud, mmLock further uses a LSTM network to extract the temporal features and detect the leaving gesture from all poses in the time series. We emphasize that mmLock re-trains a high-quality feature extractor on top of PointNet using a large set of static poses. As a result, mmLock can seamlessly obtain spatial feature vectors for LSTM model training and user leaving detection. Our system is also able to extract and continuously identify the point cloud of the target user in scenarios with attackers.

We design additional measures to reduce the impact of low-quality or irrelevant data on our PointNet-LSTM model. On the one hand, we propose to filter out low-quality frames based on signal strength in the generated range heatmap. On the other hand, we use clustering techniques to remove noise-like points and only extract the point cloud corresponding to the target user for training and detection purpose. In our experiments, we end up using less than 50% of all collected data.

We prototype the entire mmLock system on a commercial off-the-shelf (COTS) TI mmWave radar IWR6843ISK-ODS and thoroughly evaluate its efficacy in user leaving detection. Our experiments involve 16 people consisting of 11 male and 5 female college students and generate over 1 TB mmWave data in total. Through extensive experiments, mmLock is shown to achieve 100% true-positive rate in most of scenarios covering different departing gestures, speed, etc., and low false-positive rate even with nearby attackers.

The rest of the paper is organized as follows. Section II introduces the overview of mmLock. Section IV describes point cloud generation. Section V preprocesses the range heatmap and point cloud. Section VI introduces our PointNet-LSTM model. Section VII details the system implementation and evaluation. Section VIII briefs the related work. Section IX concludes the paper.

II. System Overview

Figure 1 illustrates the system overview of mmLock which uses a mmWave radar to capture the user leaving gesture. mmLock is straightforward in that it uses the radar to obtain the user's point cloud and uses a PointNet-LSTM model to detect the leaving gesture from the input point clouds. The details are as follows. The radar emits FMCW chirp signals and captures the signal reflections from the user body. Then the radar mixes the transmitted and received signals and generates the Intermediate Frequency (IF) signals. After that, the system performs a Fast Fourier Transform (FFT) on the IF signals and outputs a range heatmap corresponding to the distances of surrounding objects. In our experiments, some frames in the range heatmap do not contain useful information to create the point cloud of the target. Therefore, we first preprocess the range heatmap and only extract good frames for later point cloud generation. Provided the range heatmap, we can further compute information including the velocity and the angle of the objects and use them to generate the point clouds. After that, we remove the noise-like points in the point clouds and obtain the cluster of points corresponding to the target user.

The next step is to detect the leaving gesture in the 3D point clouds which are 3D meshes of the user. In particular, we use a hybrid PointNet-LSTM model where PointNet captures spatial features of each point cloud (i.e., the user) and LSTM extracts temporal features in the point clouds characterizing

the user activity. To reduce the training complexity, we first pretrain the PointNet model using static poses that are relevant to leaving gestures. Then we use the pretrained PointNet to extract spatial features from each point cloud for training the LSTM-based detection model. Once the models are trained, we can use the PointNet-LSTM model to detect if there is a user leaving gesture in the input time-series point clouds.

III. Threat Model

We assume that the attacker either finds the lost device or steals it from the device owner. The attacker tries to keep the device and also access the victim's sensitive information there. We aim to ascertain that the device has been locked before in the possession of the attacker. No attempt is made to prevent the attacker from cracking the password or wiping out the device for complete reinstallation.

IV. Point Cloud Generation

In this section, we introduce how to generate point clouds in mmLock. The transmission antennas of our radar first emit FMCW chirps. The signals are then reflected by each part (i.e., point) of the user body and finally received by the receiving antennas of the radar. We can extract information such as range, velocity, and angle of each point from the signals and obtain its 3D coordinate based on such information. All the points extracted from the user body by the radar can constitute a 3D point cloud representing the user. We can imagine that the time-series point clouds of the user can be used to detect the leaving gesture.

A. IF Signal Generation

The mmWave radar used in mmLock is TI IWR6843ISK-ODS which is able to capture most of the user body at short range due to its wide field-of-view (FoV) in both horizontal and elevation planes. As shown in Figure 2b, the radar consists of *three* transmission antennas and *four* receiving ones. Each transmission antenna can emit *frequency-modulated continuous wave* (FMCW) chirps which sweeps frequency from 60 GHz to 64 GHz. The transmitted signal (denoted by $S(t)$) can be written as:

$$S(t) = e^{j(2\pi f_c t + \pi \frac{B}{T_c} t^2)}, \tag{1}$$

where f_c is the starting frequency, B is the bandwidth, and T_c is the duration of each chirp. The current frequency of $S(t)$ is $f = f_c + \frac{B}{T_c} t$ as illustrated in Figure 2a. As introduced above, the transmitted signal will be reflected by all objects around the radar including the user while the receiving antennas receive the reflected signal, i.e., a delayed FMCW chirp. Hence, the received signal (denoted by $R(t)$) can be derived as:

$$R(t) = e^{j(2\pi f_c (t-t_d) + \pi \frac{B}{T_c} (t-t_d)^2)}, \tag{2}$$

where t_d is the round-trip delay. Then the radar uses a frequency mixer to combine the transmitted and received signals, i.e., $S(t)$ and $R(t)$, and generate the intermediate frequency (IF) signal (denoted by $S_{IF}(t)$) as

