This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2024.3515044

mmFinger: Talk to Smart Devices with Finger Tapping Gesture

Xuan Wang, Xuerong Zhao, Chao Feng, Dingyi Fang, Xiaojiang Chen

Abstract—Contact-free finger gesture recognition unlocks plenty of applications in smart Human-Computer Interaction (HCI). However, existing solutions either require users to wear sensors on their fingers or use continuously monitored cameras, raising concerns regarding user comfort and privacy. In this paper, we propose mmFinger, an accurate and robust mmWave-based finger gesture recognition system that can extend the range of available custom commands. The core idea is that mmFinger leverages the finger tapping pattern as a basic gesture and encodes different number combinations of the basic gesture like Morse code. To enable reliable recognition across different locations and for various users, we carefully design a robust feature Dop-profile to effectively characterize finger movements. Furthermore, by leveraging the multi-views provided by multiple antennas of radar, we develop an adaptive weighted feature fusion network to enhance the system's robustness. Finally, we devise a novel sequence prediction network to enable the system to recognize new gestures without retraining. Comprehensive experiments demonstrate that mmFinger can achieve an average recognition accuracy of 92% for 36 predefined gestures and 88% for 5 new user-defined commands, and is robust against finger location and user diversity.

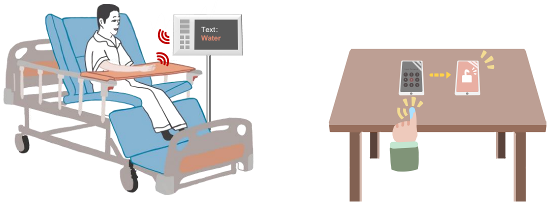
Index Terms—Finger gesture recognition; mmWave radar; HCI.

1 Introduction

Contact-free Human-Computer Interaction (HCI) is becoming increasingly popular and enables plentiful appealing applications [1]–[4]. For example, one can unlock a phone and access information without touching the screen when the fingers are wet, oily, or dirty, or wearing gloves. Similarly, in a hospital, motor neuron disease (MND) patients can interact with computers and smart devices in their homes by tapping their fingers. Furthermore, in supermarkets, customers can purchase goods without contacting self-service devices.

A natural solution for achieving contact-free interaction is voice-based [4]. Although promising, it is not suitable for quiet places and could raise privacy concerns in public settings. In addition, such a solution is ill-suited to mutes. Another option is to adopt gesture-based approaches [5], [6]. By tracking and recognizing hand or finger gestures, one can interact with the devices. Due to flexible and no need for the users to make a sound or use voice commands, gesture-based HCI schemes are more attractive.

Existing gesture recognition systems either rely on cameras [7], [8] or wearable sensors [9]–[12]. While effective, such solutions rely on ambient light conditions, incurring privacy infringement, and causing uncomfortable to users. To avoid these issues, recent advances have explored diverse wireless signals, e.g., WiFi [13], [14], RFID [15], [16], acoustic [17] and mmWave [18], [19], for gesture sensing. Although they have made great progress, there are some limitations hindering their practical usage. First, these works only identify a limited number of pre-defined gestures, which yet are not unified but different from system to system and difficult to be remembered. Second, they are susceptible to location changes and user changes. Once the location changes, the performance significantly degrades. Third, some systems employ large-scale hand movements for interaction, which would incur information leak issues as the content to express behind gestures such as interactive mode, input messages or instructions is evident and easy to be seen, stolen, and mimicked [19], [20].



(a) Disabled people to interact with the medical device. (b) Contact-free phone unlock.

Fig. 1. Potential interaction applications of finger gesture recognition.

Therefore, in this paper, we ask the following question: *Can we design an accurate, reliable, and privacy-preserving HCI approach without being limited to pre-defined gestures?* We propose an affirmative answer through mmFinger, a mmWave-based finger gesture recognition system. We leverage finger tapping (finger down and up once, like clicking the mouse) as a basic element and encode all letters, numbers, and custom commands into finger gesture combinations consisting of single and double taps. By recognizing the combination of the basic tapping elements, mmFinger can identify not only pre-defined gestures but also new gestures. With such capability, one system can enable many potential interaction applications as shown in Fig. 1.

- *Xuan Wang, Xuerong Zhao, Chao Feng, Dingyi Fang, Xiaojiang Chen are with the School of Information Science and Technology, Northwest University, China. E-mail: {xwang, xrzhao}@stumail.nwu.edu.cn, {chaofeng, dyf, xjchen}@nwu.edu.cn.*

Realizing such an idea into a practical system, we face several technical challenges. The first challenge lies in weak target reflection signal extraction. Due to the limited finger movement and small reflection area of the finger, the signal variation induced by a finger gesture is minimal and thus can easily be overwhelmed by stronger background interference, making it difficult to accurately extract the finger-reflected signal. The second challenge is the system performance degradation caused by minor location variations between the finger and radar, as well as different user typing habits. These factors can result in pattern variations of the reflected signal for the same finger gesture, ultimately leading to errors in recognition. This issue is particularly pronounced for individuals with severe physical disabilities who may have limited control over their finger movements.

The third challenge lies in scaling the recognition system to new gestures for meeting users' diverse needs. Existing methods mostly employ classification-based approaches that categorize finger gestures into limited predefined groups and assign them to specific letters or commands, which requires regathering new samples and updating the recognition model. Re-training model is computationally intensive and impractical for terminal devices with limited computing capabilities.

To overcome the above challenges, mmFinger proposes a pipeline of signal processing schemes, including signal refinement and spacial-temporal feature extraction, along with a deep learning (DL) recognition model. First, to obtain the weak signal reflected by the user's finger, mmFinger adopts a range-based location algorithm and a circle-fitting algorithm to eliminate interference caused by surrounding static and dynamic objects. Second, to reliably characterize finger gestures, we design a more robust feature Dop-profile which can reflect a consistent trajectory of the finger movement, to minimize the impact of finger location variations and user habits. Moreover, to further enhance recognition robustness, we fully leverage the complementary information provided by multiple antennas on mmWave radar to characterize the spatial variation of finger movement. To solve the third challenge, our basic idea is to use different finger-tapping patterns to match with self-defined commands or existing number combinations like Morse or ASCII codes and transform the finger-tapping pattern recognition task into a sequence prediction task. By doing so, we can significantly reduce the overhead of integrating new finger gestures into the recognition model to adapt to other applications.

We implement our system with a commodity mmWave device and evaluate the system performance in a typical indoor environment. Extensive experiments show that mmFinger can recognize 36 finger gestures and 5 user-defined commands with an average accuracy of 92.61% and 88%, respectively. The results also demonstrate that mmFinger is robust against input location and user diversity.

The main contributions can be summarized as follows.

- mmFinger is a mmWave-based finger gesture recognition system to enable interaction with humans and devices just using an index finger. It can generalize across users and locations, and easily scale to new finger gestures without retraining effort.
- We utilize unique finger-tapping patterns to match self-defined or standard codes like Morse and ASCII, enabling the system to identify new gestures without additional training or learning. We also develop a feature extraction technique that combines Doppler profiles and spatial-temporal analysis with a multi-antenna fusion approach on mmWave radar, capturing intricate finger motion details.
- Extensive real-world experiments demonstrate the effectiveness and robustness of mmFinger.

2 RELATED WORK

In this section, we discuss the related studies about gesture recognition and mmWave radar-based sensing tasks.

2.1 Contact-free finger gesture recognition

Contact-free finger gesture recognition has garnered significant attention from both academia and industry due to its potential for a wide range of real-life HCI applications. Existing research in this field can be categorized into three main approaches. The first approach involves wearable devices such as smart gloves or wristbands equipped with sensors [9], [10]. These devices are capable of capturing hand and finger movements and orientation. While effective, this solution can be inconvenient for users and may cause discomfort. Alternatively, the second approach utilizes vision-based methods [7], [8]. This method eliminates the need for users to wear or hold any devices and relies on cameras or depth sensors to capture images or depth maps of the hand and fingers. Computer vision techniques are then applied to analyze the captured data and recognize the gestures. However, this approach is susceptible to variations in ambient light conditions and raises concerns regarding privacy infringement. The third approach is based on wireless signals, such as ultrasonic [17], [21], WiFi [13], [14], [22], radar [18], [19], [23]. Sensors emit signals that measure the time it takes for the waves to bounce back after being reflected by the hand or fingers. The captured data can be utilized to estimate hand and finger positions and activities. Soli [23] presents a mmWave sensor to showcase fine hand gesture interaction. WiKey [22] leverages CSI-waveform to recognize 37 keys in a fixed keyboard. However, these works are limited to identifying a predefined set of gestures and vulnerable to change of relative position between finger and transceiver, making it challenging to extend recognition systems to new gestures and impeding their practical deployment in real-life scenarios. In Taprint [11], users need to move their fingers and click on the corresponding position of the virtual numeric keyboard, which is difficult to use for MND patients (they can only move their fingers within a centimeter). In contrast, mmFinger enables interaction with humans and devices just using an index finger in a contactless way. The main advantage of the proposed system is that it can easily scale to new finger gestures without retraining effort.

2.2 mmWave radar-based sensing applications

Millimeter wave radar has been widely used in wireless sensing applications [24]–[28] due to its wide bandwidth and high resolution. [29] and [30] use point cloud information extracted by commercial mmWave radar to recognize gestures. m3Track [31] realized a mmWave-based multi-user 3D posture tracking system. mHomeGes [29] proposed a real-time mmWave arm gesture recognition system for practical smart home-usage. [32]enables a virtual keyboard by detecting small changes in finger position.

In addition, the fine-grained sensing ability of mmWave radar can be used in many applications to monitor human health. [33] enables a non-contact high-definition heart monitoring, [34], [35] enable RF vital sign sensing under ambulant daily living conditions through capturing the sophisticated correlation between RF signal pattern, movement power, and vital signs. [36], [37]extract extremely weak reflected signals from mechanical equipment to measure micrometer-level vibrations. Unlike previous works, mmFinger aims to develop a gesture recognition system based on Morse code using mmWave signals, which is especially beneficial for individuals with physical disabilities in human-computer interaction.

