2024 IEEE 44th International Conference on Distributed Computing Systems (ICDCS)

GesturePrint: Enabling User Identification for mmWave-based Gesture Recognition Systems

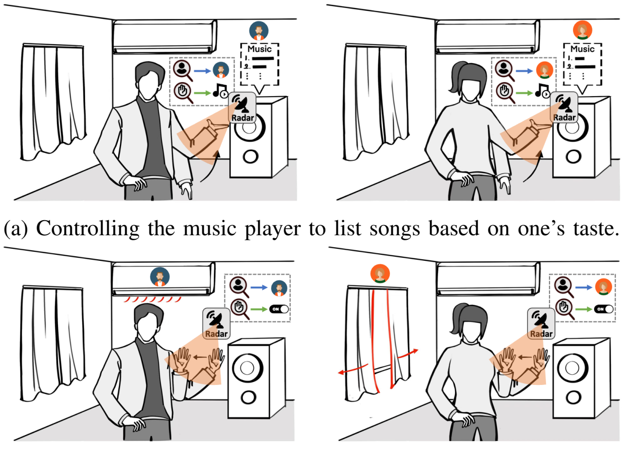
Lilin Xu*, Keyi Wang†, Chaojie Gu*, Xiuzhen Guo*, Shibo He*, Jiming Chen*
*Zhejiang University, †University of California San Diego
lilinxu@zju.edu.cn, kew035@ucsd.edu, {gucj, guoxz, s18he, cjm}@zju.edu.cn

***Abstract*—The millimeter-wave (mmWave) radar has been exploited for gesture recognition. However, existing mmWave-based gesture recognition methods cannot identify different users, which is important for ubiquitous gesture interaction in many applications. In this paper, we propose *GesturePrint*, which is the first to achieve gesture recognition and gesture-based user identification using a commodity mmWave radar sensor. *GesturePrint* features an effective pipeline that enables the gesture recognition system to identify users at a minor additional cost. By introducing an efficient signal preprocessing stage and a network architecture GesIDNet, which employs an attention-based multilevel feature fusion mechanism, *GesturePrint* effectively extracts unique gesture features for gesture recognition and personalized motion pattern features for user identification. We implement *GesturePrint* and collect data from 17 participants performing 15 gestures in a meeting room and an office, respectively. *GesturePrint* achieves a gesture recognition accuracy (GRA) of 98.87% with a user identification accuracy (UIA) of 99.78% in the meeting room, and 98.22% GRA with 99.26% UIA in the office. Extensive experiments on three public datasets and a new gesture dataset show *GesturePrint*'s superior performance in enabling effective user identification for gesture recognition systems.**

I. Introduction

Nowadays, the mmWave radar has received increasing attention from industry and academia because of its low power, high spatial resolution, and robustness to temperature and lighting conditions. The mmWave radar has empowered plenty of applications in autonomous driving [1], [2], human localization and tracking [3], [4], activity recognition [5], [6] and health care [7]–[9]. Recently, there has been a trend of utilizing mmWave radars to implement gesture recognition systems, which enables ubiquitous gesture interaction in a broad spectrum of applications, including gaming control, Internet of Things (IoT), and virtual reality (VR). Compared with traditional methods [10], [11] based on wearable devices, mmWave radars enable touchless gesture recognition without causing discomfort or extra burden to users. In addition, mmWave-based approaches are more privacy-preserving than visual-based solutions [12]. As for other RF-based solutions like using WiFi [13], [14], mmWave sensing operates at extremely high frequency (EHF) bands, and it has a higher spatial resolution for fine-grained motion detection.

In the last several years, several mmWave-based gesture recognition solutions [15]–[18] have been proposed. Although existing solutions can accurately recognize predefined gestures performed by different users, none of them can identify the user who performs the gestures. In practice, the capability of user identification can significantly improve the user experience in interacting with smart devices. For example, as shown in Fig. 1a, when a user performs a predefined gesture to turn on a music player using a mmWave-based controller, the controller obtains the user's identity and instructs the player to list songs according to the user's taste. Also, the user can personalize the meaning of gestures, e.g., waving one hand from left to right to open/close the curtain or decrease/increase the air conditioning temperature as demonstrated in Fig. 1b.



(a) Controlling the music player to list songs based on one's taste.

(b) Personalizing the meaning of gestures to operate home devices.

Fig. 1: Potential applications of mmWave-based gesture recognition systems with user identification capability. The identification capability can significantly improve the user experience in interacting with smart devices.

While there exist some gesture-based user identification solutions that utilize other modalities such as video [12], IMU [19], and WiFi [20], they cannot be directly applied to mmWave sensing due to the differences in data format and feature representation across modalities. Additionally, the mmWave radar has unique application scenarios that differ from other modalities. Thus, solutions specially designed for mmWave-based point clouds are essential to fully exploit the potential of gesture-based interaction with the mmWave radar.

To unleash the potential of mmWave-based gesture recognition systems, we propose *GesturePrint*, an effective one-stop solution to enable the gesture recognition systems to identify users at a minor additional cost. *GesturePrint* first obtains

2575-8411/24/$31.00 ©2024 IEEE
DOI 10.1109/ICDCS60910.2024.00103

point clouds related to objects in the environment by the radar device. It then segments the gesture motions utilizing a parameter-adaptive sliding window method, subsequently removing the outlier points that are distinguished from the target points reflected from people. With the preprocessed data, *GesturePrint* can accurately recognize predefined gestures and identify the user who performs the gestures based on our specially designed network architecture GesIDNet.

To achieve this, there are several practical challenges in designing *GesturePrint*, as summarized below.

Effective Gesture Capture: Reliable gesture segmentation is essential, as it influences the amount of relevant information and noise preserved for subsequent gesture recognition and user identification. However, determining the beginning and end of a gesture is challenging due to the sparsity and irregularity of point clouds obtained by the radar. To address this issue, *GesturePrint* employs a parameter-adaptive sliding window method to effectively detect the start and the end of a gesture motion. Furthermore, in real-world scenarios, there are many noise points from the surroundings. Even in the same environment, the distribution of noise points changes considerably when the surrounding objects move slightly. This adversely affects the processing of gesture-related points and consequently degrades the system's performance on gesture recognition and user identification. To mitigate this issue, *GesturePrint* eliminates outlier points unrelated to the user throughout the entire gesture motion as much as possible.

Effective Feature Extraction: On the one hand, achieving a one-stop solution for recognition and identification requires that the method can obtain effective information from the point cloud data to distinguish distinct gestures and different users, respectively. On the other hand, it is challenging to extract effective and reliable gesture features and user-personalized motion pattern features from the sparse point clouds captured by the mmWave radar sensor. To solve the problems above, we introduce a network architecture, GesIDNet, which is specially designed to accommodate the characteristics of gesture point clouds. GesIDNet features an attention-based multilevel feature fusion module that adaptively fuses low-level features and high-level features extracted from the sparse point clouds to obtain effective features.

Robust Feature Learning: There are inevitable differences among multiple repetitions of the same gesture performed by the same user, e.g., different motion speeds and different distances from the mmWave radar sensor. It is challenging for the system to stay robust to these differences. The design of GesIDNet contributes to mitigating the impact brought by this issue. Besides, *GesturePrint* employs data augmentation during the training phase to enhance the system's robustness.

In *GesturePrint*, we utilize a commodity mmWave radar sensor and a laptop for data collection and inference. We use a back-end server to train the models. We design a signal processing pipeline to segment gestures from the collected data. GesIDNet then extracts effective features from the segmented gesture data for gesture recognition and user identification. We conduct extensive experiments on three public datasets and a new gesture dataset to evaluate *GesturePrint* and compare its performance with other state-of-the-art methods. While existing mmWave-based point cloud gesture datasets only contain self-defined gestures, we build a new dataset of standard gestures in ASL (American Sign Language). Experimental results show that *GesturePrint* achieves an overall performance of 98.87% accuracy for gesture recognition and 99.78% accuracy for gesture-based user identification with 15 gestures and 17 users in the meeting room environment while 99.88% for recognition and 97.60% for identification with 5 gestures and 32 users. Besides, it achieves an average result of 0.75% EER (Equal Error Rate) for user identification across all datasets. We further evaluate *GesturePrint* under different experimental settings, and all the results indicate that it is effective for both gesture recognition and user identification. In summary, our major contributions are summarized as follows:

- To the best of our knowledge, *GesturePrint* is the first solution that augments mmWave-based gesture recognition systems with user identification capability.
- We introduce an efficient signal preprocessing stage and a specially designed network architecture GesIDNet to achieve effective feature extraction from sparse point clouds captured by the radar. The features include unique gesture features for gesture recognition and personalized motion pattern features for user identification.
- We build a mmWave-based gesture dataset, including 9,332 samples from 17 participants performing 15 ASL gestures in two different environments. Besides, we conduct extensive experiments to evaluate *GesturePrint* on three public gesture recognition datasets. The results demonstrate *GesturePrint*'s superior performance in enabling effective user identification for gesture recognition systems.