$$S_{IF}(t) = S(t) \oplus R(t) = e^{j(f_{IF} t + \phi_{IF})}, \tag{3}$$

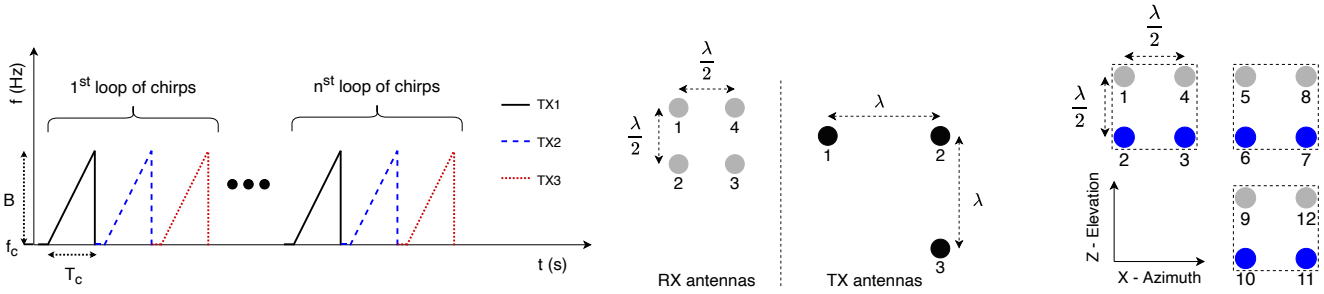
where f_{IF} is the frequency of $S_{IF}(t)$ and equals the frequency difference of $S(t)$ and $R(t)$. Similarly, ϕ_{IF} is the phase of $S_{IF}(t)$ and equals the phase difference of $S(t)$ and $R(t)$.

B. Point Localization

We now extract information such as range, velocity, and angle of the target object from the IF signal, i.e., $S_{IF}(t)$, to localize points in the 3D space. The intuition behind is that signal reflections from different distances generate different frequency components in $S_{IF}(t)$, so we can perform a *Range-FFT* on $S_{IF}(t)$ to separate reflections from different objects.
Distance. For a specific frequency component f in $S_{IF}(t)$, we can calculate the distance between the radar and the object reflecting the signal as $R = \frac{cfT_c}{2B}$, where c is the speed of light.
Velocity. To differentiate between static and moving objects, we can apply another FFT operation on the chirps to measure the phase change of $S_{IF}(t)$ and calculate the velocity of the object. In mmLock, we use *Doppler-FFT* for such purpose, which is effective for removing reflections from static objects such as tables and chairs. Specifically, mmLock uses Doppler-FFT to generate a 2D heatmap and selects prominent pixels corresponding to the points on a potential moving object from the heatmap. The velocity of these points can be calculated as $v = \frac{\lambda \omega}{4\pi T_c}$, where ω is the phase change between two adjacent chirps and λ is the wavelength of the signal.
Angle. To determine the coordinate of the candidate points in the 3D space, we need to know the angles of the incoming reflections in both horizontal and elevation planes. For that purpose, the third FFT operation (i.e., *Angle-FFT*) is applied to the chirps received by different receiving antennas to measure the phase difference. Assume that w_x is the phase difference between adjacent antennas in the horizontal plane. The azimuth θ can be calculated as $\theta = sin^{-1}(\frac{\lambda \omega_x}{2\pi d})$, where d is the distance between two adjacent antennas. In our radar, d equals $\frac{\lambda}{2}$, so $\theta = sin^{-1}(\frac{\omega_x}{\pi})$. Similarly, we can compute the elevation as $\phi = sin^{-1}(\frac{\omega_z}{\pi})$, where w_z is the phase difference between two adjacent antennas in the elevation plane.
3D coordinate. Combining the above information, the 3D coordinate of a point (x, y, z) on an object can be calculated as $x = Rcos(\phi)sin(\theta)$, $y = \sqrt{R^2 - x^2 - z^2}$, and $z = Rsin(\phi)$.

C. Multiple-Input Multiple-Output (MIMO)

As shown in Section IV-B, angle estimations in both horizontal and elevation planes are critical in generating high-quality point clouds, which in turn depend on angle resolution. Note that the angle resolution of a mmWave radar can be derived as $\theta_{res} = \frac{\lambda}{N \times d}$, where N and d are the number of receiving antennas and the distance between two adjacent antennas, respectively. The radar used in mmLock has two receiving antennas in each plane with a distance of $\frac{\lambda}{2}$ from each other. Therefore, θ_{res} in each plane is $1\ rad$ (about $57.32°$). To improve the angle resolution without adding more physical antennas, we use the TDM-MIMO (Time Division Multiplexing Multiple-Input Multiple-Output) technique in



(a) FMCW chirp signals (b) Physical antenna array (c) MIMO virtual antenna array

Fig. 2: FMCW chirps and antenna arrays in the TI IWR6843ISK-ODS radar. (a) shows n loop of FMCW chirps in a frame. Each loop contains three chirps transmitted by three antennas alternatively. (b) illustrates the physical antenna arrays of the radar consisting of three transmission and four receiving antennas. (c) shows the virtual antenna array with the MIMO technique. The blue antennas involve a π phase inversion.

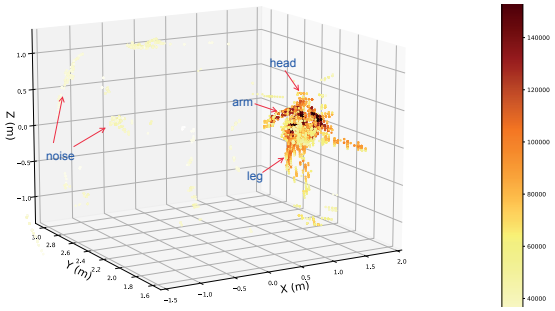


Fig. 3: The raw point cloud of a leaving user

mmLock which enables multiple antennas to transmit alternatively. Specifically, we assign three time slots to every loop showed in Figure 2a and each of the three antennas transmits one FMCW chirp in a time slot. As each chirp is received by all the four receiving antennas, in effect we can simulate 12 virtual antennas using only three transmission antennas and four receiving ones, illustrated in Figure 2c. As a result, we can use four rather than two antennas in each plane for angle estimation, thus improving estimation performance. For example, we can use virtual antennas 1, 4, 5, and 8 to calculate the azimuth θ and use virtual antennas 8, 7, 12, and 11 to calculate the elevation ϕ. According to [22], antennas marked in blue are π out of phase with respect to other antennas marked in gray, so we apply a π phase inversion before calculating θ and ϕ. By using the TDM-MIMO technique, we improve the angle resolution to $\frac{1}{2}$ rad (about $28.66°$) in both horizontal and elevation planes.