3 Motivation and Background

In this section, we first describe the motivation. Then, we describe the data processing of Frequency Modulated Continuous (FMCW) mmWave radar for further analysis.

3.1 Applications for finger gesture recognition.

By accurately predicting and recognizing finger gestures, numerous captivating applications become feasible. For example,

Providing communication service for ALS (Amyotrophic Lateral Sclerosis) patients: ALS, also known as Motor Neuron Disease (MND), is a progressive neurological disorder primarily affecting motor neurons. According to statistics, the global MND patients are over 500,000, with 12,000 of them in the United States [38], [39]. Such patients typically suffer from muscle weakness and speech difficulties and only retain slight finger and eye movements. Thus, traditional communication approaches, such as speech and gestures, are ineffective for MND patients. Previous efforts have involved the development of eye-tracking systems to aid computer interaction and communication for these patients [40]. Yet, vision-based eye-tracking systems are often intricate and costly, and sustained eye movement can be fatiguing. More importantly, such a solution causes privacy invasion. Instead, we use RF signals to provide a non-invasive solution to facilitate communication for MND patients by identifying single-finger movements. In addition, since MND patients still have normal brain functions, they thus can remember some finger-tapping interaction rules to talk with smart devices and people. Although bringing certain memory burdens, they can communicate with the outside world in a non-intrusive, contact-free, and privacy-preserve manner, which is more important for them.

Supporting finger-level interactions: In a smart home, one can control devices in smart homes through minor, simple gestures like adjusting volume, changing channels, or managing lighting. In addition, we can manipulate virtual objects in virtual and augmented reality.

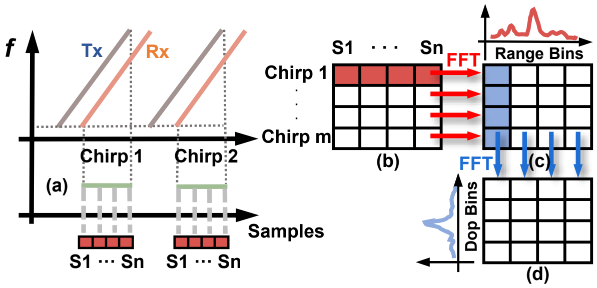


Fig. 2. FMCW radar data processing.

3.2 Basics of mmWave Radar

The mmWave radar emits Frequency Modulated Continuous (FMCW) signals that are reflected and received by the radar upon hitting a target. At the receiver, the received signal is typically multiplied by the complex conjugate of the transmitted signal to produce an intermediate frequency (IF) signal that is easier to sample, as the step (a) in Fig. 2. Mathematically, the IF signal can be represented as follows,

$$S_{IF}(t) = Ae^{-j2\pi\left\{f_0\tau(t)+\mu t\tau(t)-\frac{1}{2}\mu^2\tau^2(t)\right\}}. \tag{1}$$

where $\tau(t) = \frac{2R_0+vt}{c}$, R_0 is the distance from target to radar, v is the target velocity, t is the propagation time. μ is the slope of the FMCW signal. Next, we detail two commonly used features, i.e., distance, and Doppler velocity.

Distance estimation: When the reflecting object is stationary, the sampled intermediate frequency (IF) signal maintains a constant frequency of $f = \frac{S\times 2\times R}{c}$. Therefore, a fast Fourier transform (FFT) can be performed on a single chirp to estimate of the range R by identifying the peak of the frequency spectrum. This procedure is called range-FFT (from step (b) to (c) in Fig. 2). The range-FFT transforms the data from the time domain to the frequency domain, enabling the detection of targets at different ranges. The range resolution is inversely proportional to the bandwidth of the frequency sweep.

Doppler velocity estimation: Small human motions relative to radar can cause Doppler frequency shifts (DFS), resulting in range-FFTs corresponding to each chirp with peaks in the same range bin but with differing phases. By computing the phase difference ω between two consecutive chirps, we can estimate the Doppler velocity as $v = \frac{\lambda\omega}{4\pi T_c}$, where T_c represents the time interval between two chirps. This can be accomplished by further performing an FFT on each range bin of the Range-FFT, which is called a Doppler-FFT (from step (c) to (d) in Fig. 2).

4 System Overview

mmFinger is an accurate and reliable finger gesture recognition system built on a commodity mmWave radar. The system structure is illustrated in Fig. 3. It comprises three major modules: data collection and pre-processing module, feature extraction module, and gesture prediction module.

Data Collection and Pre-processing Module: mmFinger first uses a commodity mmWave radar to collect raw measurements when each user performs finger gestures. Then, to obtain weak target reflections induced by a finger,

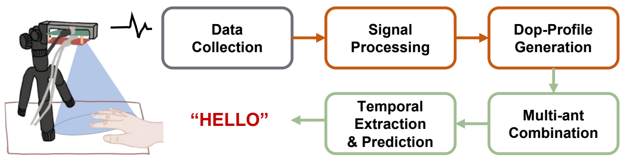


Fig. 3. mmFinger's overview.

mmFinger adopts a circle-fitting algorithm and a range-based location algorithm to eliminate interference caused by surrounding static and dynamic objects.

Spacial-Temporal Feature Extraction Module: Since minor location variations between the finger and radar could result in significant differences in the reflected signal, mmFinger designs a location-agnostic feature, i.e., Dop-profile, to reliably characterize the finger gestures. In addition, to fully leverage the advantage of the multiple viewing angles provided by multiple transmitter and receiver antennas, mmFinger designs a multi-antennas adaptive combination scheme to adaptively generate weights for each antenna of each training sample and then fuses them, greatly reducing the effects of noise and interference and obtaining more accurate recognition results. To capture the implicit temporal characteristic of the fused features, mmFinger designs an LSTM-based feature extraction module.

Gesture Prediction Module: Then, the obtained temporal representation is fed into the sequence prediction module, which maps the representation to a character sequence using the CTC loss function. By doing so, mmFinger can make predictions without requiring gesture segmentation, thus avoiding recognition errors caused by sub-gesture segmentation errors. This approach also enables the recognition of custom gestures without requiring specific training data for each gesture, which makes the system more versatile and flexible for a wide range of users and applications.

5 System Design

In this section, we elaborate on the system design of mmFinger, which is developed to recognize finger gestures using mmWave radar. mmFinger comprises three major modules: the signal pre-processing module, the dynamic feature extraction module, and the gesture prediction module.

5.1 mmWave data preprocessing

Due to the small reflection area and movement of the finger, the reflected signals are weak and easily overwhelmed by ambient reflections from other larger objects. This poses a significant challenge for detecting the signals of interest. To address this issue, we adopt a series of preprocessing steps to extract the weak target reflection signals generated by finger movements.

Generally, the received signal comprises not only the target signal reflected by the finger but also signals from other static and dynamic objects. We here refer to the reflected signals from static objects and dynamic non-target signals as static clutter and dynamic clutter, respectively. Thus, the received intermediate frequency (IF) signal $\boldsymbol{S}$ can be expressed as the sum of the target dynamic signal $\boldsymbol{S_t}$, dynamic clutter $\boldsymbol{S_d}$, static clutter $\boldsymbol{S_0}$, and noise:

$$S = \underbrace{A_t e^{j\cdot 2\pi f(\Delta\tau_{ta}(t))}}_{S_t} + \underbrace{\sum_k A_k e^{j\cdot 2\pi f_k(\Delta\tau_{dk}(t))}}_{S_d} + \underbrace{\sum_i A_i e^{j\cdot 2\pi f_i \tau_i}}_{S_0} + \text{DC} + \text{Noise} \tag{2}$$

where DC denotes the Direct Constant (DC) component.

To remove the static and dynamic clutters induced by distant objects (e.g., a wall and a walking person) relative to the human target, we adopt a range-based location algorithm. The basic idea is that the sources of interference fall in different range bins compared to the finger. Therefore, we first perform a range Fast Fourier Transform (FFT) to the IF signal and then employ a classical CFAR algorithm [41] to the range spectrum for extracting the distance information between the finger and radar antenna. By doing so, we can eliminate reflections from surrounding static objects and interference from people who may occasionally pass by.

In practice, there could exist static clutters reflection from the table on which the finger is tapped. It means that the table can be detected within the same range-bin as the finger reflection. If the static component is not suppressed, the sensing resolution and accuracy will degrade [36]. Static interferences act similarly to DC components, shifting the IQ sampling point distribution on the IQ diagram away from the origin so that it no longer centers on the origin. This displacement diminishes the observable amplitude of signal phase evolution over time. Therefore, we can correct the estimated deviation of the phase series so that the center of the circle returns to the origin to eliminate the phase offset. The specific operation is, first, to find the distribution of the current discrete points on the complex plane and perform a circle fitting algorithm [36] on these points, compensate for the offset based on the fitted circle, and finally obtain new discrete points by subtracting it from the received complex signal to remove the static component eliminate the static component caused by table reflection.

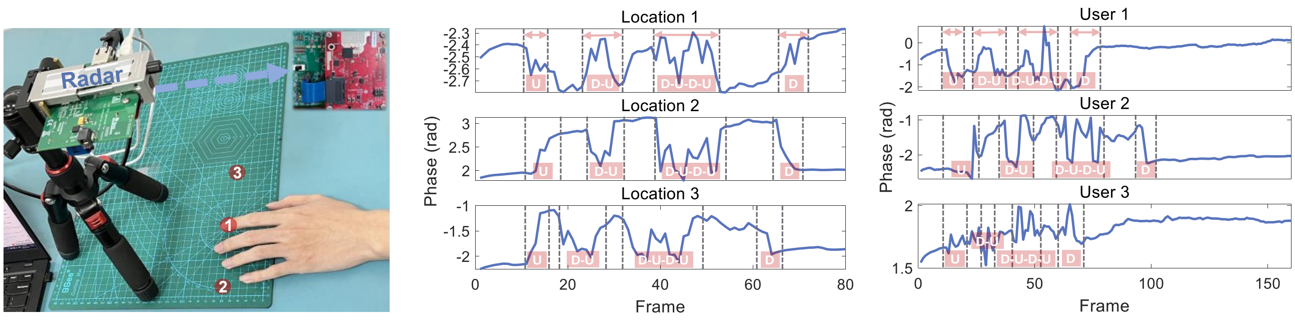
5.2 Characterize the movement of finger

So far, we have extracted the valid signal of the finger movement. Thus, our next goal is to robustly characterize the movement of the finger. In this section, we first analyze why phase information can not be directly used for the modeling of target movements. Then, we describe how to devise a feature, i.e., Dop-profile, to represent the finger movement, which is robust to the finger locations and performer's identity.