The rest of this paper is organized as follows. §II reviews related studies. §III presents the preliminary study. §IV introduces the detailed design of *GesturePrint*. §V and §VI present the implementation and evaluation results. §VII discusses some related issues. §VIII concludes this work.

II. Related work

A. mmWave-based Gesture Recognition

mmWave-based gesture recognition demonstrates great potential in interaction applications [15]–[18], as mmWave radars provide fine-grained resolution and easy deployment. Due to the collected data with high-dimensional features, existing methods mainly employ neural networks for effective feature extraction. According to the input data format, existing approaches can be broadly divided into *signal map-based* and *point cloud-based*.

Signal map-based Approaches: mmASL [21] proposes a multi-task deep learning architecture to extract gesture domain features from the spectrograms. DI-Gesture [22] uses a dynamic window mechanism and a network combined by CNN and LSTM to achieve domain-independent and real-time gesture recognition. RadarNet [23] proposes a novel CNN architecture dealing with range-Doppler maps to recognize swipe gestures. M-gesture [24] designs a compact CNN architecture to suppress the influence of different users, and achieves person-independent finger gesture recognition.

Point cloud-based Approaches: Due to advancements in handling unordered point clouds, point cloud-based methods have recently become a research trend in mmWave-based gesture recognition. mHomeGes [15] and mTransSee [16] both convert point clouds into the concentrated position-doppler profile to emphasize the positional relationship and speed differences among points, and use designed convolutional networks to extract gesture features. Pantomime [17] combines PointNet++ and LSTM to extract spatio-temporal features from mmWave radar point clouds, while Tesla-Rapture [18] utilizes a temporal K-NN module based on graph convolution.

Different from existing studies, apart from extracting unique gesture features for gesture recognition, *GesturePrint* can effectively capture personalized motion pattern features from the gesture point clouds by employing an efficient signal preprocessing stage and a specially designed network architecture. In this way, it empowers mmWave-based gesture recognition systems with the capability of user identification.

B. *mmWave-based User Identification*

As mmWave radars can capture human biometric information with fine-grained spatial resolution, some mmWave-based user identification methods have been proposed in recent years.

VocalPrint [25] uses a mmWave radar sensor to capture skin-reflect signals around the near-throat region during speech, while the work in [26] utilizes both vocal cord vibration and lip motion. Moreover, M-Auth [27] achieves multi-user authentication by capturing unique breathing patterns from radar signals. Additionally, researchers have explored gait-based identification. MU-ID [28] identifies multiple users simultaneously by capturing gait biometrics. mID [3] uses 3-D voxel grids and a deep recurrent network for identification and tracking. mmGait [29] proposes a CNN network directly using point clouds as input. SRPNet [30] employs a network that combines PointNet and Bi-LSTM to extract spatio-temporal information from point clouds.

However, these mmWave-based identification methods cannot be directly applied for user identification with gestures due to the heterogeneity in different motion patterns. Integrating these methods into existing gesture recognition systems to achieve user identification will add an extra motion burden to the user. To the best of our knowledge, *GesturePrint* is the first work that explores the feasibility of identifying users based on their gesture motions to further unleash the potential of gesture-based interaction with the mmWave radar at a minor extra cost. This one-stop solution for recognizing gestures and identifying users can enhance the applications of mmWave-based gesture interactions.

III. Preliminaries

The mmWave radar captures signals reflective of gesture motions as users perform gestures. As personal behavioral traits can serve as biometrics [31], we utilize a mmWave radar sensor to capture such characteristics, including behavior manners and personalized unconscious motion styles, from gesture motions for user identification.

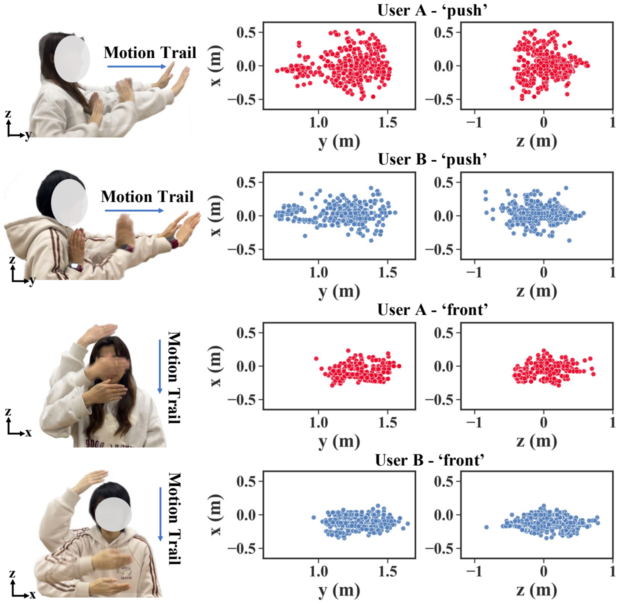


Fig. 2: The visualization of gesture point clouds obtained from User A and User B when they perform ASL gestures.

mmWave point clouds are generated through a series of signal processing steps applied to the frequency modulated continuous wave (FMCW) signal reflected by objects, which contains information about the range, velocity, and angle of the objects. The processing steps include Range Fast-Fourier Transform (FFT), Doppler FFT, Constant False Alarm Rate (CFAR), and Angle FFT. The resulting point clouds contain crucial information about the objects, which offers insights into their shapes and movements. Fig. 2 shows the visualization of point clouds captured from User A and User B when they performed ASL signs 'push' and 'front'. These two users have similar body shapes, with a height of around 160 cm and a weight of around 48 kg. The gesture point clouds exhibit characteristics that can be utilized for gesture recognition and user identification, respectively. On the one hand, point clouds can demonstrate different shapes and movements of different ASL sign gestures. On the other hand, it is observed that gesture point clouds can differ in space and time between the same gesture performed by different users, such as point number, coverage, and density. These differences are mainly caused by individual variations in arm length, motion speed, range of motion, and even implicit motion habits. For instance, as shown in Fig. 2, the range of User B's point clouds on the x-axis is narrower than that of User A, implying that User B has a more limited range of motion on the x-axis when executing gestures. However, compared with the differences in point clouds resulting from distinct gestures, the differences between point clouds of the same gesture performed by different users

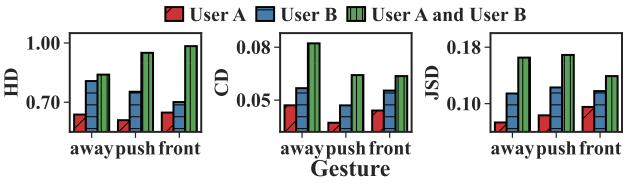


Fig. 3: Differences among point clouds are measured by three metrics: Hausdorff distance (HD), Chamfer distance (CD), and Jensen-Shannon divergence (JSD).

are not as pronounced, indicating that gesture-based user identification is a more challenging task.

To further study the feasibility of identifying users based on gesture point cloud data captured by the radar, we utilize three metrics, i.e., Hausdorff distance (HD) [32], Chamfer distance (CD) [33], and Jensen-Shannon divergence (JSD) [34], to evaluate gesture points clouds. The HD measures the extent to which each point in one point cloud lies near some point in another point cloud; the CD measures the average bidirectional closest-point distance between two distinct point clouds; the JSD measures the degree to which points in one point cloud tend to occupy similar locations as those of another point cloud. Let $\boldsymbol{G} = \{g_1, g_2, ..., g_{N_g}\}$ denote N_g different gestures. For each user, there are N_c point clouds $\boldsymbol{C}^{g_i} = \{c_1^{g_i}, c_2^{g_i}, ..., c_{N_c}^{g_i}\}$ for gesture $g_i \in \boldsymbol{G}$ after the preprocessing stage (see §IV-B for more details). Then, we measure the difference between point clouds as

$$d(g_i) = \frac{\sum_{m=1}^{N_{c2}} \sum_{n=1}^{N_{c1}} D(c_n^{g_i}, c_m^{g_i})}{N_{c1} N_{c2}}, \tag{1}$$

where $c_n^{g_i} \in \boldsymbol{C}_1^{g_i}$, $c_m^{g_i} \in \boldsymbol{C}_2^{g_i}$ and $c_n^{g_i} \neq c_m^{g_i}$. $D(\cdot)$ denotes the metric utilized for measurement, and it can be either $HD(\cdot)$ or $CD(\cdot)$ or $JSD(\cdot)$. When measuring the difference between the same gesture performed by the same user, $\boldsymbol{C}_1^{g_i}$ and $\boldsymbol{C}_2^{g_i}$ in Eq. 1 are from the same user; when measuring the differences between the same gesture performed by different users, $\boldsymbol{C}_1^{g_i}$ and $\boldsymbol{C}_2^{g_i}$ are from different users.