D. Preliminary Point Cloud

Figure 3 shows a raw point cloud captured behind a leaving user with the TI IWR6843ISK-ODS radar by combing points of five adjacent frames. In the experiment, the user initially sat on the chair about 1.5 m away from the radar. Then, the user stood up, turned around, and finally left the radar. We use Flat Top window for all the point clouds in the paper because it has minimal scalloping loss in the frequency domain which is desirable for amplitude measurements of sinusoidal frequency components. We can clearly see the head, arms, and legs of the user and some noise-like points in the generated point clouds. One observation is that that signal reflections from the trunk are much stronger than random reflections in the environment and reflections from other parts of the user body (e.g., hands). The figure suggests that we can detect the leaving gesture as we can see how the point cloud of the user moves in the 3D space.

V. Data Preprocessing

In this section, we discuss preprocessing the range heatmap and the point clouds.

A. Range Heatmap Preprocessing

Figure 4 illustrates a range heatmap generated from the IF signals. We collected the reflection signals when the user was about 2.3 meters away from the radar and remained static. We can see that the radar was able to capture the user for some frames (showed in red pixels) but missed for a few frames (showed in blue pixels). We conjecture that the blue range bins do not contain any useful information about the user and thus the corresponding point clouds should not be used for detecting user leaving. Therefore, we aim to extract only the useful frames and filter out the others. We achieve such a goal by calculating the sum of signal strengths in all range bins of a frame and filter out those with lower signal strength. In our experiments, we notice that we only keep 33% of all frames for point cloud generation and user leaving detection.

B. Point Cloud Preprocessing

We describe how we implement point cloud generation here. For one selected pixel in a Doppler heatmap, we can use the TDM-MIMO technique and generate four points in the 3D space for it. Specifically, we first calculate two azimuth angles from virtual arrays $\{1, 4, 5, 8\}$ and $\{2, 3, 6, 7\}$, respectively. Similarly, we calculate two elevation angles from antenna arrays $\{5, 6, 9, 10\}$ and $\{8, 7, 12, 11\}$, respectively. Hence we generate four points from two azimuth angles and two elevation angles by following Section IV-B. Assume 300

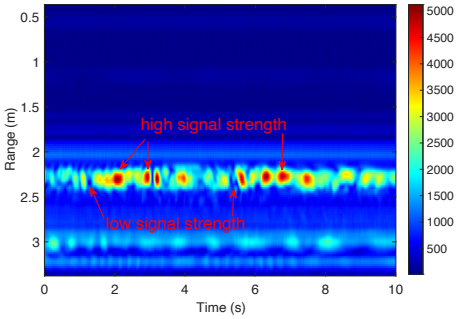


Fig. 4: The range heatmap of a static user

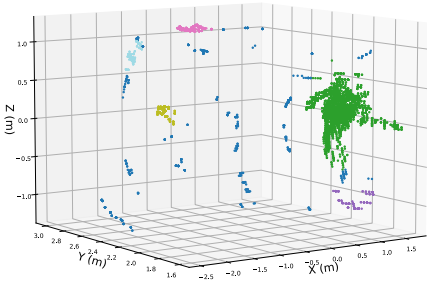


Fig. 5: The clusters of points in Figure 3 generated by DBSCAN. The cluster of green points on the right represents the user.

pixels are selected from one Doppler heatmap, we thus have $300 \times 4 = 1,200$ points from a single frame. Considering that such a point cloud is usually sparse, we further combine the point clouds from five adjacent frames together. Hence we end up with $6,000$ points in the cloud, which we will use for detecting user leaving.

Intuitively, due to wireless reflection, scattering, and refraction in the environment, the generated point cloud is too noisy and calls for de-noising. Our de-noising proceeds in two steps. First, we simply use clustering techniques to group points depending on mutual distance. Then we compute the total energy strength in each cluster and filter out those under a certain threshold. In our implementation, we use DBSCAN [23] to apply clustering. With a maximum neighboring distance of 35 cm and a minimum cluster size of 200, DBSCAN outputs the corresponding clusters in Figure 5 provided the point cloud in Figure 3 as the input. By further computing the total energy strengths, we can obtain the target cluster consisting of green points, which indeed corresponds to the user in our experiment. The last step of preprocessing point clouds is to match the size of an arbitrary cluster to be the one requested from our PointNet-LSTM model. We achieve this in mmLock by either random downsizing (i.e., we remove points randomly) if there are extra points or upsampling using Agglomerative Hierarchical Clustering (AHC) if otherwise. Finally, all preprocessed point clouds, i.e., all extracted clusters, are in the same size of 2,048 points.

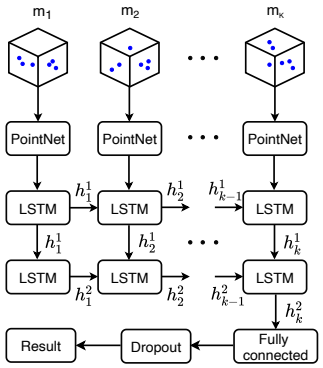


Fig. 6: Network architecture of our point cloud classifier

VI. Leaving Gesture Detection

We first consider leaving gesture detection in environments with only one legitimate user and then discuss how to deal with the attackers.

A. Leaving Gesture Detection via Point Cloud Classification

We assume that the leaving gesture generates K point clouds m_k, $k \in [1, K]$. Our system, mmLock, uses a hybrid model illustrated in Figure 6 to detect the user leaving gesture. In short, mmLock uses *PointNet* [21] to generate the spatial features of each input point cloud, e.g., m_k. Providing the spatial features of all m_k, $k \in [1, K]$, mmLock further uses a stacked LSTM network to extract temporal features from the spatial features and compute the final prediction, which would be 0 or 1.

We use PointNet in mmLock simply because it is a good fit for our user leaving detection. On the one hand, PointNet works consistently regardless of whether the input vector/sample is permuted or altered with linear transformations such as rotation and translation. On the other hand, PointNet extracts local as well as global features. The two items are critical for user leaving detection because we do not have control of the ordering, orientation, rotation, etc. of the input point cloud while we require spatial features to provide more activity information to LSTM network.