5.2.1 *Phase variation*

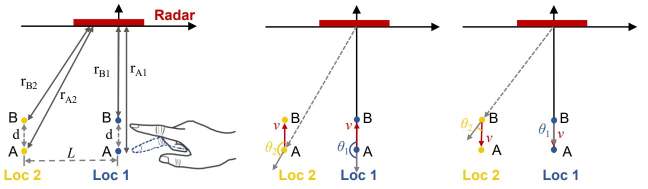
The movement of the finger in the up-down direction is typically small (less than 1 cm). However, the range resolution of a mmWave radar with 4 GHz bandwidth is 3.75 cm, which is not sufficient to capture such a small variation in range. Fortunately, the small wavelength of mmWave signals (about 4 mm) allows us to utilize phase information to characterize this movement.

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2024.3515044



(a) Experimental setup. (b) Different locations. (c) Different users.

Fig. 4. The phase variation at three locations and with three users.



(a) Sensing model. (b) Gesture "up". (c) Gesture "down".

Fig. 5. Impact analysis of finger location.

As mentioned in Sec. 3.2, the phase change of the IF signal corresponds to the path length change caused by target movement, we thus can express the phase variation as follows:

$$\phi(t) = 2\pi \cdot \frac{2R(t)}{\lambda} = \frac{4\pi(R_0 \pm v \cdot t)}{\lambda} \tag{3}$$

The equation in Eq. 3 shows that the phase change is capable of capturing finger movements. However, it is also vulnerable to slight changes in location or orientation between the finger and radar. These changes can result in significant phase variations in the received signal, making it challenging to accurately detect and recognize gestures. Furthermore, the thickness of the user's fingers, as well as the magnitude and speed of the performed gestures, can also impact the phase of the received signal. The high sensitivity of the phase to minor changes in gesture motion presents a challenge for achieving accurate gesture recognition.

To investigate the impact of the relative position of the radar and finger on phase measurements, we conduct benchmark experiments, as shown in Fig. 4(a). We fix a mmWave radar on a desktop at a height of 25 cm and ask users to perform the letter "A" containing four sub-gestures at three marked locations (Loc 1 - Loc 3), respectively. We then extracted phase variations from the target range bin. Fig. 4(b) displays the phase variations caused by the gesture at different locations, and we can see that the phase patterns are different at different locations. Fig. 4(c) shows the phase variation patterns of different users at Loc 1, and we can also see that the patterns vary significantly with the performer's identity. These results demonstrate that phase variation is highly dependent on the location and user. Thus, we cannot use phase information directly to characterize finger gestures accurately.

5.2.2 *Dop-profile extraction*

To mitigate the problem of phase instability, we propose a feature, i.e., Dop-profile, to represent the finger movement. Different from phase patterns, the Dop-profile feature is more responsive to the movements of the target's gestures while simultaneously being less susceptible to minor positional alterations. We now detail how to obtain the Dop-profile feature.

In the radar sensing model illustrated in Fig. 5(a), wherein the transmitter and receiver are co-located and stationary, the movement of the finger is analogous to the relative motion of the transmitter and receiver. If the target is moving radially with respect to the radar, then the values of R and phase ϕ will vary with time. By calculating the derivative of phase with time, we can successfully eliminate the impact of the factor R_0:

$$\omega = \frac{\mathrm{d}\phi}{\mathrm{d}t} = \frac{\mathrm{d}\frac{4\pi(R_0 \pm v \cdot t)}{\lambda}}{\mathrm{d}t} = \frac{4\pi v}{\lambda} = 2\pi \cdot \frac{2v}{\lambda} \tag{4}$$

where $\frac{2v}{\lambda}$ is the Doppler frequency shift (DFS) induced by the target. We denote it as f_D, which can also be expressed as the difference between the frequency of the transmitted signals f_s and received signals f_r:

$$f_D = f_r - f_s = \frac{2v}{\lambda} = -\frac{2v_t \cos(\theta)}{c} f_s \tag{5}$$

v_t denotes the real velocity of the target, and θ denotes the angle formed between the direction of movement and the Line of Sight (LoS) stretching from the transceiver to the target. The radial velocity v determines the magnitude and direction of DFS caused by the target. *For the same gesture, although they were performed at different locations or by different users, the moving direction and trajectory of the finger are the same.* The moving direction and velocity derived from DFS are more consistent across different users, locations, and distances.

We take the starting gesture "up" (Fig. 5(b)) and ending gesture "down" (Fig. 5(c)) as examples for analysis. In Fig. 5(b), we observe that when the finger executes the "up" gesture at Loc 1, f_D reaches its maximum value. This is because the finger moves towards the radar and $\cos(\theta_1) = -1(\theta_1 = 180°)$. The "up" gesture has a state sequence of static-accelerate-slow down-static, causing the DFS to first rise from 0 and then decline to 0. Alternatively, when the gesture is performed at Loc 2, which is located at a distance of L from Loc 1, $\cos(\theta_2) = -\frac{L}{\sqrt{(r_{A1}^2 + L^2)}}$. When

Loc 2 is positioned 50° away (i.e., 6 dB bandwidth) to the left of Loc 1, resulting in $\cos(\theta_2) = -0.6428$, the Doppler velocity v_2 at Loc 2 is equal to $0.6428v_1$. Fig. 6(a) and (b) show the Doppler velocity of two different locations. v_2 is significantly less than the v_1 and its profile is blurred to be recognized. By changing the observation angle, as depicted in Fig. 6(d), we observed that the magnitude profile of maximum Doppler velocities corresponding to each frame still exhibited a distinct pattern for all the sub-gestures as Fig. 6(c), we call it Dop-profile.

Therefore, we extract Dop-profile to characterize the movement of the finger in mmFinger. The specific processing procedure is as follows. Initially, a Range-FFT is performed separately on N sampling points of each chirp within reshaped datacube C_1. Following this, background interference cancellation is executed, and a Doppler-FFT is performed on the M points (all M chirps) of each range bin within the matrix obtained in the previous step, resulting in the Range-Doppler matrix X. We accumulate the energy across all M points within each range bin and select the largest range bin i. Similarly, the energy of all N points within each Doppler bin is accumulated, and the largest Doppler bin j is chosen. We extract the energy of $x_{i,j}$ for the current frame q. These steps are repeated for all frames, generating an energy value $x_{i,j}^q$ for each frame to form the Dop profile Dp.

To display the validity of the proposed feature Dop-profile, in addition to the benchmark experiment in Fig. 4(a), we also conduct an experiment that varied the distance between the finger and the radar by lifting the radar to three different heights (25 cm, 45 cm, 65 cm). Fig. 7 (a)-(c) show the captured Dop-profiles from finger movements at three distinct locations, by three different users, and at three distances, respectively. Their Dop-profiles have a unified pattern, demonstrating that the proposed Dop-profile feature is capable of effectively characterizing finger movements while minimizing the impact of location, individual differences, and distances.

5.3 Spacial-Temporal Feature Extraction Module

This module aims to further capture complex spacial and temporal features that reflect finger movement over time. The spacial-temporal features are fed into the prediction module to achieve robust gesture recognition.

5.3.1 Multi-antenna Based Spatial Feature Extraction

To further robustly represent the finger gestures, we introduce a muli-antenna combination scheme to enrich the Dop-profile features. Generally, a commodity mmWave radars consist of multiple transmit antennas (Tx) and multiple receive antennas (Rx). In our work, we utilize the mmWave chip TI IWR1843, whose antenna layout is displayed in Fig. 8. This chip contains four Rx antennas with an element spacing of $dl = \frac{\lambda}{2}$, resulting in phase differences between signals arriving at these antennas. For simplicity, we use the signal transmitted from Tx1 and arrived at Rx1 as the reference, so the phase difference between Rx2 - Rx4 and the reference are ω, 2ω, and 3ω, respectively ($\omega = \frac{2\pi dl}{\lambda}$). Since Tx2 is 2λ apart from Tx1, the phase difference of the signal sent by Tx2 when it arrives at the Rx antennas are 4ω, 5ω, 6ω, and 7ω. Similarly, as shown in Fig. 8(b), the phase difference of the signal sent by Tx3 received by Rx1-Rx4 can be equivalent to Rx9-Rx12. *The virtual antenna array can provide multi-view for observing finger movements, leading to differences in the received signal variations.* This is due to two reasons. Firstly, the length of the virtual antenna array is larger than the width of a single finger, and secondly, the surface of a finger is not as smooth and reflective as that of a metal plate, which can cause variations in the scattering of the signal. To verify the diversity of radar antennas when performing gestures with a finger, we ask a user to perform gesture "A". Fig. 9 shows the Dop-profiles extracted by all virtual antennas. We can see that different antennas have different Dop-profiles. This provides us an opportunity to employ antenna diversity for achieving robust gesture recognition.