For a fair comparison, both User A and User B are instructed to execute 3 ASL gestures ('away', 'push', and 'front'), with each gesture being performed 10 times. The HD, CD, and JSD results of the gesture point clouds are displayed in Fig. 3. These results demonstrate that, for the same ASL gesture, the differences between gesture point clouds from different users are more noticeable than those from the same user. This suggests that gesture point clouds captured by the radar contain information that can be used to differentiate between users when they perform gesture motions. Although the mmWave radar can sense these differences, it is difficult to directly distinguish distinct gestures and different users from the point clouds since the point clouds are abstract and contain high-dimensional features, which cannot be fully exploited through only calculating distances and divergence between the point clouds. Thus, it is desirable to design an efficient data preprocessing method and an effective network architecture, which can extract unique gesture features and features containing personalized motion patterns from point clouds, to effectively achieve both gesture recognition and user identification.

IV. GesturePrint Design

In this section, we first provide an overview of *GesturePrint*. Then, we present the detailed design of *GesturePrint*.

A. System Overview

GesturePrint is designed to extract effective features regarding specific gestures and users from sparse point clouds captured by the mmWave radar. Fig. 4 demonstrates the system pipeline of *GesturePrint*, which has two major stages with six modules. The data preprocessing stage includes *point clouds capture*, *gesture segmentation*, *noise canceling*, and *data augmentation*. The classification stage includes *gesture recognition* and *user identification*. *GesturePrint* works with a commodity mmWave radar sensor, after obtaining the points converted from signal data through the radar, it segments out gestures from temporal point cloud frames by using an adaptive sliding window method. After gesture segmentation, *GesturePrint* discards outlier noise points that are not reflected from the human body. After noise canceling, the points captured by the radar in the whole gesture process are aggregated as the gesture point cloud, which is then fed into GesIDNet, a specially designed network architecture, for gesture recognition. With the recognition result, GesIDNet is employed to further identify the user performing the gesture with the gesture-corresponding recognition model. Finally, the gesture and the user are both inferred by *GesturePrint*. In particular, during training, we augment the data by adding some random jitters to the points. Although the models for the two tasks are trained individually, and the features extracted for the two tasks are different, *GesturePrint* does not need any extra data compared with the stand-alone gesture recognition task. We use the same data to dig for more information from another dimension for user identification.

B. Data Preprocessing Stage

In this stage, *GesturePrint* processes the raw data captured by the radar to obtain gesture point clouds, which are then utilized as the input in the subsequent classification stage.

Point Clouds Capture: The transformation from raw signal data reflected by objects in the environment into point cloud data format is done by the mmWave radar device. Besides, to mitigate the influence of reflections from surroundings in the environment that are unrelated to the user's activities, we enable the static clutter removal function before object detection. In this way, the objects detected at the zero Doppler velocity bins by the device can be discarded.

Gesture Segmentation: This module segments complete gesture motions from the captured point cloud data. Unlike DI-Gesture [22] segmenting gestures by applying a dynamic window mechanism to DRAI (Dynamic Range Angle Image) captured by the radar, we segment gestures based on radar

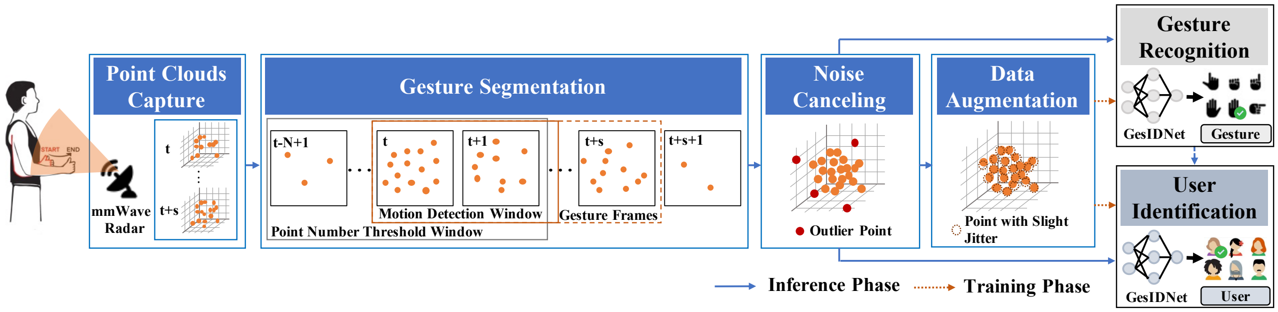


Fig. 4: Overview of *GesturePrint*: a system that enables mmWave-based gesture recognition and gesture-based user identification, comprising the data preprocessing and classification stages.

point clouds. We exploit the number of points in each frame and design a parameter-adaptive sliding window method to detect which frame a gesture motion starts and ends. *GesturePrint* first calculates the cumulative distribution density of points over N frames and obtains a dynamic point number threshold P_{Thr} based on the distribution. It then uses a sliding motion detection window of length n with P_{Thr} to determine whether the current frame contains gesture motion or not. Once a frame is determined as a motion frame, *GesturePrint* checks the motion frame count in the sliding window. If the number exceeds the minimum frame threshold F_{Thr}, the frame is regarded as the start of the motion, and the subsequent frames are added to the motion frame group until all frames in the sliding window are determined as static frames, signaling the end of the gesture motion. The points in the motion frame group are then aggregated.

Noise Canceling: Although some noise reflected from the environment can be removed by enabling the static clutter removal function during data collection, the remaining points are not all from the user's gesture motion. There are still some noise points caused by reflectors' subtle movement or signal multipath reflection. These noise points indeed adversely affect gesture recognition and user identification. To better obtain user-related points and mitigate the influence of noise, *GesturePrint* apply DBScan, a density-aware algorithm, to cluster points based on pairwise distances within the whole point cloud. We set the parameters for DBScan to get point clusters, including the maximum distance between pair points D_{max}, and the minimum point number of a point cluster N_{min}. Among all the clusters obtained through DBScan, the cluster containing most of the points is retained as the main cluster, while others are discarded. This main cluster, related to the human body and regarded as the gesture point cloud, is then prepared for being fed into GesIDNet.

Data Augmentation: To enhance the system's robustness, data augmentation is employed during the training phase. We introduce a random subtle displacement j to each point p in the gesture point cloud P. This process is repeated to augment the data three times. Displacements for each gesture point cloud are generated using a Gaussian distribution with the mean $\mu = 0$ and the standard deviation $\sigma = 0.02$.

C. *Classification Stage*

In this stage, we propose GesIDNet (as demonstrated in Fig. 5) to address recognition and identification tasks based on the same gesture point cloud obtained in the data preprocessing stage. For the gesture recognition task, GesIDNet can effectively extract unique features for different gestures. For the user identification task, GesIDNet can extract the high-dimensional features related to personalized motion patterns, which represent the motion styles of different users. The same data are fed to GesIDNet with different labels, i.e., gesture labels and user identity labels, to train gesture recognition models and user identification models, respectively.

At runtime, there are two identification modes in our system, i.e., *serialized mode* and *parallel mode*. The serialized mode identifies users with the result of gesture recognition. The gesture point cloud obtained in the preprocessing stage is fed into the GesIDNet recognition model to recognize the gesture. Based on the inferred gesture, *GesturePrint* selects the corresponding identification model for further user identification. In contrast, the parallel mode conducts gesture recognition and user identification separately. The main difference between these two modes is that during training time, the user identification model is trained for every gesture separately or trained with all predefined gestures. According to the overall performance of these two modes (see §VI for more details) and the capability of handling random gestures and unauthorized people, we set the serialized mode as our default identification mode. Finally, with the outputs of GesIDNet, *GesturePrint* achieves gesture recognition and user identification. The design details of GesIDNet are described below.

Fig. 5 demonstrates the design of GesIDNet. PointNet++ [35] can extract details and features from data structured in point cloud format. However, while PointNet++ is typically employed with large-scale dense point clouds, the gesture point clouds obtained by the mmWave radar are usually sparse. Thus, to effectively extract features from the sparse gesture point clouds, we adopt the set abstraction block of PointNet++ and further design a multilevel feature fusion module with an attention mechanism.