Assume that the spatial feature vector generated by PointNet is s_k, where $k \in \{1, 2, ..., K\}$ is the index of point clouds. Then, we send $s_k, k \in \{1, 2, ..., K\}$ to the LSTM network to extract the temporal features. As illustrated in Figure 6, there are $2 \times K$ LSTM units in the LSTM network, each of which consists of a cell, an input gate, an output gate, and a forget gate. We pass the cell state c_t and hidden state vector h_t between adjacent LSTM units and update c_t, h_t in one unit using the following Equation:

$$
\begin{aligned}
f_t &= \sigma_g(W_f x_t + U_f h_{t-1} + b_f), \\
i_t &= \sigma_g(W_i x_t + U_i h_{t-1} + b_i), \\
o_t &= \sigma_g(W_o x_t + U_o h_{t-1} + b_o), \\
\tilde{c} &= \sigma_c(W_c x_t + U_c h_{t-1} + b_c), \\
c_t &= f_t \odot c_{t-1} + i_t \odot \tilde{c}, \\
h_t &= o_t \odot \sigma_h(c_t),
\end{aligned} \tag{4}
$$

where x_t is the input vector for the LSTM unit at step t, W and U are weight matrixes, b is the model bias, h is the hidden state vector of the first or second layer of the LSTM network, σ_g is the sigmoid activation function, σ_c and σ_h are the hyperbolic tangent functions, $\odot$ is the Hadamard product, and f, i, o, c represent forget, input, update, and output gates, respectively. The temporal feature vector is passed to a fully connected layer FC_1, a dropout layer DO_1, another fully connected layer FC_2, and another dropout layer DO_2. Finally, the prediction result can be obtained as the class corresponding to $\max(r)$ from below:

$$
r = (\mathrm{DO}_2 \circ \mathrm{FC}_2 \circ \mathrm{DO}_1 \circ \mathrm{FC}_1 \circ \mathrm{LSTM} \circ \mathrm{LSTM})(s_k), \tag{5}
$$

where $\circ$ denotes the function composition. For example, $(g \circ f)(x)$ represents $g(f(x))$.

B. Leaving Detection in Scenarios with Attackers

Our basic system in VI-A considers only one legitimate user around the device. In this section, we focus on how to detect the leaving gesture of the legitimate user when there exist other moving objects including attackers. Figure 7 illustrates two point clusters representing the legitimate user in the green bounding box and an attacker in the brown bounding box, respectively. The point cloud of the legitimate user is much more clear as he is moving away (Recall we rely on detecting moving objects to generate point clouds). In most cases, the two clusters can be separated easily while the challenge is to associate the clusters in adjacent frames so that the system can track the gesture of our target user continuously. In mmLock, we achieve this based on two metrics of the clusters: center and center of mass. In particular, we calculate the center of a cluster by averaging the coordinates of all its points. This is, $(c_x, c_y, c_z) = \frac{\sum_{i=1}^{n}(x_i, y_i, z_i)}{n}$, where (x_i, y_i, z_i) is the coordinate of the i_{th} point in the cluster. Following that we compute the center of mass of a cluster as $(cm_x, cm_y, cm_z) = \frac{\sum_{i=1}^{n} p_i(x_i, y_i, z_i)}{P_{sum}}$, where p_i is the signal strength of the i_{th} point and p_{sum} is the sum of energy of all the points in the cluster. The system associates clusters in two neighboring frames by the closest center and center of mass because the object can only move a small distance during such a period.

In the case When the attacker comes very close to the legitimate user, the two clusters may merge with each other and are thus regarded as one cluster by DBSCAN. The merged cluster may lead to a false positive or negative in mmLock. We observe that most of the points in two close clusters do not overlap with each other and just merge into a larger cluster.

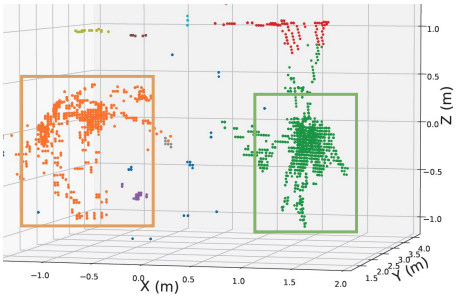


Fig. 7: Point clusters of an environment with an attacker

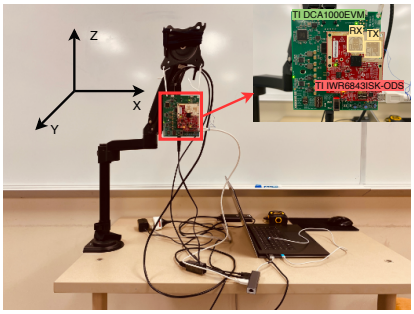


Fig. 8: Testbed and environmental setup

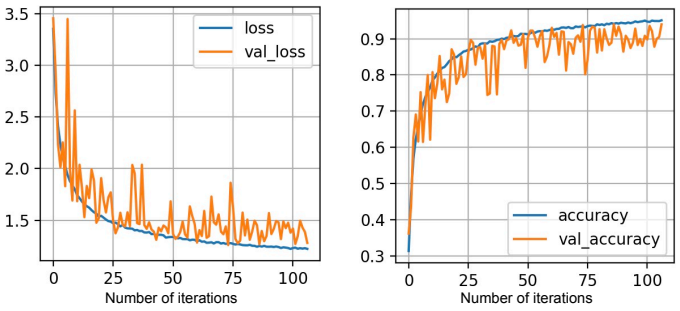
That is, the dramatically increased size of a cluster indicates the merging of two clusters. The system separates them by checking point coordinates of previous frames.