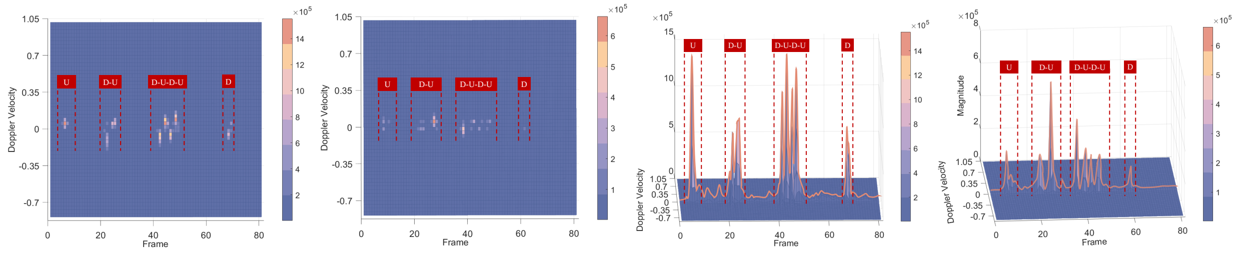
However, a challenge we faced is that how to effectively incorporate the readings from the virtual antenna array. One straightforward method is to directly add all antennas' Dop-profiles. However, such a solution can not obtain optimal performance since some antennas may have negative effects, causing accuracy degradation. To tackle this problem, we propose MAF (Multi-antennas fusion), an adaptive weighting approach to make the network automatically assign weights for each antenna, and then incorporate information from them to extract optimal gesture features. Specifically, this adaptive feature fusion method comprises three operators: extraction, weight adaptive selection, and fusion. The extraction operator is first applied to eliminate gesture-irrelevant features for input Dop-profiles from all antennas. The selection operator combines information from various channels to obtain a comprehensive representation that facilitates weight selection. The fusion operator aggregates feature maps of all channels based on the selected weights for subsequent temporal feature extraction. The proposed adaptive feature fusion module significantly enhances the quality of feature extraction and improves recognition robustness by fusing the representations of multi-channel inputs. Fig. 10 shows the structure of the module and the specific details of each module are described below.

Extraction: To eliminate the impact of gesture-independent features, we employ a weight shared Convolutional Neural Network (CNN) block which takes as input the extracted Dop-profiles from all 12 antennas. The CNN block contains several layers, including convolutional layers (Conv), batch normalization layers (BN), ReLU layers, dropout layers, and dense layers. By sharing weights across the input channels, the CNN block can effectively capture the gesture-specific features while filtering out the irrelevant ones.

Selection: After obtaining the High-dimensional representation for each antenna, we fuse results from multiple branches (twelve in our system) via an element-wise summation:

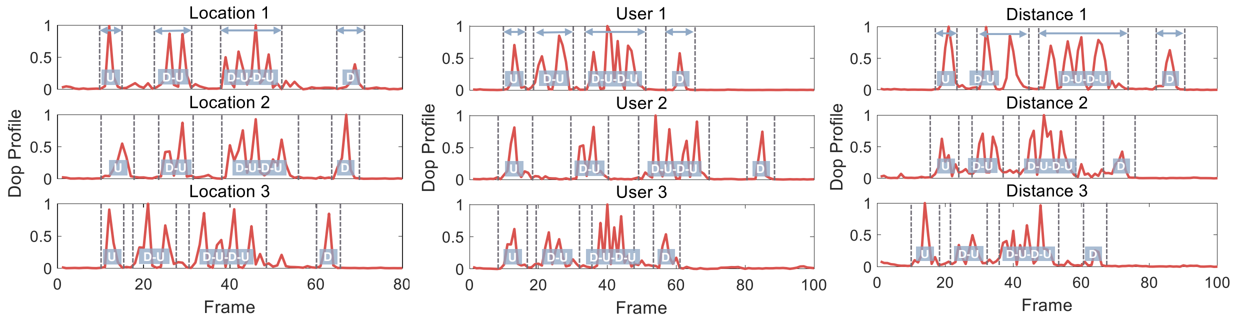
$$V = v_1 + v_2 + \cdots + v_K, \tag{6}$$

then we added a Global Average Pooling layer F_{gb} to effectively summarizes the most important features and reduce the spatial dimensions of the feature. The resulting feature fusion result is then fed as input to a fully connected (FC) layer F_{fc} to generate $Z = z_1, z_2, \cdots, z_K$.



(a) Doppler of Loc 1 (b) Doppler of Loc 2 (c) Dop-profile of Loc 1 (d) Dop-profile of Loc 2

Fig. 6. Robust Dop-profile extraction.



(a) Different locations. (b) Different users. (c) Different distances.

Fig. 7. Dop profiles across different factors.

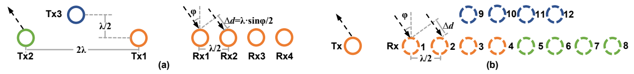


Fig. 8. Virtual multi-antennas array.

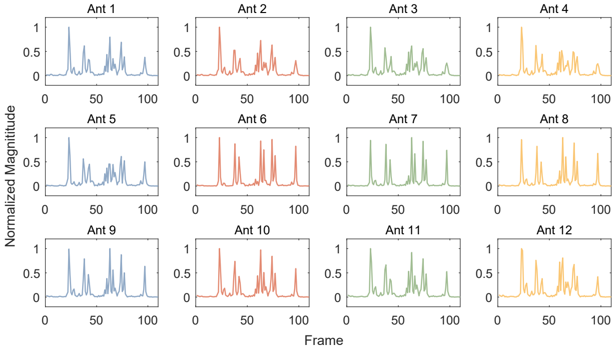


Fig. 9. Dop profiles of all virtual antennas

A $softmax$ across channels is used to adaptively select the information from different antennas. Specifically, a softmax operator is applied on the channel dimension (antenna dimension) to generate weight vectors,

$$\alpha_k = \{\alpha_k^1, \alpha_k^2, \cdots, \alpha_k^j\}, \tag{7}$$

where $\alpha_k^j = \frac{exp(k_j/\varepsilon)}{\sum_k (exp(k_j/\varepsilon))}$, k_j is the j_{th} element of the input z_k and the sum is taken over the j_{th} element of all k input. $\sum_k \alpha_k^j = 1$. The output of the softmax function is a vector of the same dimension as the input, with each element being a value between 0 and 1 that represents the probability of the corresponding class.

Fusion: The result of the module is a linear combination of k channels $(v_1, v_2, \cdots, v_k)$ each weighted by the corresponding coefficient vector $(\alpha_1, \alpha_2, \cdots, \alpha_k)$, which can be expressed as,

$$F = \alpha_1 \cdot v_1 + \alpha_2 \cdot v_2 + \cdots + \alpha_k \cdot v_k \tag{8}$$

where F is vector with the same dimension as v_k. The coefficients α_k determines the contribution of each vectors v_k to the resulting linear combination.

5.3.2 LSTM Based Temporal Feature Extraction

We apply Bidirectional Long Short-Term Memory (BiLSTM [42]) to model the temporal dependencies between the different elements of the input sequence. The choice of BiLSTM is based on several considerations. Firstly, compared to traditional RNNs, the gating mechanism in BiLSTM helps mitigate the issue of vanishing gradients when handling long sequences. Secondly, BiLSTM's sensitivity to temporal dynamics allows it to effectively capture patterns in time series data, making it advantageous for our prediction tasks. While Transformers are widely used for sequence data, BiLSTMs are computationally more efficient for shorter sequences, making them better suited for real-time applications. Additionally, BiLSTM offers lower model complexity and is easier to debug and optimize, whereas Transformers require a larger dataset to achieve optimal performance. Given the time-sequential nature of gesture data, BiLSTM is a more appropriate choice for our needs. By analyzing both the forward and backward sequences, BiLSTM can capture the velocity and acceleration information. By using these rich, complex features as input to the prediction module, the model can learn to distinguish between different gestures based on their unique features.

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2024.3515044

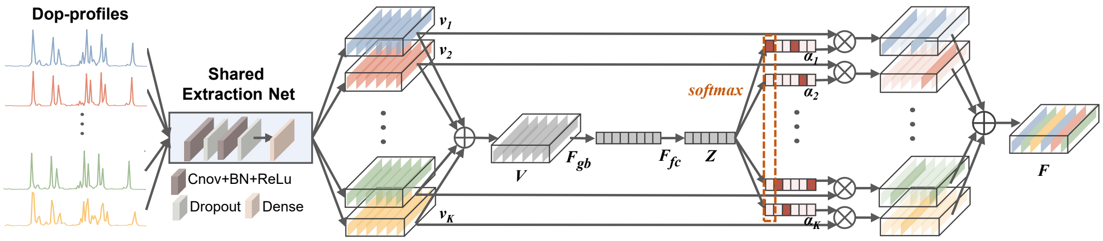
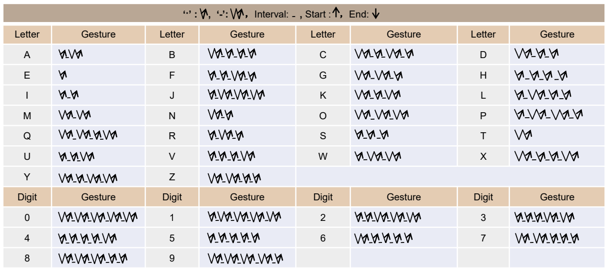


Fig. 10. Multi-antennas combination module.



‘·’ : ·, ‘-’: -, Interval: _ , Start : ↑, End: ↓

Letter	Gesture	Letter	Gesture	Letter	Gesture	Letter	Gesture
A	·_-	B	-_·_·_·	C	-_·_-_·	D	-_·_·
E	·	F	·_·_-_·	G	-_-_·	H	·_·_·_·
I	·_·	J	·_-_-_-	K	-_·_-	L	·_-_·_·
M	-_-	N	-_·	O	-_-_-	P	·_-_-_·
Q	-_-_·_-	R	·_-_·	S	·_·_·	T	-
U	·_·_-	V	·_·_·_-	W	·_-_-	X	-_·_·_-
Y	-_·_-_-	Z	-_-_·_·				
Digit	Gesture	Digit	Gesture	Digit	Gesture	Digit	Gesture
0	-_-_-_-_-	1	·_-_-_-_-	2	·_·_-_-_-	3	·_·_·_-_-
4	·_·_·_·_-	5	·_·_·_·_·	6	-_·_·_·_·	7	-_-_·_·_·
8	-_-_-_·_·	9	-_-_-_-_·				

Fig. 11. Summary of the finger tapping patterns.