On the one hand, since the sparse points in each frame captured by the radar provide insufficient information for fine-

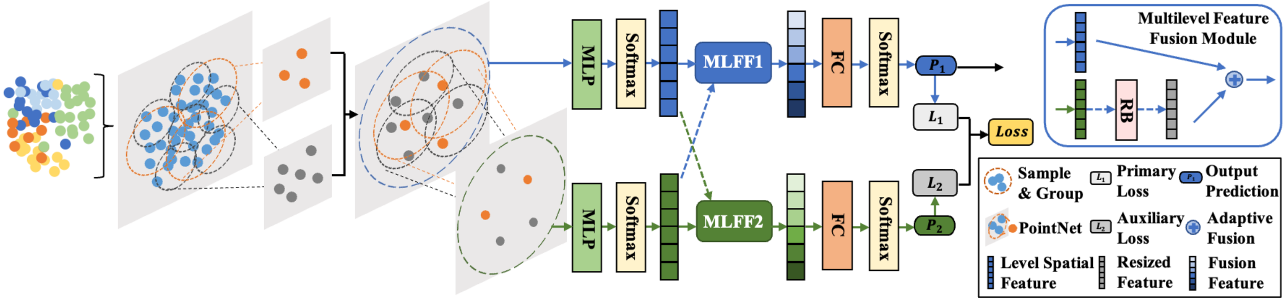


Fig. 5: The architecture of GesIDNet. MLP: Multilayer Perceptron, FC: Fully Connected Layer, RB: Resizing Block.

grained local features extraction [17], we aggregate points captured by the radar in the whole gesture process following the gesture segmentation module, and the aggregated points are further processed as the input of GesIDNet. Although we aggregate all the frames together, owing to noise canceling, point clouds in every frame are usually in a relatively concentrated place in the space. Therefore, by extracting and combining local features, a part of the information in the frame dimension is still preserved, which facilitates the effective extraction of gesture features and user-personalized motion pattern features. GesIDNet employs the set abstraction block of PointNet++ to extract local spatial features at different scales from the aggregated gesture point clouds, and these multiscale local features are combined for extracting higher-level features. To be specific, one single set abstraction block can be described as follows: with a set of unordered points $\{p_1, p_2, ..., p_N\}$ as the input, GesIDNet samples n_i points from the point set, and groups the nearest m_i points within the radius d_i around them as representation points $\{p_1^i, p_2^i, ..., p_{n_i}^i\}$ of local regions, then it uses multilayer perceptron (MLP) to extract features. n_i, m_i, and d_i are hyperparameters of the set abstraction block i, and local features $\boldsymbol{f}_i$ of different scales are combined as $\boldsymbol{f}_s$, which is the output of the block.

On the other hand, the aggregated gesture point clouds comprise an unordered set of points with varying numbers and strong spatio-temporal correlation. To exploit the data characteristics, we introduce an attention-based multilevel feature fusion module to adaptively combine low-level features and high-level features extracted from the point clouds. With the module, GesIDNet adaptively assigns large weights to effective features for gesture recognition and user identification. To be specific, GesIDNet extracts level feature $\boldsymbol{F}$ from $\boldsymbol{f}_s$ by grouping all the representation points and applying MLP. There are two feature levels, i.e., l_1 (low-level) and l_2 (high-level), in GesIDNet. The sizes of feature vectors in each level are different, so before fusing features of different levels, in each level, GesIDNet resizes the other level's feature vectors to match the feature vector size at the current level by using a resizing block. The block includes a linear layer followed by a ReLu layer. Let $\boldsymbol{F}^k$ denote the combined feature of point set abstraction at level k, $\boldsymbol{F}^{l->k}$ denote the feature $\boldsymbol{F}^l$ resized from level l to level k to match the size of the combined feature at level k. The fusion feature $\boldsymbol{Y}^k$ of $\boldsymbol{F}^{l->k}$ and $\boldsymbol{F}^k$ at level k can be described as

$$\boldsymbol{Y}^k = S(\boldsymbol{F}^{l->k}) \cdot \boldsymbol{F}^{l->k} + S(\boldsymbol{F}^k) \cdot F^k, \tag{2}$$

where $S(\boldsymbol{F}^{l->k})$ and $S(\boldsymbol{F}^k)$ refer to the adaptive weights of different level features at level k. $S(\cdot)$ denotes a softmax function, and $S(\boldsymbol{F}^{l->k})$ and $S(\boldsymbol{F}^k)$ can be expressed by

$$\begin{aligned} S(\boldsymbol{F}^{l->k}) &= \frac{e^{g(\boldsymbol{F}^{l->k})}}{e^{g(\boldsymbol{F}^{l->k})} + e^{g(\boldsymbol{F}^k)}}, \\ S(\boldsymbol{F}^k) &= \frac{e^{g(\boldsymbol{F}^k)}}{e^{g(\boldsymbol{F}^{l->k})} + e^{g(\boldsymbol{F}^k)}}. \end{aligned} \tag{3}$$

In Eq. 3, $S(\boldsymbol{F}^{l->k}) + S(\boldsymbol{F}^k) = 1$, $S(\boldsymbol{F}^{l->k})$, $S(\boldsymbol{F}^k) \in [0, 1]$, and $g(\cdot)$ denotes a convolutional layer. In this way, the weights of different level spatial features can be adaptively learned.

To fully exploit the fusion results at each level, we adopt the idea of auxiliary loss [36]. The fusion feature $\boldsymbol{Y}^k$ at level k is then fed to a couple of FC (fully connected) layers followed by final classifiers, and the number of FC layers depends on the level. In this way, GesIDNet produces two classification prediction results P_1 and P_2 at two feature levels l_1 and l_2, respectively. We regard P_1 as the primary prediction result and the cross-entropy loss calculated from it is called primary loss L_1 while regarding P_2 as the auxiliary result with a loss called auxiliary loss L_2. During training, we add the auxiliary loss to the primary loss as the final loss, while in the inference phase, we just use the primary classification result as the final prediction result of GesIDNet.

To demonstrate the effectiveness of GesIDNet in exploiting the data characteristics of gesture point clouds for gesture recognition and user identification, we visualize features extracted by GesIDNet with t-SNE [37]. As shown in Fig. 6 (a), for gesture recognition, low-level gesture features and high-level gesture features do not exhibit the same clear clusters for different gestures as the fusion features, indicating that the multilevel feature fusion further extracts effective unique features for different gestures by adaptively combining the low-level and high-level features. Besides, as shown in Fig. 6 (b-1) & (b-2), compared with gesture recognition, the low-level and high-level features extracted for user identification exhibit much lower clustering effect, indicating that features related to personalized motion patterns are more difficult

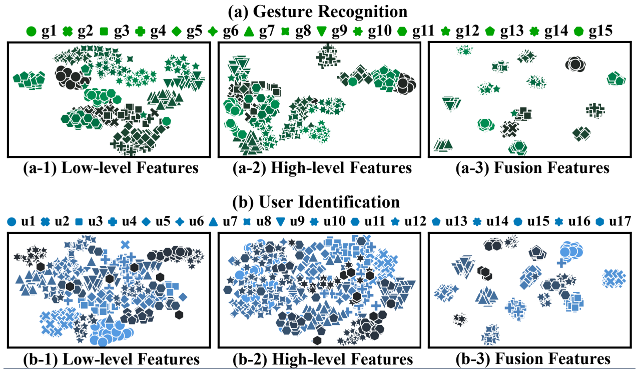


Fig. 6: The visualization of features extracted by GesIDNet for gesture recognition and user identification, respectively. Different shapes denote different gestures/users.

to extract. However, after multilevel feature fusion, gesture samples belonging to the same user form into a clear cluster as demonstrated in Fig. 6 (b-3), indicating that GesIDNet achieves effective feature extraction for user identification.

V. Implementation

Hardware. *GesturePrint* consists of a commodity mmWave radar sensor, a laptop, and a back-end server. We use the IWR6843 antenna-on-package (AoP) evaluation module (EVM) [38] for data collection. The radar is mounted at a fixed height of 1.25 m with a tripod. As shown in Fig. 7, the collected data is transmitted to the laptop with an Intel i7-9750H CPU and an NVIDIA GeForce GTX 1660 Ti GPU for signal processing and inference. Note that we can only use the CPU of the laptop for the inference phase (see §VI-B5 for more details). The back-end server has an AMD EPYC 7402 24-Core CPU and an NVIDIA GeForce RTX 3090 GPU. The back-end server is used for model training and testing. Besides, we also run the inference phase on Jetson Nano [39] to evaluate the performance of *GesturePrint* on the edge device.