VII. Implementation and Evaluation

A. Testbeds

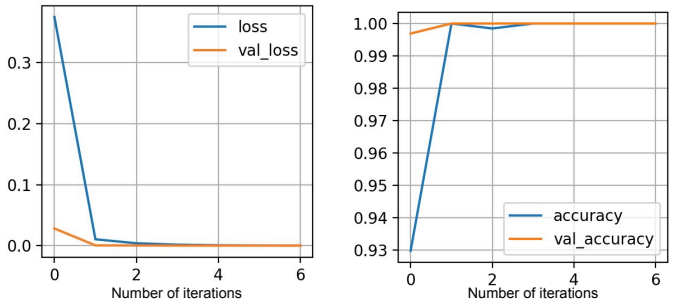
mmWave Testbed. The mmWave radar used in our paper is the Texas Instruments IWR6843ISK-ODS which is a commercial and portable mmWave sensing board as illustrated in Figure 8. The radar operates in a frequency band from 60 GHz to 64 GHz with a 4 mm wavelength. It has three transmitting antennas and four receiving antennas and can cover a sensing FoV of 120° in the E-plane and 120° in the H-plane. We also use the DCA1000EVM board for realtime data capture and streaming from the mmWave radar. We configure the radar to sample 63 frames per second with 24 chirp loops per frame in the training phase to generate more point clouds and sample 33 frames per second with 18 chirp loops per frame in the testing phase. Each chirp consists of 578 data samples. The above configuration supports a maximum detectable range of 15 m with 3.9 cm resolution and a maximum detectable velocity of 2 m/s with 0.27 m/s resolution.

Computing Environment. We generate the point cloud data and train the deep learning model using a custom-built server in the paper. The server has a AMD 5950x 3.4G CPU, a NVIDIA 3090 GPU consisting of 24 GB VRAM, and 128 GB 3600 MHz DDR4 RAM.



(a) Training loss (b) Training accuracy

Fig. 9: Training results of PointNet



(a) Training loss (b) Training accuracy

Fig. 10: Training results of LSTM

B. Model Training

We divide the PointNet-LSTM model training process into three steps. First, we pretain PointNet based on static user poses. After that, we use the pre-trained PointNet model to extract spatial features from each point cloud of dynamic gestures. Finally, we use the spatial features train the LSTM model for user leaving detection.

Unless specifically noticed, our experiments in the paper are done in a $24' \times 24'$ classroom with chairs and tables around. We install the radar on a monitor stand of 1.2 m height as illustrated in Figure 8.

1) PointNet Pretrain: The PointNet pretrain involves 16 college students including 11 males and 5 females of different ages and heights. We design 34 static poses consisting of at least 14 poses which are related to leaving gesture as illustrated in Table I. For each static pose, we ask the student to keep the pose for 24 s and collect 1500 frames which generate about 100 point clouds for the model training.

Since PointNet accepts 2048 3D points, the input dimension of the PointNet network is 2048×3. The output dimension of PointNet is the one-hot encoding of 34×1. We use the categorical cross entropy as the loss and an Adam Optimizer with a learning rate of 5×10^{-4}. To avoid overfitting, we use an early stopping strategy to monitor $val\_accuracy$ during the training process with the patience of 20. The batch size used in the training is 32. We also use the multi-process acceleration of 4 processes to send data into the GPU and process the results returned by the GPU. Figure 9 shows the loss and accuracy trend during the PointNet training process. The classification accuracy reaches 95% after 100 iterations.

2) LSTM Training: To train the LSTM model, we include 7 college students in the experiments to generate point clouds of dynamic gestures. We design 5 common leaving gestures and 4 unrelated gestures. Each gesture is repeated for 110 times in total and generates 30 point clouds. We use 66.7% of the data as the training dataset and the rest of the data as the testing dataset.

We first use the pre-trained PointNet to extract spatial features of the 2048×3 input from each point cloud of the dynamic gestures in the penultimate layer. Then we pass the spatial features to the LSTM model which receives features of 30 frames at the same time and outputs the one-hot encoded 0 and 1 representing leaving gestures and unrelated gestures, respectively. The LSTM model consists of two layers of 100 units, a dense layer of 256 units, a dropout layer of 0.5, another dense layer of 256 units, another dropout layer of 0.5, and an output layer of 2 units. The LSTM model also uses categorical cross entropy as the loss and an Adam Optimizer with a learning rate of 10^{-4}. To avoid overfitting during training, we use an early stopping mechanism based on $val\_accuracy$ with the patience of 20 and save the best weights. We use a batch size of 8 and 4 processes to speed up the interaction between the CPU and GPU. Figure 10 shows the training results of the LSTM model. The model converges much faster than the PointNet model and reaches 100% classification accuracy.

C. Evaluation with a Single User

We first evaluate the false negatives and positives of the system with a single user surrounding the radar in the default experimental setting. After that, we evaluate the system performance with different departing gestures, initial positions, leaving angles, radar heights, etc.

False Negatives We conduct leaving detection experiments with 6 college students in the default experimental setting illustrated in Figure 8. Initially, the users either sit on a chair or stand facing the radar in the position showed with the blue square in 90° in Figure 12. The users leave the radar in their usual way and generate 366 leaving samples. Our system successfully detects all the leaving gestures, which lead to a true-positive rate of 100% or a false-negative rate of 0%.

False Positives. We evaluate the false-positive rate with the same group of users in the default experimental setting. In the experiment, the users perform common non-leaving gestures and movements in front of the radar such as swinging back-and-forth, swinging left-and-right, standing up, sitting down, and gentle exercise. The system falsely recognizes four gestures as leaving gestures, which lead to a false-positive rate of 1.67%.