5.4 Gesture Recognition Module

To predict the label of a finger gesture, a straightforward solution is to use a classic classifier network [8] to map temporal features to a specific category. However, directly performing multi-classification tasks by treating each character as a separate category can be a cumbersome process, especially when more characters or user-defined commands are needed for recognition. This is because there is no corresponding category for the new character in the original network, we have to retain the network when new characters are added. This approach can be impractical for users and lacks user-friendliness in practical applications.

To cope with this challenge, our basic idea is to use different finger-tapping patterns to match with self-defined commands or existing number combinations like Morse or ASCII codes and transform the finger-tapping pattern recognition task into a sequence prediction task. In our study, we choose Morse code-like gesture patterns as an example to illustrate our fundamental concepts. Note that this strategy can be adapted and integrated into other current language systems, users thus can randomly group the basic finger-tapping gesture to customize their interactive semantics based on their preferences.

Specifically, we employ a one-finger input method and assign predefined tapping patterns to represent characters referring to Morse code encoding rules, where the “down-up” gesture represents “dot” and “down-up-down-up” represents “dash”. To separate individual characters, we begin with an “up” gesture and ends with a “down” gesture for each respective character. The combination of these gestures in a sequence is used to encode the letters “A-Z” and digits “0-9”, as illustrated in Fig. 11.

To enable the prediction of new finger-tapping gesture sequences without retraining the network, we treat each character as a sequence of gestures using numbers “0-3” to represent four sub-gestures: starting (up), single tapping (down-up), double tapping (down-up-down-up), and ending (down). For example, “A” and “B” are represented as “0123” and “021113”, respectively.

Based on this proposed strategy, we use Connectionist Temporal Classification (CTC) [43] as a probabilistic approach to translate temporal features into character sequences to recognize new gestures at zero manpower cost and zero learning curve. CTC loss function can quantify the dissimilarity between the input temporal sequence and the actual output after neural network processing. It introduces a blank symbol to the output space and considers all possible alignments between the input and output sequences, including those with repeated symbols and different lengths. By summing over all possible alignments and then removing the blank symbols to obtain the final predicted output sequence. This approach allows for end-to-end translation of characters without pre-segmenting individual gestures and connecting them into a complete sequence of character gestures. In mathematical notation, the CTC loss function The loss function is defined as:

$$\mathcal{L} = -\sum_{\boldsymbol{y} \in Y} \ln p(\boldsymbol{y} \mid \boldsymbol{x}) \tag{9}$$

Here, $\boldsymbol{y}$ denotes the target sequence, Y denotes the set of all possible target sequences, $\boldsymbol{x}$ denotes the input temporal representation, and $p(\boldsymbol{y}|\boldsymbol{x})$ denotes the probability of the target sequence $\boldsymbol{y}$ given the input sequence $\boldsymbol{x}$. The logarithmic form avoids numerical underflow and improves computation stability while making the loss function more interpretable as it measures the negative log-likelihood of the correct target sequence given the input sequence.

The process of minimizing the loss function is typically implemented using optimization algorithms such as gradient descent. In our system, we adopt the Adam optimizer [44], which adaptively adjusts the learning rate for each parameter and dynamically adjusts the size of the learning rate during the training process. This helps improve the efficiency and accuracy of the system.

Finally, we create a LUT to map gesture sequences to tapping patterns and recognize finger gestures by predicting a number sequence consisting of several “1” and “2”, a “0” and a “3”. By doing so, our system can break the bottleneck that requires collecting samples of new gestures to retrain the model.

6 Implementation

mmWave radar platform and parameters setting. For data collection, we utilize a commercial mmWave radar sensor chip (TI-IWR1843BOOST [45]) with three transmitting antennas and four receiving antennas. To capture the subtle finger movements, the radar is configured to transmit

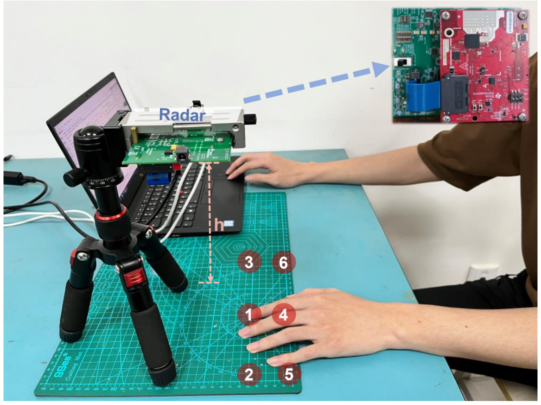


Fig. 12. Experimental deployments.

TABLE 1
Summary of Datasets Collection

	Samples Description
Dstaset 1	26 Letters × 15 Instances × 10 User = 3900 Samples
Dataset 2	10 Digits × 20 Instances × 10 User = 2000 Samples
Dataset 3	5 Commands × 20 Instances × 10 User = 1000 Samples
Dataset 4	6 Distances × 5 Instances × 26 Letters × 2 User = 780 Samples
Dataset 5	6 Positions × 5 Instances × 26 Letters × 2 User = 1560 Samples
Dataset 6	3 Env × 10 Instances × 26 Letters × 2 User = 1560 Samples
Dataset 7	3 Dep × 10 Instances × 26 Letters × 2 User = 1560 Samples

FMCW signals with a bandwidth of 3.985 GHz and a slope of 69.9 MHz/us. It sends 160 frames each time, and each frame contains 255 × 3 chirps (three transmitting antennas work in a time-division multiplexing manner). In addition, we set sampling points per chirp to 100, and the receiver is set to a sampling rate of 2 MHz. The sensor is attached to a tripod to maintain a fixed position and orientation during data collection.

Post-processing platform. The data obtained from the TI radar is preprocessed on a PC (Intel i5-10400F CPU @ 2.90 GHz, 32 GB memory). Then, extracted Dop-profile features are stored and fed into the server equipped with NVIDIA GeForce RTX 2080Ti GPU for deep learning based recognition.

Deployment and data collection. We summarize the collected datasets in Table 1. The **Dataset 1-3** are collected under the default deployment as Fig. 12 in a conference room. The height of radar h = 25 cm and finger typing at Location 1. We invite ten volunteers (4 female and 6 male) to perform finger tapping gestures and our study is approved by the Institutional Review Board (IRB). The specific deployments of **Dataset 4-7** are described in corresponding section.

Hyperparameters Declaration and Sample Selection Strategy. In our training process, we employ 100 iterations (epochs) to train the network using the "adam" optimizer with an initial learning rate of 0.01. We set the batch size to 16, that is, each iteration involves 16 samples. We set dropout to 0.25 to reduce the number of parameters and avoid overfitting.

Evaluation metrics: We apply character recognition accuracy as an evaluation metric. The recognition accuracy is the percentage of correctly predicted characters to the total predicted characters, which can be described as:

$$Accuracy = \frac{Num.\ of\ correctly\ predicted\ character}{Num.\ of\ all\ test\ character} \tag{10}$$

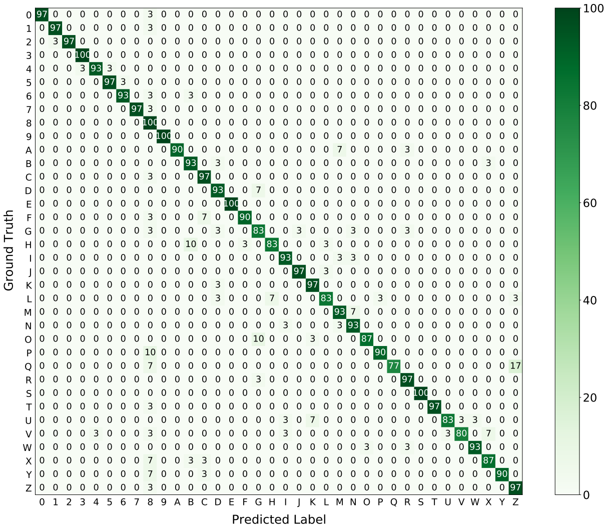


Fig. 13. Confusion matrix of 36 finger gestures across 10 participants.

TABLE 2
Comparison of feature extraction methods and machine learning models.

Test Num.	Feature	Methods	Average Accuracy	Params ($\times 10^5$)
1	Dop-profile of single antenna	CNN+LSTM	89.14%	3.8
2	Doppler-Time of single antenna	CNN+LSTM	88.04%	90.32
3	Dop-profiles of multi antennas	CNN+LSTM	89.68%	52.2
4	Dop-profiles of multi antennas	ResNet+LSTM	90.38%	74.19
5	Dop-profile of multi antennas	Atten-TsNN	90.85%	52.38
6	**Dop-profile of multi antennas**	**MAF+LSTM**	**93.61%**	**4.1**

7 Evaluation

7.1 Overall Performance

We evaluate the overall performance of mmFinger on **Dataset 1** and **Dataset 2**. Specifically, we select 80% of 4300 samples from 10 users for training and the remaining samples for testing. During model training, the training set is divided into five parts, and a four-fold rotation is employed, wherein four parts are used for training and one part for validation, following the five-fold cross-validation method. The model with the highest validation performance is ultimately retained. Fig.13 shows the confusion matrix of 36 finger gestures across 10 participants, where the dark green color indicates higher recognition accuracy while the white color means (close to) zero prediction. We can see that mmFinger achieves an average recognition accuracy of 92.61%. The results demonstrate that mmFinger can effectively recognize finger gestures.

7.2 Micro-benchmarks

To verify the effectiveness of our extracted feature Dop-profile and machine learning model, we conduct two microbenchmarks on **Dataset 1**, and the results are shown in Table 2. Specifically,

Verification of the feature extraction method: To verify the effectiveness of our proposed feature extraction method, we compared our extracted Dop-profile of a single antenna, commonly used Doppler-Time feature, and Dop-profile of multi antennas by inputting them into the same feature extraction network (ResNet+LSTM) as test 1-3.

From the comparison of recognition results, we can see that while the accuracy of dop-profle and Doppler-time features shows minimal variance, the number of parameters is significantly reduced by 30 times. This reduction occurs because that our dop-profile features are one-dimensional data, allowing the convolution kernel to operate along a single dimension. In contrast, the convolution of two-dimensional Doppler-time data requires operations across both dimensions. Moreover, integration of Dop-profiles from multiple antennas yields superior results compared to using a single antenna's Dop-profile. This improvement arises from our dynamic multi-antenna fusion approach, which leverages antenna diversity to obtain more robust features.