System Parameter Settings. In *GesturePrint* implementation, the parameter settings are as follows. The mmWave radar sensor operates in the 60-64 GHz RF band and enables 3 TX and 4 RX antennae. Its frame rate is 10 fps, range resolution is 0.04 m, maximum unambiguous range is 8.2 m, maximum radial Doppler velocity is 2.7 m/s, and radial velocity resolution is 0.34 m/s. As for the parameters in the gesture segmentation module, the point number threshold window length N is 50, the sliding motion detection window length n is 10, and the minimum frame threshold F_{Thr} is 8. For noise canceling, the maximum distance D_{max} is 1 m and the minimum point number N_{min} is 4.

Model Training and Testing. GesIDNet is implemented by using Python and PyTorch [40]. Typically, for all the datasets evaluated in the experiments, the split ratio of the training set and the test set is usually 8:2 with 5-fold cross-validation for reliable results. We employ the same split ratio and settings adopted by existing state-of-the-art methods [15]–[18] on the three public datasets used in the experiments to do a fair comparison with them on the gesture recognition task.

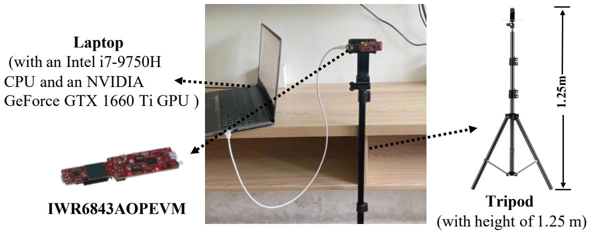


Fig. 7: *GesturePrint* utilizes an IWR6843AOPEVM mmWave radar sensor and a laptop for runtime inference. The radar is positioned at a height of 1.25 m with a tripod.

TABLE I: Dataset summary.

Dataset	Main Scenario	Gesture	User
Pantomime [17]	Office Open	21 (self-defined)	26 14
mHomeGes [15]	Home	10 (self-defined)	8-14
mTransSee [16]	Home	5 (self-defined)	32
Self-collected (GesturePrint)	Office Meeting Room	15 (ASL gestures)	17

VI. Performance Evaluation

A. Datasets and Methodology

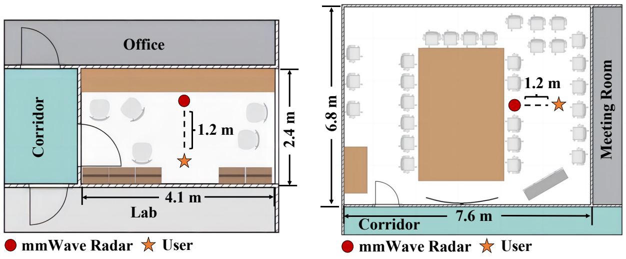
1) Gesture Datasets: We evaluate *GesturePrint* on four datasets that span diverse scenarios (office, meeting room, home, and open space), user scales, and predefined gestures. These datasets include our self-collected dataset (GesturePrint), and three public datasets, i.e., the Pantomime dataset [17], the mHomeGes dataset [15], and the mTransSee dataset [16]. All the public datasets contain self-defined gestures. In contrast, gestures in our self-collected dataset are standard gestures in ASL (American Sign Language). Tab. I provides a summary of these datasets.

Pantomime dataset [17]. It contains 22,291 samples of 21 self-defined gestures, including 9 easy single-arm gestures and 12 bimanual complex gestures, from 4-26 volunteers in different environments, positions, and articulation speeds.

mHomeGes dataset [15]. It contains 22,000 samples from 8-14 participants performing 10 self-defined large arm movements at different anchor points ranging from 1.2 m to 3.0 m with an equal interval of 0.15 m.

mTransSee dataset [16]. It contains data from 32 participants at 13 anchor positions ranging from 1.2 m to 4.8 m, and the predefined gestures are 5 self-defined arm motions.

Self-collected dataset (GesturePrint). We conducted our data collection in two major environments, i.e., a small office and a large meeting room, as shown in Fig. 8. As demonstrated in Fig. 9, we choose 15 basic and representative ASL signs from ASLLVD [41] as predefined gestures. The selected ASL signs contain 9 single-arm gesture motions and 6 bimanual



(a) Office. (b) Meeting room.

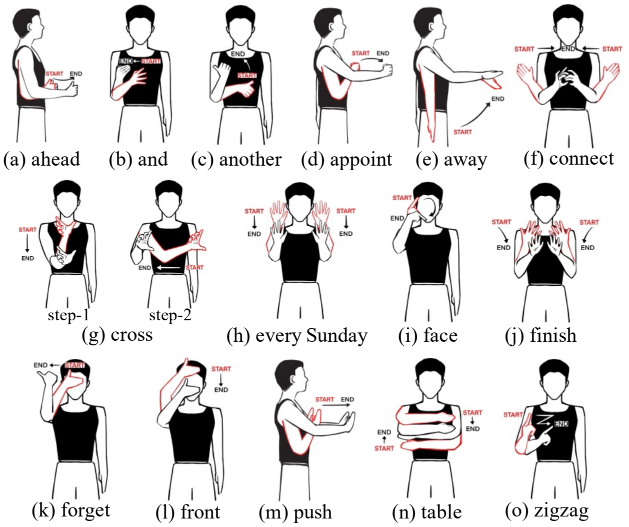
Fig. 8: Experimental environments.

gesture motions, including 'ahead', 'and', 'another', 'appoint', 'away', 'connect', 'cross', 'every Sunday', 'face', 'finish', 'forget', 'front', 'push', 'table' and 'zigzag'. The chosen signs encompass a variety of combinations involving motions of the hands, forearms, elbows, and arms. We recruited 17 participants (10 females and 7 males) aged 20–27, weight 40–85 kg, and height 1.55–1.80 m to perform each gesture 12-25 times in each environment. Before collecting gesture data from the participants, we demonstrated how to perform the ASL gestures by playing a video illustrating standard gesture motion instructions, and the participants learned how to perform gestures based on their understanding. During data collection, the participants were instructed to face the device at a distance of 1.2 m and performed gestures continuously with a time interval of about 2-4 seconds between every two gesture motions. The data collection process spanned 16 days. For the two different environments, individual participants were involved in the data collection on different days. Finally, with 2 environments, 17 participants, and 15 ASL gestures, we collected 9,332 samples.

2) Baseline: We compare *GesturePrint* with state-of-the-art methods, including the Pantomime network (denoted as PanArch) from Pantomime [17], Tesla from Tesla-Rapture [18] (Pantomime's follow-up work that is evaluated on the Pantomime dataset), mGesNet from mHomeGes [15], and mSeeNet from mTransSee [16]. All these existing gesture recognition studies are evaluated on their own gesture datasets as mentioned above. Differently, to evaluate the universality of *GesturePrint*, we also apply it to the other three public mmWave-based point clouds gesture datasets apart from our self-collected dataset. Moreover, we reuse these datasets from another dimension for user identification without any extra information. Note that the state-of-the-art methods are not designed for the user identification task. Thus, we only compare *GesturePrint* with them on the gesture recognition task.

3) Evaluation Metrics: We employ the following metrics to evaluate the performance of *GesturePrint* on gesture recognition (GR) and user identification (UI).

Accuracy (GRA & UIA). GRA is the accuracy of the gesture recognition task, measuring the proportion of correctly predicted gesture samples among all samples. UIA is the accuracy of the user identification task. For our default identification mode, i.e., serialized mode, UIA is the average identification accuracy of all gestures. For the parallel mode, UIA is directly computed once on all gestures. The settings for other metrics related to user identification are similar.



(a) ahead (b) and (c) another (d) appoint (e) away (f) connect (g) cross (h) every Sunday (i) face (j) finish (k) forget (l) front (m) push (n) table (o) zigzag

Fig. 9: 15 ASL signs in the GesturePrint dataset. The start of the gesture is colored in red; the end is colored in black.

F1-Score (GRF1 & UIF1). GRF1 is the F1-Score of recognizing gestures, considering both false positives and false negatives for each gesture. UIF1 is the F1-Score of identifying users, taking into account both false positives and false negatives for each user.

AUC (GRAUC & UIAUC). GRAUC is the area under the receiver operating characteristic (ROC) curve, and it measures the system's discriminatory power in recognizing gestures. UIAUC is employed for the user identification task.

Equal Error Rate (EER). It is the rate at which false positive rate (FPR) equals false negative rate (FNR). The metric is used to measure user identification performance. FPR is the probability that the system fallaciously identifies others as the target user; FNR is the probability that the system identifies the target user as others.