Impact of Departing Gestures. We then evaluate the impact of different departing gestures on the system performance. In the experiment, we evaluate 5 different leaving gestures and perform each gesture 20 times. The first one is our default gesture in which the user stands up, turns right, and walks

TABLE I: Static poses used for PointNet pretraining

Index	Pose Description	Index	Pose Description
1	Upright sitting facing the radar	18	Standing facing right-front
2	Leaning forward in chair	19	Facing left-front and stepping forward with left foot
3	Halfway between sitting and standing up facing the radar	20	Facing left-front and step forward with right foot
4	Standing facing the radar	21	Standing facing left-front
5	cStepping left-front with right foot and turning left a little bit	22	Upright sitting facing the radar and lifting arms
6	Facing left-rear and keeping weight on left foot	23	Keeping pose 22 and turning left
7	cFacing left-rear and stepping forward with left foot	24	Keeping pose 22 and turning right
8	cFacing left-rear and stepping forward with right foot	25	Upright sitting and opening arms
9	cStepping right-front with left foot and turning right a little bit	26	Keeping pose 25 and turning left
10	Facing right-rear and keeping weight on right foot	27	Keeping pose 25 and turning right
11	cFacing right-rear and stepping forward with right foot	28	Keeping pose 1 and turning upper body left
12	cFacing right-rear and stepping forward with left foot	29	Keeping pose 1 and turning upper body right
13	Sitting facing left	30	Upright sitting facing the radar and raising both hands
14	Sitting facing right	31	Upright sitting facing the radar with hands behind head
15	Sitting rear facing	32	Standing facing the radar and stretching hands forward
16	Facing right-front and stepping forward with right foot	33	Standing facing the radar with hands on hips
17	Facing right-front and stepping forward with left foot	34	Standing facing the radar and opening arms

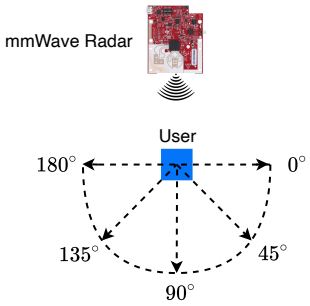


Fig. 11: Experimental setup with five leaving angles

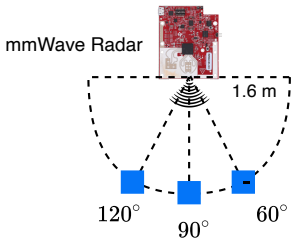


Fig. 12: Experimental setup with three departing positions

away. In the second gesture, the user stands up, turns left, and walks away. In the third gesture, the user rotates the chair, stands up, and then walks away. In the fourth gesture, the user initially stands in front of the table, turns right, and walks away. In the last gesture, the user initially stands in front of the table, turns left, and walks away. The system successfully recognizes all the departing gestures.

Impact of Leaving Angles. We now investigate the impact of different leaving angles. In the experiment, the user leaves in 5 different angles from 0° to 180° as illustrated in Figure 11. The user performs leaving gestures 20 times in each angle. The system correctly recognizes leaving gestures in all angles.

Impact of Departing Positions. Each user may leave with different initial positions. We evaluate three different departing positions showed in blue squares in Figure 12 and perform leaving gesture 20 times in each position. The system correctly recognizes all 60 leaving gestures with three departing positions.

Impact of Initial Distance. To evaluate the impact of the initial distance between the user and radar, we perform leaving gestures with four different distances: 1 m, 1.6 m, 2.1 m, and 3.8 m. The user performs leaving gestures 20 times for each distance. The system successfully recognizes leaving gestures in all distances. Therefore, our system can also be used in other scenarios with larger distances between the device and user.

Impact of Departing Speeds. We evaluate the impact of user moving speeds, with slow, normal, and fast speeds, corresponding to about 1.1, 1.5, and 2.0 steps/second, respectively. The user performs leaving gestures 20 times with each speed. Fortunately, the system correctly recognizes leaving gestures in all speeds because of the high sampling rate.

Impact of Vertical Positions. We now investigate the impact of the vertical positions of the mmWave radar. In the experiments, we place the radar in four different heights: 1 m, 1.2 m, 1.4 m, and 1.6 m. The user performs the leaving gesture 20 times for each height. The system can recognize leaving gestures in all the heights. Therefore, mmLock can be used for devices placed on tables of different heights.

Impact of Experimental Environments. We evaluate our system in the lobby and study areas of our university library with 40 leaving gestures. The lobby is about 30,000 square feet and contains many tables, chairs and public desktop computers. During our experiment, there is a lot of noise from the vending machines, public computers, and students. In addition, the students walk around without our control. Our system correctly detects all leaving gestures.

D. Evaluation in Scenarios with Attackers

The above evaluations only consider one legitimate user who is close to the radar. We now evaluate the system performance when there is an attacker in the surrounding of the radar. With the presence of the attacker, mmLock can detect multiple movement traces and needs to decide which trace is associated with the target user. For this experiment, we use the Precision and Recall metrics defined as follows,

$$\texttt{Precision} = \frac{\#\texttt{TP}}{\#\texttt{TP} + \#\texttt{FP}}, \texttt{Recall} \quad = \frac{\#\texttt{TP}}{\#\texttt{TP} + \#\texttt{FN}} \tag{6}$$

where $\#\texttt{TP}$ is the number of user departures correctly associated with the user, $\#\texttt{FP}$ is the number of the attacker's departures incorrectly associated with the user, and $\#\texttt{FN}$ refers to the number of user departures not associated with the user by mistake.

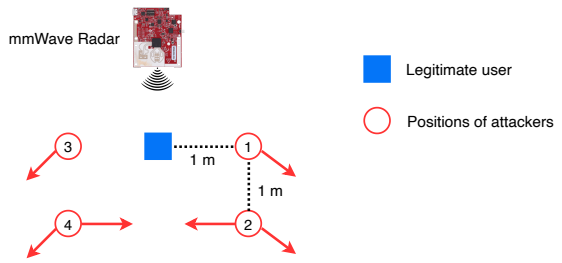


Fig. 13: Experimental setup with nearby attackers. The red arrows denote the attackers' leaving directions.

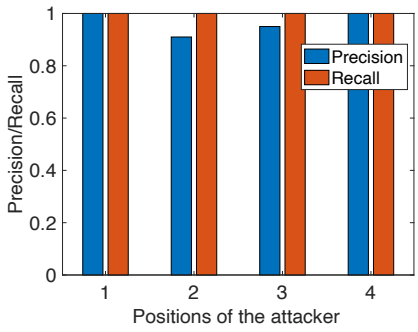


Fig. 14: `Precision` and `Recall` with a nearby attacker

As illustrated in Figure 13, the blue square represents the legitimate user and the red circles show the positions of the attackers. For each position from 1 to 4, both the legitimate user and the attacker perform leaving gestures for 20 times. In addition, the attacker walks back and forth between position 4 and 2. As a result, we can calculate precision and recall rates for each position and a false-positive rate for the walking of the attacker between position 4 and 2.