Verification of the machine learning model: To verify the effectiveness of the proposed machine learning model, as shown in Test 3-6, we used the extracted Dop-profiles features of multiple antennas as network input and compared the following four different models: CNN+LSTM, ResNet+LSTM, attention-based network Atten-TsNN [46], and our proposed MAF+LSTM. The results show that our proposed method MAF+LSTM outperforms the other methods, demonstrating the effectiveness of our proposed Multi-antennas fusion method weight fusion. In addition, the superior performance of ResNet over traditional CNNs lies in its incorporation of skip or shortcut connections, which ensure that vital features are preserved and propagated even in deep networks by directly copying the input to the output.

7.3 Robustness of mmFinger in the field

To validate the effectiveness of the proposed feature Dop-profile and the proposed multi-antennas combination scheme, we respectively collect different test data under different factors (distances, locations, users, environments, and radar deployments) to test the performance of the model trained by the data in Sec. 7.1. Note that when evaluating each specific factor, we ensure the other factors are the same.

Impact of tapping speed: To assess the impact of tapping speed on recognition accuracy, we invited the same user to perform gestures for all 26 letters at the same distance. We define the tapping speed as the duration of a single tap and denote it as slow (about 0.9 s), normal (about 0.65 s), and fast (about 0.4 s). For instance, the letter "A" with 4 sub-gestures and intervals lasts approximately 2.5 s, 3.5 s, and 4.5 s under three execution speeds. Testing the model trained on Sec. 7.1 with the newly collected dataset, the results are displayed in Fig. 14. A slight decrease in recognition accuracy is observed with increased speed, with an average accuracy of 86%. In practical applications, a medium speed satisfies the majority of users' daily needs.

Impact of the interval between sub-gestures: To assess the impact of sub-gesture intervals (pause time) on recognition accuracy, we manipulated existing data by cropping and compensating to ensure intervals of 0.1 s, 0.2 s, and 0.3 s, constructing a dataset. The reason for not re-collecting data is the difficulty in accurately controlling intervals. For example, assuming a required 0.2 s interval for each sub-gesture, we handle samples as follows: if the interval between sub-gestures is less than 0.2 s, we replicate ground noise samples to ensure a 0.2 s interval between sub-gestures for the letter "A." Conversely, if the interval exceeds 0.2 s, we trim the excess sampling points and concatenate them to the end of the valid sample to maintain a 0.2-second interval between sub-gestures. Testing the model trained on Section 7.1 with the constructed dataset, the results are displayed in Fig. 15. Recognition accuracy is extremely low with a 0.1 s interval because the interval is too small to correctly learn the segmentation of each sub-gesture, leading to increased segmentation errors. When the interval exceeds 0.2 s, recognition accuracy improves to 92%. A 0.2 s interval aligns with the daily needs of the majority of users.

Impact of the interval between characters: To evaluate the effect of character intervals on continuous character recognition performance, we generated 240 samples of 6 words with consistent character intervals (0.2 s, 0.3 s, 0.4 s, 0.5 s). This involved adding or removing ground noise sampling points before and after each character, following a similar procedure to the prior experiment. We test the model trained in Section 7.1 on the constructed samples and the results are illustrated in Fig. 16. We can see that an average accuracy is higher than 93% when the character interval exceeds 0.3s. Greater interval between characters improves the precision of segmentation, resulting in higher accuracy when predicting the final characters and words.

Impact of the distance between radar and hand: As the distance between the radar and the hand increases, the received reflected signal strength decreases and more interference surrounds the signal, making it difficult to identify significant patterns for recognition. In this section, we investigate the impact of distance on recognition accuracy by changing the height of the radar to 25 cm, 35 cm, 45 cm, 1 m, 1.5 m, and 2 m respectively. We use **Dataset 4** to test the trained model with data from each user at each distance. The results in Fig. 17 show that the average accuracy achieved by a single antenna remains above 84.24% across two radar heights of 35 cm and 45 cm, only decreasing by 2% compared with 25 cm, indicating that the Dop-profile is robust to changes in distance. Furthermore, the multi-antenna approach provides an additional average accuracy improvement of 6% and the recognition accuracy of mmFinger surpasses 88% within a 1-meter range, demonstrating that the proposed multi-antenna combination module can further enhance robustness across distances. However, beyond this distance, particularly at 2 meters, there is a notable decline in recognition accuracy. This decline primarily arises due to the increased distance, resulting in a considerably weakened reflected signal from the small finger area that becomes challenging to accurately extract as the distance grows. Despite efforts to eliminate strong reflection signals

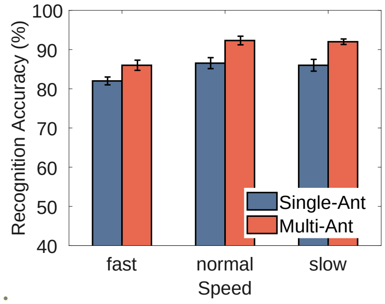


Fig. 14. Impact of tapping speed.

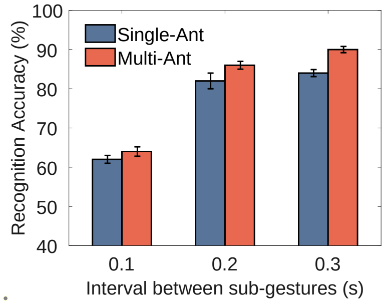


Fig. 15. Impact of interval between sub-gestures.

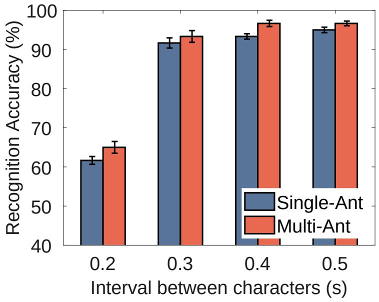


Fig. 16. Impact of interval between characters on word recognition.

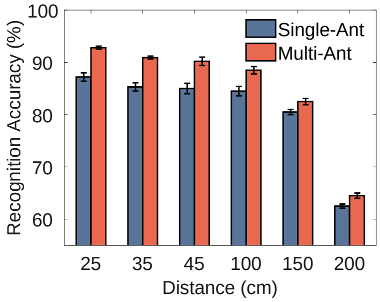


Fig. 17. Impact of different distances.

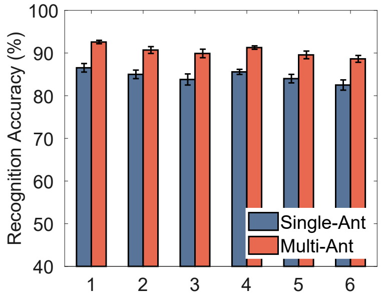


Fig. 18. Impact of finger location.

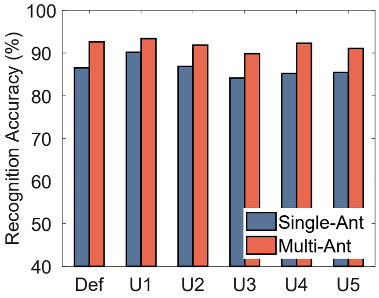


Fig. 19. Impact of user diversity.

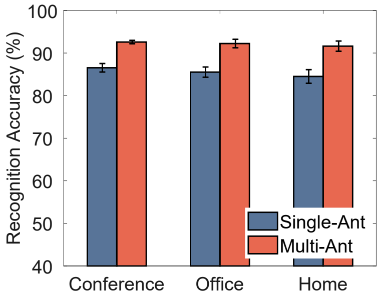


Fig. 20. Impact of environment.

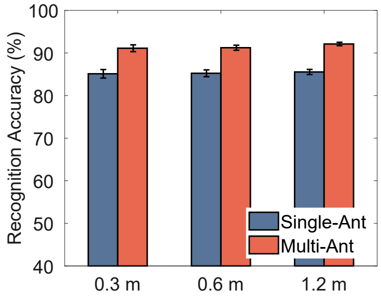


Fig. 21. Impact of surroundings.

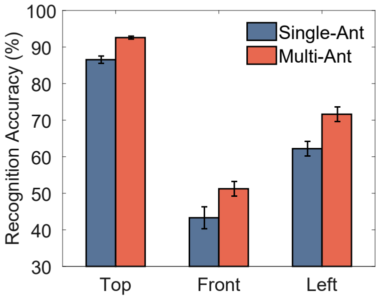


Fig. 22. Impact of radar deployment.

from other objects, such as the human body and table, the extraction of the correct dynamic signal remains challenging due to the weakened nature of the target signal.

Impact of finger location: The location changes of the finger result in variations of the reflection path, causing variations in the reflected signal. In this section, we assess the performance of mmFinger when participants perform gestures at different locations relative to the radar. Specifically, we use **Dataset 5**, which includes data collected at six different locations (Locations 2, 3, 4, 5, and 6 are situated 9 cm, 9 cm, 4 cm, 7 cm, and 7 cm away from Location 1, respectively), to test the trained model. The results are presented in Fig. 18. It is observed that the average accuracy of a single antenna is 84.57% across different locations, indicating the robustness of the Dop-profile to changes in location. Furthermore, the multi-antenna approach improves the average accuracy to 90.44%, demonstrating that the proposed multi-antenna combination module can further enhance robustness across locations.

Impact of user diversity: To account for the fact that users have different hand shapes and varying habits of performing gestures, such as the time interval of sub-gestures and finger movement amplitude, the reflection signals generated can differ. In order to assess the ability of mmFinger to perform gesture recognition across users, we train a recognition model on data from 9 participants in **Dataset 1**, with data from the remaining participant used for testing. As shown in Fig. 19, the average accuracy of 86.38% across 5 users is not significantly decreased compared with the result in Sec. 7.1 (86.54%), indicating the robustness of the extracted Dop-profile to user diversity. Additionally, the multi-antenna approach provides an additional accuracy improvement of $3-5\%$, thus demonstrating the effectiveness of our multi-antenna combination approach.