B. Experimental Results

1) ***Overall Performance****:* We evaluate the performance of *GesturePrint* on four datasets, including three public datasets and our self-collected dataset. To fairly compare *GesturePrint* with state-of-the-art methods on the three public datasets, the distance between the radar and the user should be the same across all the datasets. Thus, we utilize gesture data collected at 1.2 m in the mHomeGes, mTransSee, and self-collected datasets for overall performance comparison. However, the Pantomime dataset does not have such data. We utilize gesture data collected at 1 m in the Pantomime dataset, which is the closest distance to 1.2 m. Besides, state-of-the-art methods on the Pantomime dataset (PanArch and Tesla) are evaluated with the models trained on data from both the Office and

TABLE II: Overall gesture recognition and user identification performance. GP represents *GesturePrint*. GP-S denotes GP-Serialized (the default identification mode), while GP-P denotes GP-Parallel. SOTA represents state-of-the-art results. *GesturePrint* achieves comparable performance with SOTA on gesture recognition and enables effective user identification.

Dataset	Self-collected (GesturePrint)						Pantomime						mHomeGes			mTransSee		
Scenario	Office			Meeting Room			Office			Open			Home			Home		
Metrics	GRA	GRF1	GRAUC	GRA	GRF1	GRAUC	GRA	GRF1	GRAUC	GRA	GRF1	GRAUC	GRA	GRF1	GRAUC	GRA	GRF1	GRAUC
SOTA	/			/			0.9714	-	0.9994 [18]	0.9612	-	0.9994 [17]	0.9800 [15]	-	-	0.9800 [16]	-	-
GP (ours)	0.9822	0.9821	0.9908	0.9887	0.9885	0.9942	0.9854	0.9846	0.9997	0.9662	0.9633	0.9993	0.9960	0.9957	0.9966	0.9988	0.9988	0.9992
Metrics	UIA	UIF1	UIAUC	UIA	UIF1	UIAUC	UIA	UIF1	UIAUC	UIA	UIF1	UIAUC	UIA	UIF1	UIAUC	UIA	UIF1	UIAUC
SOTA	/			/			/			/			/			/		
GP-S (ours)	**0.9926**	**0.9901**	**0.9947**	**0.9978**	**0.9972**	**0.9990**	**0.9985**	**0.9972**	**0.9987**	**0.9931**	**0.9902**	**0.9962**	**0.9933**	**0.9925**	**0.9969**	**0.9760**	**0.9707**	**0.9913**
GP-P (ours)	0.9863	0.9811	0.9908	0.9909	0.9906	0.9954	0.9909	0.9908	0.9959	0.9865	0.9850	0.9928	0.9897	0.9895	0.9938	0.9398	0.9386	0.9742

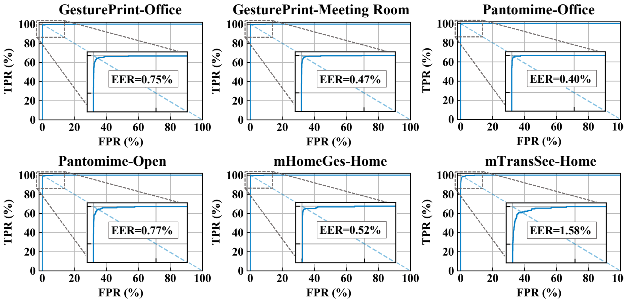


Fig. 10: The ROC curves and EER results of user identification. *GesturePrint* achieves an average result of 0.75% EER.

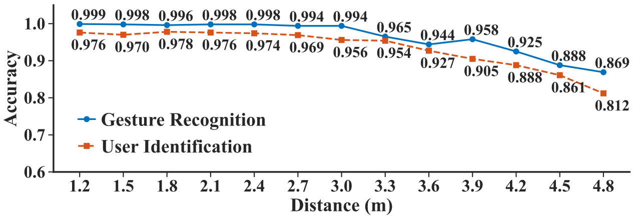


Fig. 11: Impact of distance on *GesturePrint*'s performance, the metrics in the figure are GRA and UIA.

Open subsets while being tested against different environments respectively. Thus, for gesture recognition, we follow their settings. For user identification, we train and test on the Office and Open subsets separately, since the participants in these two environments are different in the Pantomime dataset.

Gesture Recognition. To validate *GesturePrint*'s ability to achieve reliable gesture recognition, we use gesture samples collected from all participants for training and testing. As shown in Tab. II, the overall recognition accuracy is above 96% for gesture recognition across all the datasets. On our self-collected dataset, *GesturePrint* achieves GRA of 98.22% and 98.87% in the office and meeting room environments, respectively. Furthermore, we evaluate *GesturePrint* on the other three datasets. To be specific, on the Pantomime dataset, we achieve 98.54% GRA in the office environment, and 96.62% GRA in the open space environment. On the mHomeGes dataset, we achieve 99.60% GRA at the 1.2 m anchor position. On the mTransSee dataset, we achieve 99.88% GRA at 1.2 m. Compared with the state-of-the-art results on the three datasets [15]–[18], *GesturePrint* achieves comparable results with 0.5%-1.88% accuracy improvements. Besides, the system maintains GRF1 above 0.96 and GRAUC exceeding 0.99 across all the datasets. All the results indicate *GesturePrint*'s effectiveness and universality on the gesture recognition task.

User Identification. As for our default identification mode, i.e., serialized mode, the overall identification performance is over 97% as shown in Tab. II. On the self-collected dataset, *GesturePrint* achieves 99.26% UIA and 99.78% UIA in the office and meeting room, respectively. Furthermore, we evaluate *GesturePrint* on the other three datasets, and find it can maintain high accuracy for user identification based on different gestures. The results show that *GesturePrint* is effective with different user scales. With the scale of users increasing, *GesturePrint* can still successfully maintain 97.60% UIA for recognizing 32 users on the mTransSee dataset. For *GesturePrint* with parallel mode, the UIA results have a slight drop within 4% compared with the serialized mode, indicating that our method can indeed extract personal motion patterns even across different gestures. Besides, the system consistently achieves reliable UIF1 and UIAUC across all the datasets. Moreover, as shown in Fig. 10, *GesturePrint* achieves an average result of 0.75% EER across all the datasets, with none exceeding 1.6% EER. The above results indicate the system's effectiveness in identifying users.

Overall, *GesturePrint* achieves superior performance (GRA and GRF1 above 96%, GRAUC above 99%; UIA and UIF1 above 97%, UIAUC above 99%, EER below 1.6%) across all the datasets, indicating that it is effective in achieving gesture recognition and enabling gesture-based user identification.

*2) **Impact of Distance:*** In this part, we first discuss the impact of the distance between the radar and the user on *GesturePrint*'s performance. Subsequently, we further evaluate *GesturePrint* within a certain range of distance variations.

We evaluate *GesturePrint* across all the anchor positions within the mTransSee dataset, and the results are demonstrated in Fig. 11. As for the overall performance of *GesturePrint* at different positions, *GesturePrint* can maintain both reliable

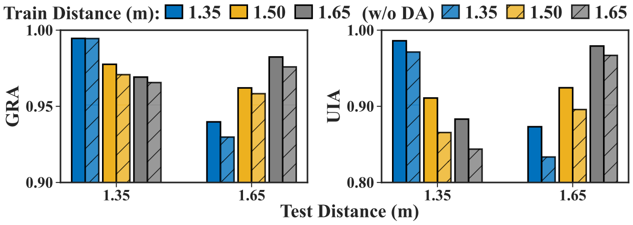


Fig. 12: Distance robustness experiments on different anchor positions (1.35 m, 1.5 m, 1.65 m). DA: data augmentation.

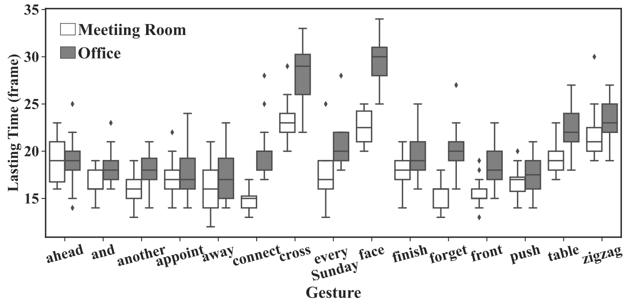
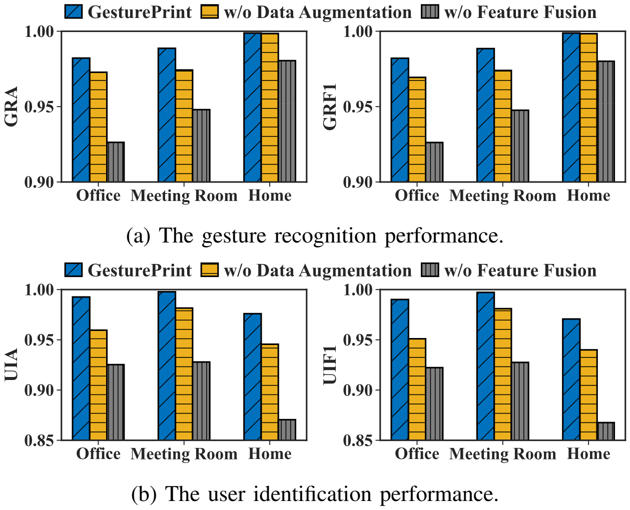


Fig. 13: Lasting time of gesture motions performed by the same user. Users exhibit variations in their motion speed when repeatedly executing the same gesture.