Figure 14 illustrates the precision and recall when there is an attacker in the positions showed in Figure 13. The system achieves precision above 90% for all the positions. In particular, the precision for position 2 and 3 are 91% and 95%, respectively. Three leaving gestures performed by the attacker are recognized as legitimate user because of incorrect point cluster association in different frames. The recall rate for all the four positions are 100% which means no leaving gestures performed by the legitimate user is missed by the system. The attacker walking between position 2 and 4 does not trigger any false positives.

VIII. Related Work

There are three ways to prevent the attackers' illegal access to smart devices and the sensitive data therein. The first one is one-time authentication which has been widely used on current smart devices to authenticates users when they try to unlock and use the device. Based on the information used for authentication, one-time authentication schemes can be classified into three categories: *Something-You-Know*, *Someone-You-Are*, and *Something-You-Have*. In *Something-You-Know* paradigm, users input PINs or passwords to unlock the device. The *Someone-You-Are* paradigm relies on physiological or behavioral biometrics which are unique to each person. Common physical features include fingerprints, retina patterns, and facial features. Behavioral features consists of keystroke patterns [24], touching gestures [25], gaits [26], etc. The *Something-You-Have* paradigm requires auxiliary hardware (e.g. Signet Ring [27]) which is possessed only by the legitimate user. One-time authentication cannot prevent data theft if the user leaves and forgets to lock the device which may be controlled by attackers in the unlock mode.

The second method is to authenticate users continuously when they are using the device [4], [28]. In this way, after the attacker uses the device for a while, the device can detect the unauthorized user and log out based passively collected sensing data in the device. In [4], the user needs to wear a bracket with a built-in motion sensors to be authenticated by a laptop while typing. A paper [28] points out attacks on the technique in [4]. However, continuous authentication can only detect the attacker after he has used the device for a while. As a result, the attacker still has a good chance to obtain the victim's sensitive data before being logged out. In addition, if the attacker just watches content (e.g., photos and messages) on the screen and does not use the device, he would not be detected by continuous authentication methods at all.

The third mechanism is to lock the device immediately once the user has left. Existing work [6] [7] in this category relies on acoustic ranging techniques to detect the user leaving gesture. But these techniques require line-of-sight between the device and user in the whole leaving process which may not be always available because of blocks caused by objects around the device. In addition, acoustic-based techniques can hardly differentiate the leaving user from other moving objects due to the limited number of microphones. mmLock can detect the leaving gesture faster at the beginning of the leaving process and identify the target user in environments with multiple moving objects accurately.

Our system is also related to previous work on gesture recognition with wireless signals [19], [29]. The most traditional way is to recognize the gesture based on the received signal strength indicator (RSSI) which measures the distance and channel between the transmitter and receiver. Each gesture may generate a unique signal pattern which can be used for gesture recognition [15], [16]. Channel state information (CSI) is a more popular technique used for gesture recognition in recent years. CSI measures the channel properties and reflects scattering, fading, and power decay of a communication link. CSI patterns have been used in many systems and achieve better performance than RSSI because of its high sensitivity to human movements [18], [19]. Recent papers reduce training efforts by extracting environment-independent features [30], [31]. However, same gestures in different orientations may generate different signal patterns which is an obstacle for the applications of above techniques. mmLock constructs the intuitive user point cloud first and recognizes the leaving gestures in the point clouds. Pantomime [32] recognizes hand gestures based on point clouds generated from mmWave signals. But it

did not provide a solution to extract and continuously identify the point cloud of the target user in scenarios with multiple moving objects. In addition, we generate higher resolution full-body images which support more accurate leaving detection.

Recently, some papers developed imaging systems based on mmWave signals [33]. [34] generates 3D human meshes using a deep learning model and relies on the 3D human model extracted from a sophisticated 3D camera system as the ground truth. [35] generates 3D point clouds for static objects using a customized mmWave radar with 6 TX and 8 RX antennas. mmLock generates high-quality point clouds directly to characterize the user leaving gestures with COTS mmWave radar and does not rely on vision-assisted training.

IX. Conclusion

In this paper, we designed and evaluated mmLock, a user leaving detection system against data theft based on the TI mmWave radar. mmLock first generates high-quality point cloud for the target user and recognizes the leaving gesture in the point cloud. In contrast to the previous work based on acoustic ranging, our system is more robust and can identify the target user in environments with multiple moving objects. Extensive experiments on TI mmWave radar confirmed the high efficacy of mmLock with negligible false positives and negatives.

X. Acknowledgement

This work was supported in part by the US National Science Foundation under grants CNS-2245760 and CNS-2055751.

References

[1] Statista, "Forecast number of mobile devices worldwide from 2020 to 2025." 2018. [Online]. Available: https://www.statista.com/statistics/245501/multiple-mobile-device-ownership-worldwide/

[2] Asurion, "What to do if your phone is lost or stolen." 2021. [Online]. Available: https://www.asurion.com/connect/tech-tips/what-to-do-when-your-phone-is-lost-or-stolen.

[3] Verizon, "2021 data breach investigations report." 2021. [Online]. Available: https://www.verizon.com/business/resources/reports/dbir/2021/incident-classification-patterns/lost-and-stolen-assets/.

[4] S. Mare, A. Molina-Markham, C. Cornelius, R. Peterson, and D. Kotz, "ZEBRA: Zero-effort bilateral recurring authentication," in *IEEE S&P*, San Jose, CA, May 2014.

[5] M. Frank, R. Biedert, E.-D. Ma, I. Martinovic, and D. Song, "Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 1, pp. 136–148, 2013.

[6] T. Li, Y. Chen, J. Sun, X. Jin, and Y. Zhang, "ilock: Immediate and automatic locking of mobile devices against data theft," in *ACM CCS*, Vienna, Austria, October 2016.