Impact of environment: The varying multipath conditions in different environments can result in different mixed signals being received by the radar receiver, which in turn can affect the recognition accuracy. To examine this impact, we conduct experiments in three different environments: a conference room (little multipath), office (moderate multipath), and home (severe multipath) to form **Dataset 6**. We then test the trained model using samples from each of

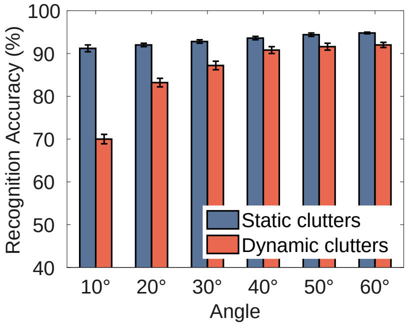


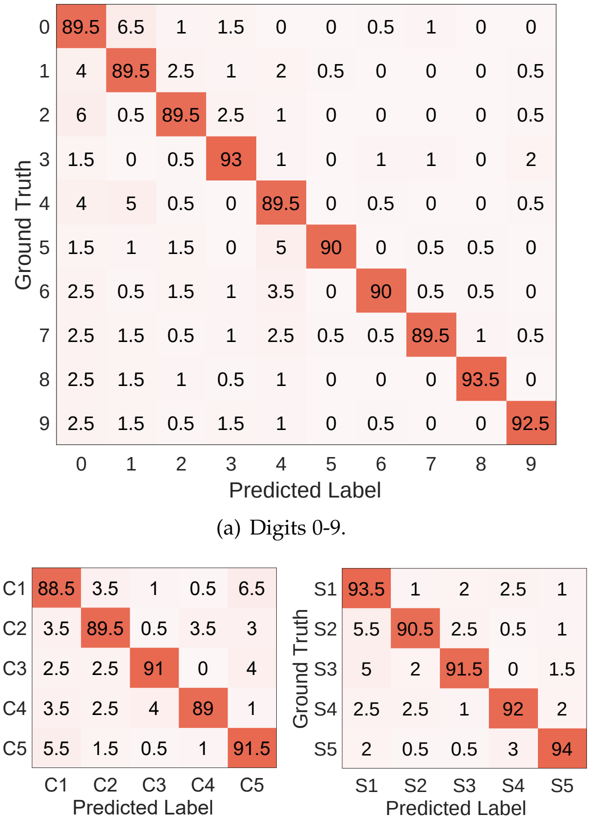
Fig. 23. Recognition accuracy in the interference scene.

the three environments. The results, presented in Fig. 20, indicate a slight decrease in accuracy (only approximately $2-3\%$ for multi-antennas) when the environment changes, indicating mmFinger is robust to environment changes.

Impact of interference from surrounding dynamic and static objects: In practical scenarios, the received signal can be influenced by both the target and surrounding objects. Therefore, in this section, we investigate the impact of surrounding objects' activities on the target's movement detection performance. Firstly, we ask a person to walk around the target at distances of 0.3 m, 0.6 m, and 1.2 m while the target performed letters "A-E" ten times. We test the trained model using the extracted Dop-profiles. The results are depicted in Fig. 21. We observe that the interference caused by surrounding people's movements was minimal (about 2%) across all distances. This is because the mmWave radar can distinguish at least two objects 4 cm apart, allowing us to filter out the interference signals based on range and angle. Our results demonstrate that mmFinger can achieve reliable recognition in practical scenarios where the person causing interference is usually more than 30 cm away from the target user.

Next, we assess the impact of interference from nearby objects (e.g., those at similar distances to a human hand). In the first experiment, we positioned a 2 cm × 6 cm wooden block at multiple sequential locations, each maintaining a constant distance from the radar, equivalent to that of the hand. These positions were incrementally spaced at 10-degree intervals from the hand, moving outward. The results, depicted in Fig. 23, reveal that the recognition accuracy remained above 91% across different object positions (at the same distance as the hand). This indicates that the circle-fitting algorithm effectively mitigates interference from static objects, even when those objects are positioned at the same range as the hand relative to the radar.

In the second experiment, the target user's finger executed gestures directly below the radar, while a volunteer placed one hand in various predetermined positions and tapped their index finger at random intervals. The results, shown in Fig. 23, display that recognition accuracy decreased as the dynamic interference (tapping finger) moved closer to the target hand. To counteract this type of interference, we first separated the signals of the interference object and the target hand by angle. The accuracy dropped sharply when the angle between the interference object and the hand was less than 15 degrees, due to the radar's angular resolution limit of 15 degrees. Within this limit, the radar struggles to distinguish between two objects at similar ranges in the same range bin, leading to a reduction in accuracy. When the angle exceeded 15 degrees, the radar system could effectively separate the signals, mitigating the interference and achieving high recognition accuracy.



(a) Digits 0-9.

(b) User-defined commands.

(c) Unseen sentences.

Fig. 24. The recognition performance on unseen finger gestures.

Impact of radar deployment: To assess the impact of radar deployment on recognition performance, we conduct additional experiments where the radar is placed in front of and to the left of the hand. The data collected from these scenarios are denoted as **Dataset 7** and are used to test the trained model in Sec. 7.1 respectively. The results are presented in Fig. 22. We observe that the "front" deployment yielded the lowest accuracy, followed by the "left" deployment, while the default "top" deployment provides the highest accuracy. This is because when the radar is placed in front of the finger, the reflection area is the smallest and the movement direction of the finger is nearly perpendicular to the path change direction, resulting in a small DFS. Although the reflection area of the "left" deployment is larger than that of the "top" deployment, the DFS remained small since the finger's movement direction is almost perpendicular to the path change direction, as the "front" deployment. In contrast, the "top" deployment has the largest reflection area and the highest mapping value of finger movement velocity in the path direction, resulting in a larger DFS. Therefore, we select the deployment shown in Fig. 12(a) as the default for our evaluation. Further exploration is required to achieve accurate recognition under other deployment scenarios to be suitable for various

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2024.3515044

applications.

7.4 Performance on unseen finger gesture

To assess the scalability of mmFinger to new gestures, we train a recognition model using **Dataset 1** and test it with **Dataset 2** and **Dataset 3**. The results presented in Fig. 24 show an average recognition accuracy of 90.5% and 89% for ten digits and five user-defined commands that do not duplicate existing gesture codes, respectively. These results demonstrate that the system can accurately recognize new gestures without requiring retraining of the network, thus minimizing additional costs.

To evaluate the feasibility of mmFinger for text input, we collect five different user-defined abbreviated sentences. The sentences are: S1 = "I want water," S2 = "Have a good day," S3 = "Need doctor," S4 = "I like it," and S5 = "Thank you." We ask ten users to perform each sentence twenty times and ensure a certain interval between each character. The resulting data is used to test the trained model. As shown in Fig. 24(c), mmFinger achieves an average recognition accuracy of 91.5%, indicating that mmFinger is effective to recognize sentences without pre-segmenting.

7.5 Performance on prediction delay

It is noted that our system processes data offline, with gesture data input into a model trained on a local server for prediction. Our default configuration consists of an IWR1843 and DCA1000, along with an Intel i5-10400F CPU for signal processing and an RTX 2080 Ti GPU for model prediction. Under this setup, the data transmission frame rate is 30 frames per second, resulting in a latency of 33 ms. The signal processing latency is 81 ms, and the model recognition latency is 57 ms. The average total response time required for a test sample with mmFinger is 171 ms. The response time may vary depending on device configurations.

8 Discussion

Although our study has yielded promising results, there is still considerable room for future work and opportunities for further improvement. We discuss a few key points here.

The quality of training samples. Improving the quality of training samples is essential for achieving higher recognition accuracy and enhancing the network's generalization performance. One effective strategy is to ensure that the training dataset includes a diverse range of input conditions, encompassing different user characteristics, sensor positions, and distances from the mmWave radar. A rigorous data screening process should also be employed to eliminate irrelevant or inaccurate data, which can lead to incorrect learning by the network. Additionally, applying data augmentation techniques such as random rotations, translations, and scaling can increase the variability of the training data, further boosting the generalization capability of our network.

Limited sensing range. The current version of mmFinger only works on a limited sensing range due to the relatively small reflection area of the human finger. To overcome this challenge, novel signal processing techniques need to be explored to enhance the Signal-to-Noise Ratio (SNR) of the sensing signal. This may require the development of advanced weak signal extraction and enhancement schemes to improve the sensitivity of the mmWave radar and increase the detection range.

Limited interaction velocity. Compared to gesture or speech-based systems that typically require 1˜2 seconds to complete a single letter, our system may not exhibit a significant advantage in terms of throughput and efficiency, particularly when inputting lengthy commands, which could be perceived as cumbersome and time-consuming. However, the merit of our system lies in its ability to offer more interactive options when individuals experience difficulty or inconvenience in speaking or using their arms or when wearing gloves. Our rhythm-based interactive design allows users the flexibility to set custom finger-tapping patterns for input and maps them to unique functionalities of the desired commands. This differs from traditional classification-based HCI systems, which are constrained by limited and fixed pre-defined patterns of the system.

9 Conclusion

This paper presents mmFinger, an accurate and robust finger gesture recognition system with a single mmWave radar. By carefully designing a robust Dop-profile feature and a multi-antenna adaptive combination scheme to characterize finger gestures' movement, mmFinger is robust to the changes of locations and users. In addition, mmFinger realizes an end-to-end recognition system by encoding one gesture into a sub-gesture sequence, allowing the recognition of user-defined new gestures without requiring specific data for training. Extensive experiments demonstrate that mmFinger achieves reliable, robust, and scalable finger gesture recognition.

References

[1] J. Kim, J. He, K. Lyons, and T. Starner, "The gesture watch: A wireless contact-free gesture based wrist interface," in *2007 11th IEEE International Symposium on Wearable Computers*, 2007, pp. 15–22.