(a) The gesture recognition performance.

(b) The user identification performance.

Fig. 14: Ablation experiments about data augmentation and multilevel feature fusion. 'Office' and 'Meeting Room' are subsets from the GesturePrint dataset and 'Home' is from the mTransSee dataset. The results show that these components effectively contribute to *GesturePrint*'s performance.

recognition performance (≥ 94.4% GRA) and identification performance (≥ 92.7% UIA) within a distance of 3.6 m. Besides, although when working at a distance exceeds 3.9 m, *GesturePrint*'s performance is not as good as that within 3.6 m, the system can still achieve 86.9% GRA and 81.2% UIA at the furthest distance of 4.8 m. The performance degradation of *GesturePrint* working at a distant position is because the point number captured by the radar rapidly decreases with increasing distance due to the limitation of the hardware settings [15].

We further evaluate the distance robustness of *GesturePrint* on a subset comprising three different anchor positions (1.35 m, 1.5 m, and 1.65 m) of the mHomeGes dataset. We use data from different distances as the training set and the test set, respectively. As shown in Fig. 12, the results indicate that *GesturePrint* can achieve reliable performance on both gesture recognition and user identification when confronted with positions that are previously unseen during training.

Considering the limitation of the device configuration, distance robustness of *GesturePrint*, and practical application scenarios, when users try to interact with *GesturePrint* from a distant position, *GesturePrint* can remind the user to step closer and enter the area where it can work reliably.

*3) **Impact of Motion Speed**:* An implicit insight behind *GesturePrint* is that there is some information in the gesture point clouds that can reflect one's specific motion style and behavior personality, thus we can design effective network architecture to extract high-dimension features containing such information for user identification. However, there is a question of whether such features will distort when users perform the same gesture at different speeds, leading to the extracted features becoming ineffective. As shown in Fig. 13, it is inevitable that users slightly change their motion speed unconsciously, evidenced by the gesture's lasting time measured by the frame number. Based on the observation, we further evaluate *GesturePrint* on a subset of the Pantomime dataset, which includes three different motion speeds. The results demonstrate that, with deliberate change in the motion speed, *GesturePrint* can still reach 97.73% GRA and 98.81% UIA. Thus, even if users change motion speeds when performing gestures, *GesturePrint* can still extract effective features for gesture recognition and user identification, respectively.

*4) **Ablation Experiment**:* In this part, we discuss the impacts of data augmentation in the data preprocessing stage and the attention-based multilevel feature fusion module in GesIDNet through ablation experiments. As shown in Fig. 14, data augmentation and the feature fusion module can improve the performance of *GesturePrint* on both the gesture recognition and user identification tasks. The designed feature fusion module contributes considerably to the performance improvement, and the contribution is extremely obvious in the scenarios with a large user scale, e.g., the 'Home' scenario from the mTransSee dataset. Moreover, data augmentation enhances *GesturePrint*'s robustness to distance variations. As shown in Fig. 12, without data augmentation, *GesturePrint*'s performance decreases at distances unseen during training.

*5) **Time Consumption**:* The time consumption of *GesturePrint* to process a single gesture sample comprises the data preprocessing time and the classification inference time. To evaluate *GesturePrint*'s efficiency, we measure the average preprocessing time based on our self-collected dataset, and the classification inference time is measured by averaging the inference results over 500 runs.

The experimental results show that the average preprocessing time is $405.93\,\mathrm{ms}$. As for the classification inference time, when using the CPU of the laptop alone, the average inference time for recognition and identification is $677.14\,\mathrm{ms}$. By using the GPU of the laptop, the inference time can be reduced to $530.99\,\mathrm{ms}$. The average total time consumption for processing a single gesture sample is $936.92\,\mathrm{ms}$ ($0.94\,\mathrm{s}$), while the average gesture duration in our dataset is $2.43\,\mathrm{s}$. Thus, we believe that the time consumption of *GesturePrint* can meet the requirements of most gesture interaction applications.

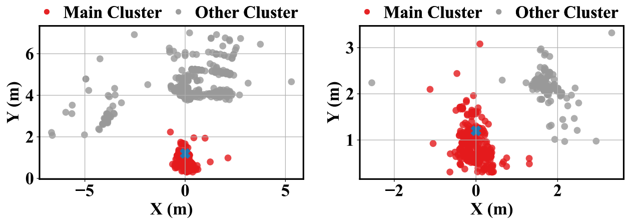
Besides, we deploy the models on Jetson Nano [39] to evaluate the inference time on an edge device. The inference time is $1.58\,\mathrm{s}$ in total for the gesture recognition and user identification tasks. There are some other edge devices delivering superior computing power and AI performance compared with Jetson Nano. For example, the newly launched Jetson Orin Nano Series [42] can offer up to $80\times$ the performance of the standard Jetson Nano we utilized in the experiments. Implementing *GesturePrint* on these more powerful devices could further reduce the inference time. This indicates the potential for applications of *GesturePrint* on the edge.

VII. Discussion

*1) **Multi-person Scenarios:*** Multiple persons may be active in the scenario when a user performs gestures, which affects the system's performance adversely. Although *GesturePrint* is mainly designed to unleash the potential of existing gesture recognition systems by enabling user identification, it can handle some common multi-person cases. By distinguishing main point clusters and noise canceling, *GesturePrint* reduces distractions from others when the user interacts with the system. Fig. 15 shows two common cases, i.e., someone else walking around or performing gestures. *GesturePrint* can figure out the point clusters related to the user during the data preprocessing stage. Note that the minimum distinguishable distance between the user and others is primarily determined by the parameters of DBScan, which can be measured and configured accordingly before the deployment of *GesturePrint*. Furthermore, in practice, a work zone can be predefined to remind users to perform gestures within a specific area for reliable identification as suggested in §VI-B2. In this way, the impact of other people outside the area can be further mitigated. As for those who perform gestures very close to the user on purpose, the user could easily notice such interference.

In future work, *GesturePrint* could be extended to address more complex multi-user scenarios, where the system needs to simultaneously process interaction gestures from multiple users. The multi-user detection method proposed by m^3Track [43] provides a solution to improve *GesturePrint*.

*2) **Cross-domain Scenarios:*** Cross-domain adaptability is a pervasive and important issue in sensing. As discussed in RF-Net [44], RF sensing is more sensitive to environmental changes as it detects both the subject and background. Consequently, exploring the impacts of different positions and distinct environments on our system is beneficial. We have conducted experiments under cross-distance and cross-environment settings to gain insights.



(a) Someone else walks past behind the user.

(b) Someone else performs gestures at the same time.

Fig. 15: *GesturePrint* separates main point clusters related to the target user from other clusters. The blue cross in the figures denotes the position of the user.

As discussed in §VI-B2, Fig. 12 demonstrates that *GesturePrint* maintains over 93% GRA and 87% UIA under cross-distance settings, indicating the system's capability and potential for handling gesture interactions in unseen positions. For conducting cross-environment experiments, we utilize the two distinct environments in our self-collected dataset, training models on data from Office/Meeting Room while testing them on Meeting Room/Office. The results indicate that *GesturePrint* maintains over 90% GRA and about 75% UIA under two cross-environment situations. The performance decline resulting from cross-environment challenges can be mitigated by fine-tuning the models with data collected from the target environment.

Currently, *GesturePrint* is mainly designed for scenarios where authorized users interact with devices, such as smart homes/buildings. In these scenarios, mmWave radar sensors are typically placed in a fixed position, which allows for a predefined work zone. Future work that addresses the cross-domain issue could extend *GesturePrint* to address broader application scenarios.

VIII. Conclusion

This paper proposes *GesturePrint*, a one-stop solution including gesture recognition and further gesture-based user identification based on the mmWave radar. *GesturePrint* extracts unique gesture features for gesture recognition and personalized motion pattern features for user identification, respectively. We design a signal processing stage to efficiently preprocess the raw data to obtain gesture point clouds. Furthermore, we design a network architecture GesIDNet with an attention-based multilevel feature fusion mechanism to extract effective features from gesture point clouds. Extensive experiments demonstrate *GesturePrint*'s effectiveness in achieving reliable gesture recognition and enabling user identification. To summarize, *GesturePrint* is an effective system that unleashes the potential of mmWave-based gesture recognition systems with user identification at a minor cost.