[7] J. Chen, U. Hengartner, H. Khan, and M. Mannan, "Chaperone: real-time locking and loss prevention for smartphones," in *USENIX Security*, Virtual, August 2020.

[8] "Qualcomm snapdragon 5g mmwave." 2022. [Online]. Available: https://www.qualcomm.com/5g/mmwave/.

[9] "Qualcomm 802.11ad mmwave." 2022. [Online]. Available: https://www.qualcomm.com/products/features/80211ad/.

[10] J. Guan, S. Madani, S. Jog, and H. Hassanieh, "High resolution millimeter wave imaging for self-driving cars," in *https://arxiv.org/abs/1912.09579*, 2022.

[11] C. Xu, H. Li, Z. Li, H. Zhang, A. S. Rathore, X. Chen, K. Wang, M.-c. Huang, and W. Xu, "Cardiacwave: A mmwave-based scheme of non-contact and high-definition heart activity computing," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 5, no. 3, pp. 1–26, 2021.

[12] F. Wang, F. Zhang, C. Wu, B. Wang, and K. R. Liu, "Vimo: Multiperson vital sign monitoring using commodity millimeter-wave radio," *IEEE Internet of Things Journal*, vol. 8, no. 3, pp. 1294–1307, 2020.

[13] A. Dhekne, M. Gowda, Y. Zhao, H. Hassanieh, and R. R. Choudhury, "Liquid: A wireless liquid identifier," in *ACM MobiSys*, Munich, Germany, June 2018.

[14] C. Jiang, J. Guo, Y. He, M. Jin, S. Li, and Y. Liu, "mmvib: micrometer-level vibration measurement with mmwave radar," in *ACM MobiCom*, London, United Kingdom, September 2020.

[15] H. Abdelnasser, M. Youssef, and K. Harras, "Wigest: A ubiquitous wifi-based gesture recognition system," in *IEEE INFOCOM*, Hong Kong, April 2015.

[16] J. Wang, X. Zhang, Q. Gao, H. Yue, and H. Wang, "Device-free wireless localization and activity recognition: A deep learning approach," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 7, pp. 6258–6267, 2016.

[17] S. Sigg, S. Shi, F. Buesching, Y. Ji, and L. Wolf, "Leveraging rf-channel fluctuation for activity recognition: Active and passive systems, continuous and rssi-based signal features," in *ACM MoMM*, Vienna, Austria, December 2013.

[18] X. Guo, B. Liu, C. Shi, H. Liu, Y. Chen, and M. C. Chuah, "Wifi-enabled smart human dynamics monitoring," in *ACM SenSys*, Delft, The Netherlands, November 2017.

[19] R. Venkatnarayan, G. Page, and M. Shahzad, "Multi-user gesture recognition using wifi," in *ACM MobiSys*, Munich, Germany, June 2018.

[20] D. Wu, D. Zhang, C. Xu, Y. Wang, and H. Wang, "Widir: walking direction estimation using wireless signals," in *ACM PerCom*, Sydney, Australia, March 2016.

[21] C. Qi, H. Su, K. Mo, and L. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *IEEE CVPR*, Honolulu, HI, July 2017.

[22] "User's guide 60ghz mmwave sensor evms of texas instruments," 2022. [Online]. Available: https://www.ti.com/lit/pdf/swru546/.

[23] M. Ester, H. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *ACM KDD*, Portand, OR, August 1996.

[24] E. Maiorana, P. Campisi, N. Gonzalez, and A. Neri, "Keystroke dynamics authentication for mobile phones," in *ACM SAC*, TaiChung, Taiwan, March 2011.

[25] H. Chen, F. Li, W. Du, S. Yang, M. Conn, and Y. Wang, "Listen to your fingers: User authentication based on geometry biometrics of touch gesture," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 4, no. 3, pp. 1–23, 2020.

[26] J. Kwapisz, G. Weiss, and S. Moore, "Cell phone-based biometric identification," in *IEEE BTAS*, Washington DC, September 2010.

[27] T. Vu, A. Baid, S. Gao, M. Gruteser, R. Howard, J. Lindqvist, P. Spasojevic, and J. Walling, "Distinguishing users with capacitive touch communication," in *ACM MobiCom*, Istanbul, Turkey, August 2012.

[28] O. Huhta, P. Shrestha, S. Udar, M. Juuti, N. Saxena, and N. Asokan, "Pitfalls in designing zero-effort deauthentication: Opportunistic human observation attacks," in *NDSS*, San Diego, CA, February 2015.

[29] K. Niu, F. Zhang, J. Xiong, X. Li, E. Yi, and D. Zhang, "Boosting fine-grained activity sensing by embracing wireless multipath effects," in *ACM CoNEXT*, Sydney, Australia, December 2018.

[30] Y. Zheng, Y. Zhang, K. Qian, G. Zhang, Y. Liu, C. Wu, and Z. Yang, "Zero-effort cross-domain gesture recognition with wi-fi," in *ACM MobiSys*, Seoul, Korea, 2019.

[31] Z. Shi, J. Zhang, R. Xu, Q. Cheng, and A. Pearce, "Towards environment-independent human activity recognition using deep learning and enhanced csi," in *IEEE GLOBECOM*, Virtual, December 2020.

[32] S. Palipana, D. Salami, L. A. Leiva, and S. Sigg, "Pantomime: Mid-air gesture recognition with sparse millimeter-wave radar point clouds," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 5, no. 1, pp. 1–27, 2021.

[33] Y. Zhu, Y. Zhu, B. Y. Zhao, and H. Zheng, "Reusing 60ghz radios for mobile radar imaging," in *ACM MobiCom*, Paris, France, September 2015.

[34] H. Xue, Y. Ju, C. Miao, Y. Wang, S. Wang, A. Zhang, and L. Su, "mmmesh: Towards 3d real-time dynamic human mesh construction using millimeter-wave," in *ACM MobiSys*, Virtual, June 2021.

[35] K. Qian, Z. He, and X. Zhang, "3d point cloud generation with millimeter-wave radar," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 4, no. 4, pp. 1–23, 2020.