[2] J. Liu, D. Li, L. Wang, and J. Xiong, "Blinklistener: " listen" to your eye blink using your smartphone," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 5, no. 2, pp. 1–27, 2021.

[3] M. Razavi, S. Adavi, M. Z. Alam, D. Mohr, and G. Zachmann, "Innovative and contact-free natural user interaction with cars," in *Proceedings of the International Conference on Virtual and Augmented Reality (GI VR/AR Workshop)*, 2014.

[4] M. Katore and M. R. Bachute, "Speech based human machine interaction system for home automation," in *2015 IEEE Bombay Section Symposium (IBSS)*, 2015, pp. 1–6.

[5] Z. Ren, J. Meng, and J. Yuan, "Depth camera based hand gesture recognition and its applications in human-computer-interaction," in *2011 8th international conference on information, communications & signal processing*. IEEE, 2011, pp. 1–5.

[6] K. H. Shibly, S. K. Dey, M. A. Islam, and S. I. Showrav, "Design and development of hand gesture based virtual mouse," in *2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT)*. IEEE, 2019, pp. 1–5.

[7] R. Li, M. Nguyen, and W. Q. Yan, "Morse codes enter using finger gesture recognition," in *2017 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, 2017, pp. 1–8.

[8] F. Rosado, I. Rumbo, F. Daza, H. Mercado, and D. Mier, "Morse code-based communication system focused on amyotrophic lateral sclerosis patients," in *2016 XXI Symposium on Signal Processing, Images and Artificial Vision (STSIVA)*. IEEE, 2016, pp. 1–6.

[9] P. Kasnesis, C. Chatzigeorgiou, D. G. Kogias, C. Z. Patrikakis, H. V. Georgiou, and A. Tzeletopoulou, "Morse: Deep learning-based arm gesture recognition for search and rescue operations," *arXiv preprint arXiv:2210.08307*, 2022.

[10] F. Schweitzer and A. Campeau-Lecours, "Imu-based hand gesture interface implementing a sequence-matching algorithm for the control of assistive technologies," *Signals*, vol. 2, no. 4, pp. 729–753, 2021.

[11] W. Chen, L. Chen, Y. Huang, X. Zhang, L. Wang, R. Ruby, and K. Wu, "Taprint: Secure text input for commodity smart wristbands," in *Proceedings of the ACM Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2019, pp. 1–16.

[12] W. Chen, L. Chen, M. Ma, F. S. Parizi, S. Patel, and J. Stankovic, "Vifin: Harness passive vibration to continuous micro finger writing with a commodity smartwatch," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 5, no. 1, pp. 1–25, 2021.

[13] K. Niu, F. Zhang, X. Wang, Q. Lv, H. Luo, and D. Zhang, "Understanding wifi signal frequency features for position-independent gesture sensing," *IEEE Transactions on Mobile Computing*, vol. 21, no. 11, pp. 4156–4171, 2021.

[14] J. Zhang, Z. Chen, C. Luo, B. Wei, S. S. Kanhere, and J. Li, "Metaganfi: Cross-domain unseen individual identification using wifi signals," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 6, no. 3, pp. 1–21, 2022.

[15] Y. Zou, J. Xiao, J. Han, K. Wu, Y. Li, and L. M. Ni, "Grfid: A device-free rfid-based gesture recognition system," *IEEE Transactions on Mobile Computing*, vol. 16, no. 2, pp. 381–393, 2016.

[16] Y. Yu, D. Wang, R. Zhao, and Q. Zhang, "Rfid based real-time recognition of ongoing gesture with adversarial learning," in *Proceedings of the 17th Conference on Embedded Networked Sensor Systems*, 2019, pp. 298–310.

[17] V. Becker, L. Fessler, and G. Sörös, "Gestear: combining audio and motion sensing for gesture recognition on smartwatches," in *Proceedings of the 23rd International Symposium on Wearable Computers*, 2019, pp. 10–19.

[18] Z. Li, Z. Lei, A. Yan, E. Solovey, and K. Pahlavan, "Thumouse: A micro-gesture cursor input through mmwave radar-based interaction," in *2020 IEEE International Conference on Consumer Electronics (ICCE)*. IEEE, 2020, pp. 1–9.

[19] H. Liu, Y. Wang, A. Zhou, H. He, W. Wang, K. Wang, P. Pan, Y. Lu, L. Liu, and H. Ma, "Real-time arm gesture recognition in smart home scenarios via millimeter wave sensing," *Proceedings of the ACM on interactive, mobile, wearable and ubiquitous technologies*, vol. 4, no. 4, pp. 1–28, 2020.

[20] W. Jiang, Y. Ren, Y. Liu, Z. Wang, and X. Wang, "Recognition of dynamic hand gesture based on mm-wave fmcw radar micro-doppler signatures," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 4905–4909.

[21] W. Wang, A. X. Liu, and K. Sun, "Device-free gesture tracking using acoustic signals," in *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*, 2016, pp. 82–94.

[22] K. Ali, A. X. Liu, W. Wang, and M. Shahzad, "Keystroke recognition using wifi signals," in *Proceedings of the 21st annual international conference on mobile computing and networking*, 2015, pp. 90–102.

[23] J. Lien, N. Gillian, M. E. Karagozler, P. Amihood, C. Schwesig, E. Olson, H. Raja, and I. Poupyrev, "Soli: Ubiquitous gesture sensing with millimeter wave radar," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 4, pp. 1–19, 2016.

[24] S. M. Kwon, S. Yang, J. Liu, X. Yang, W. Saleh, S. Patel, C. Mathews, and Y. Chen, "Hands-free human activity recognition using millimeter-wave sensors," in *2019 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*. IEEE, 2019, pp. 1–2.

[25] G. Li, Z. Zhang, H. Yang, J. Pan, D. Chen, and J. Zhang, "Capturing human pose using mmwave radar," in *2020 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*. IEEE, 2020, pp. 1–6.

[26] A. Sengupta, F. Jin, and S. Cao, "Nlp based skeletal pose estimation using mmwave radar point-cloud: A simulation approach," in *2020 IEEE Radar Conference (RadarConf20)*. IEEE, 2020, pp. 1–6.

[27] A. Sengupta, F. Jin, R. Zhang, and S. Cao, "mm-pose: Real-time human skeletal posture estimation using mmwave radars and cnns," *IEEE Sensors Journal*, vol. 20, no. 17, pp. 10 032–10 044, 2020.

[28] Y. Wang, W. Wang, M. Zhou, A. Ren, and Z. Tian, "Remote monitoring of human vital signs based on 77-ghz mm-wave fmcw radar," *Sensors*, vol. 20, no. 10, p. 2999, 2020.

[29] H. Liu, Y. Wang, A. Zhou, H. He, and H. Ma, "Real-time arm gesture recognition in smart home scenarios via millimeter wave sensing," *Proceedings of the ACM on Interactive Mobile Wearable and Ubiquitous Technologies*, vol. 4, no. 4, pp. 1–28, 2020.

[30] 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, p. 27, 2021.

[31] H. Kong, X. Xu, J. Yu, Q. Chen, C. Ma, Y. Chen, Y.-C. Chen, and L. Kong, "m3track: mmwave-based multi-user 3d posture tracking," in *Proceedings of the 20th Annual International Conference on Mobile Systems, Applications and Services*, 2022, pp. 491–503.

[32] Y. Hu, B. Wang, C. Wu, and K. R. Liu, "mmkey: Universal virtual keyboard using a single millimeter-wave radio," *IEEE Internet of Things Journal*, vol. 9, no. 1, pp. 510–524, 2021.

[33] 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.

[34] Z. Chen, T. Zheng, C. Cai, and J. Luo, "Movi-fi: Motion-robust vital signs waveform recovery via deep interpreted rf sensing," in *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*, 2021, pp. 392–405.

[35] J. Gong, X. Zhang, K. Lin, J. Ren, Y. Zhang, and W. Qiu, "Rf vital sign sensing under free body movement," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 5, no. 3, pp. 1–22, 2021.

[36] C. Jiang, J. Guo, Y. He, M. Jin, S. Li, and Y. Liu, "mmvib: micrometer-level vibration measurement with mmwave radar," in *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, 2020, pp. 1–13.

[37] Y. Yang, H. Xu, Q. Chen, J. Cao, and Y. Wang, "Multi-vib: Precise multi-point vibration monitoring using mmwave radar," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 6, no. 4, pp. 1–26, 2023.

[38] K. Niu, F. Zhang, Y. Jiang, J. Xiong, Q. Lv, Y. Zeng, and D. Zhang, "Wimorse: A contactless morse code text input system using ambient wifi signals," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 9993–10 008, 2019.

[39] J. M. Statland, R. J. Barohn, A. L. McVey, J. S. Katz, and M. M. Dimachkie, "Patterns of weakness, classification of motor neuron disease, and clinical diagnosis of sporadic amyotrophic lateral sclerosis," *Neurologic clinics*, vol. 33, no. 4, pp. 735–748, 2015.

[40] B. INSIDER, *Paralyzed patients can control computers just by moving their eyes, thanks to this free software*, September 2015. [Online]. Available: https://www.businessinsider.com/an-eye-tracking-interface-helps-als-patients-use-computers-2015-9

[41] H. Rohling, "Radar cfar thresholding in clutter and multiple target situations," *IEEE transactions on aerospace and electronic systems*, no. 4, pp. 608–621, 1983.

[42] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional lstm and other neural network architectures," *Neural networks*, vol. 18, no. 5-6, pp. 602–610, 2005.

[43] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 369–376.

[44] TensorFlow, "tf.keras.optimizers.adam," March 2023. [Online]. Available: https://www.tensorflow.org/api_docs/python/tf/keras/optimizers/Adam

[45] TI, "Iwr1843," January 2022. [Online]. Available: https://www.ti.com/product/AWR1843

[46] B. Jin, Y. Peng, X. Kuang, Z. Zhang, Z. Lian, and B. Wang, "Robust dynamic hand gesture recognition based on millimeter wave radar using atten-tsnn," *IEEE Sensors Journal*, vol. 22, no. 11, pp. 10 861–10 869, 2022.