Acknowledgment

Chaojie Gu is the corresponding author. This work was completed when Keyi Wang was a final-year undergrad at Zhejiang University. Thank the anonymous reviewers for providing valuable feedback on this work. Thank the volunteers for participating in the data collection. This research is supported by the National Natural Science Foundation of China under Grants No. (62302439, U23A20296), the Key Research and Development Program of Zhejiang Province (2021C03098), and the Fundamental Research Funds for the Central Universities (226-2023-00111, 226-2024-00004).

References

[1] S. Chadwick, W. Maddern, and P. Newman, "Distant vehicle detection using radar and vision," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 8311–8317.

[2] S. Chang, Y. Zhang, F. Zhang, X. Zhao, S. Huang, Z. Feng, and Z. Wei, "Spatial attention fusion for obstacle detection using mmwave radar and vision sensor," *Sensors*, vol. 20, no. 4, p. 956, 2020.

[3] P. Zhao, C. X. Lu, J. Wang, C. Chen, W. Wang, N. Trigoni, and A. Markham, "mid: Tracking and identifying people with millimeter wave radar," in *IEEE DCOSS*, 2019, pp. 33–40.

[4] C. Wu, F. Zhang, B. Wang, and K. R. Liu, "mmtrack: Passive multi-person localization using commodity millimeter wave radio," in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*. IEEE, 2020, pp. 2400–2409.

[5] A. D. Singh, S. S. Sandha, L. Garcia, and M. Srivastava, "Radhar: Human activity recognition from point clouds generated through a millimeter-wave radar," in *Proceedings of the 3rd ACM Workshop on Millimeter-wave Networks and Sensing Systems*, 2019, pp. 51–56.

[6] L. Xu, C. Gu, R. Tan, S. He, and J. Chen, "Mesen: Exploit multimodal data to design unimodal human activity recognition with few labels," in *Proceedings of the 21st ACM Conference on Embedded Networked Sensor Systems*, 2023, pp. 1–14.

[7] Z. Yang, P. H. Pathak, Y. Zeng, X. Liran, and P. Mohapatra, "Monitoring vital signs using millimeter wave," in *Proceedings of the 17th ACM international symposium on mobile ad hoc networking and computing*, 2016, pp. 211–220.

[8] 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.

[9] 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.

[10] G. Laput, R. Xiao, and C. Harrison, "Viband: High-fidelity bio-acoustic sensing using commodity smartwatch accelerometers," in *UIST*, 2016.

[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 *ACM MobiCom*, 2019.

[12] K. Guo, H. Zhou, Y. Tian, W. Zhou, Y. Ji, and X.-Y. Li, "Mudra: A multi-modal smartwatch interactive system with hand gesture recognition and user identification," in *IEEE INFOCOM*, 2022.

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

[14] A. Virmani and M. Shahzad, "Position and orientation agnostic gesture recognition using wifi," in *ACM MobiSys*, 2017, pp. 252–264.

[15] 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," *ACM IMWUT*, vol. 4, 2020.

[16] H. Liu, K. Cui, K. Hu, Y. Wang, A. Zhou, L. Liu, and H. Ma, "mtranssee: Enabling environment-independent mmwave sensing based gesture recognition via transfer learning," *ACM IMWUT*, vol. 6, 2022.

[17] S. Palipana, D. Salami, L. A. Leiva, and S. Sigg, "Pantomime: Mid-air gesture recognition with sparse millimeter-wave radar point clouds," *ACM IMWUT*, vol. 5, no. 1, 2021.

[18] D. Salami, R. Hasibi, S. Palipana, P. Popovski, T. Michoel, and S. Sigg, "Tesla-rapture: A lightweight gesture recognition system from mmwave radar sparse point clouds," *IEEE TMC*, 2022.

[19] X. Yu, Z. Zhou, M. Xu, X. You, and X.-Y. Li, "Thumbup: Identification and authentication by smartwatch using simple hand gestures," in *IEEE PerCom*, 2020.

[20] C. Li, M. Liu, and Z. Cao, "Wihf: enable user identified gesture recognition with wifi," in *IEEE INFOCOM*, 2020.

[21] P. S. Santhalingam, A. A. Hosain, D. Zhang, P. Pathak, H. Rangwala, and R. Kushalnagar, "mmasl: Environment-independent asl gesture recognition using 60 ghz millimeter-wave signals," *ACM IMWUT*, vol. 4, no. 1, pp. 1–30, 2020.

[22] Y. Li, D. Zhang, J. Chen, J. Wan, D. Zhang, Y. Hu, Q. Sun, and Y. Chen, "Towards domain-independent and real-time gesture recognition using mmwave signal," *arXiv preprint arXiv:2111.06195*, 2021.

[23] E. Hayashi, J. Lien, N. Gillian, L. Giusti, D. Weber, J. Yamanaka, L. Bedal, and I. Poupyrev, "Radarnet: Efficient gesture recognition technique utilizing a miniature radar sensor," in *CHI*, 2021, pp. 1–14.

[24] H. Liu, A. Zhou, Z. Dong, Y. Sun, J. Zhang, L. Liu, H. Ma, J. Liu, and N. Yang, "M-gesture: Person-independent real-time in-air gesture recognition using commodity millimeter wave radar," *IEEE Internet of Things Journal*, vol. 9, no. 5, pp. 3397–3415, 2021.

[25] H. Li, C. Xu, A. S. Rathore, Z. Li, H. Zhang, C. Song, K. Wang, L. Su, F. Lin, K. Ren *et al.*, "Vocalprint: exploring a resilient and secure voice authentication via mmwave biometric interrogation," in *SenSys*, 2020.

[26] Y. Dong and Y.-D. Yao, "Secure mmwave-radar-based speaker verification for iot smart home," *IEEE IoT-J*, vol. 8, no. 5, 2020.

[27] Y. Wang, T. Gu, T. H. Luan, and Y. Yu, "Your breath doesn't lie: Multi-user authentication by sensing respiration using mmwave radar," in *IEEE SECON*, 2022.

[28] X. Yang, J. Liu, Y. Chen, X. Guo, and Y. Xie, "Mu-id: Multi-user identification through gaits using millimeter wave radios," in *IEEE INFOCOM*, 2020.

[29] Z. Meng, S. Fu, J. Yan, H. Liang, A. Zhou, S. Zhu, H. Ma, J. Liu, and N. Yang, "Gait recognition for co-existing multiple people using millimeter wave sensing," in *AAAI*, 2020.

[30] Y. Cheng and Y. Liu, "Person reidentification based on automotive radar point clouds," *IEEE TGARS*, vol. 60, pp. 1–13, 2021.

[31] K. Pfeuffer, M. J. Geiger, S. Prange, L. Mecke, D. Buschek, and F. Alt, "Behavioural biometrics in vr: Identifying people from body motion and relations in virtual reality," in *ACM CHI*, 2019.

[32] M. Berger, J. A. Levine, L. G. Nonato, G. Taubin, and C. T. Silva, "A benchmark for surface reconstruction," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 2, pp. 1–17, 2013.

[33] H. Fan, H. Su, and L. J. Guibas, "A point set generation network for 3d object reconstruction from a single image," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 605–613.

[34] P. Achlioptas, O. Diamanti, I. Mitliagkas, and L. Guibas, "Learning representations and generative models for 3d point clouds," in *International conference on machine learning*. PMLR, 2018, pp. 40–49.

[35] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *31st NIPS*, 2017.

[36] P. Ballester and R. M. Araujo, "On the performance of googlenet and alexnet applied to sketches," in *AAAI*, 2016.

[37] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.

[38] Texas Instruments, "IWR6843AOPEVM." [Online]. Available: https://www.ti.com/tool/IWR6843AOPEVM

[39] NVIDIA Corporation, "Jetson Nano Developer Kit." [Online]. Available: https://developer.nvidia.com/embedded/jetson-nano-developer-kit

[40] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, 2019.

[41] V. Athitsos, C. Neidle, S. Sclaroff, J. Nash, A. Stefan, Q. Yuan, and A. Thangali, "The american sign language lexicon video dataset," in *IEEE CVPR Workshops*, 2008.

[42] NVIDIA Corporation, "Jetson Orin Modules and Developer Kits," 2023. [Online]. Available: https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-orin/#advanced-features

[43] 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 *ACM MobiSys*, 2022.

[44] S. Ding, Z. Chen, T. Zheng, and J. Luo, "Rf-net: A unified meta-learning framework for rf-enabled one-shot human activity recognition," in *ACM SenSys*, 2020.