Real-time and Accurate Gesture Recognition with Commercial RFID Devices

Shigeng Zhang, *Member, IEEE,* Zijing Ma, Chengwei Yang, Xiaoyan Kui, Xuan Liu, Weiping Wang, Jianxin Wang, *Senior Member, IEEE,* and Song Guo *Fellow, IEEE*

Abstract—Gesture recognition based on radio frequency identification (RFID) has attracted much research attention in recent years. Most existing RFID-based gesture recognition approaches use signal profile matching to distinguish different gestures, which incur large recognition latency and fail to support real-time applications. In this paper, we design and implement ReActor, a real-time and accurate gesture recognition system that recognizes a user's gestures with low latency and high accuracy even when the gestures'speed varies. ReActor combines the time-domain statistical features and the frequency-domain features to precisely represent the signal profile corresponding to different gestures. To maintain high accuracy across different environments, we preprocess the signals to remove reflection signals from surrounding objects and use only the signals related to gestures to train the classifier. Moreover, we train a classifier to predict the speed of the gesture and feed the extracted features to different classifiers according to the speed. We implement ReActor and evaluate its performance in different scenarios. Experimental results show that ReActor achieves an average accuracy of 97.2% in recognizing 18 different gestures with an average latency of 72 ms, more than two orders of magnitude faster than approaches based on profile template matching.

Index Terms—gesture recognition, radio frequency identification, real time, machine learning, contactless, see-through walls

1 Introduction

SMART wireless sensing has emerged as an enabling technology for many smart applications, e.g., contactless vital signs monitoring [1], [2], novel human-machine interaction (HMI) [3], augmented reality (AR) [4], and ubiquitous computing [5]–[7]. For example, wireless gesture recognition can be used to control computers without traditional input devices such as keyboards and mouses [8], [9]. In many scenarios like AR games, gesture recognition is essential to capture user motions and map the motions to the game scenarios [10]. Real-time and accurate gesture recognition is also important to enable smart healthcare for elder people, e.g., to understand their intention or to detect the fall event of the elderly [11].

Existing gesture recognition approaches can be roughly classified into two categories: contact-based approaches and contactless approaches. In contact-based approaches, the user needs to wear some dedicated devices that contain multiple sensors (e.g., accelerometers or gyroscopes), and the gestures are recognized by tracing the motions of these devices [12]–[14]. In contrast, contactless approaches do not require the users to wear such sensors and thus are more convenient to use. The general idea of contactless gesture recognition approaches is to detect changes in signals (e.g., depths for Kinect-like systems, signal strengths or phases for Wi-Fi and millimeter-wave systems) caused by different gestures and distinguish them by matching the changing patterns of the signals with some pre-defined templates [14]. Due to the convenience in use, contactless gesture recognition has attracted increasing research attention in recent years, e.g., vision-based systems like Kinect [15], wireless-based systems [6], [16]–[19], acoustic-signal-based systems [20], [21] and millimeter-wave-based systems [9].

Compared to other contactless gesture recognition approaches, RFID-based approaches [17], [18], [22]–[24] have attracted increasing interest in recent years due to several reasons. First, compared to systems based on computer vision that can work in only bright environments [15], RFID-based approaches can be used in pervasive environments and provide better availability. Second, compared to approaches based on Wi-Fi signals, RFID can support simultaneously gesture recognition multiple users due to its ability in distinguishing signals from different tags (users), while multiple user differentiation remains a difficult problem in Wi-Fi sensing [16]. Third, compared to approaches based on acoustic-signals [20] or millimeter-wave [9], RFID-based approaches have larger operational range (tens of feet vs. several centimeters in acoustic-based approaches), making them more suitable for gesture recognition in real world scenarios that require high flexibility.

Although many RFID-based gesture recognition approaches have been proposed, however, they cannot simultaneously provide *fine-grained* and *real-time* gesture recognition. In a preliminary version of this work [25], we design and implement *ReActor*, a gesture recognition approach that achieved high recognition accuracy with far smaller delay than existing works. First, ReActor can recognize fine-

- *Shigeng Zhang, Zijing Ma, Xiaoyan Kui, Weiping Wang, and Jianxin Wang are with the School of Computer Science and Engineering, Central South University, China. Shigeng Zhang is also with the Zhengzhou Xinda Institute of Advanced Technology. E-mail: {sgzhang, mazijingcsu, xykui, wpwang,jxwang}@csu.edu.cn.*
- *Chengwei Yang is with Research Institute of China Telecom Co., Ltd., China. E-mail: yangcw2@chinatelecom.cn.*
- *Xuan Liu is with the College of Computer Science and Electronic Engineering, Hunan University, China, 410082. E-mail: xuan_liu@hnu.edu.cn.*
- *Song Guo is with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong. Email: song.guo@polyu.edu.hk.*

Manuscript received January 1st, 2021; revised XX, 2021.

grained gestures that have only subtle differences, e.g., *zoom in* and *zoom out*, which cannot be well distinguished in previous works [14], [22]–[24], [26]. To this end, ReActor fuses two types of attributes, namely the coarse-grained statistical features of the signal profile and the wavelet coefficients of the signal profile that characterize fine-grained local features. The combination of the two types of attributes well characterizes the signal profiles related to different gestures, which contributes to accurate gesture recognition. Second, ReActor speeds up the recognition process by building a machine-learning-based classifier, which avoids time-consuming template matching adopted in existing works and significantly reduces recognition latency. The average recognition latency of ReActor is two orders of magnitude lower than traditional approaches based on template matching ($\sim$ 50ms vs. $\sim$10 seconds). Moreover, ReActor uses both time-domain features and frequency-domain features and thus achieves higher accuracy than approaches using only time-domain features such as GRfid [17] and RF-Finger [18].

In this paper, we extend ReActor by considering two factors that might degrade the accuracy of gesture recognition in real environments: the *multi-path* reflection signals from surrounding objects in the environments and the *speed* of the gesture. The multi-path signals reflected from surrounding objects might tangle with the signal patterns caused by different gestures. In both ReActor and other previous works [17], [18], the tangled signals are used to build classifiers [18], [25] or encoded in the profile templates [17], degrading the accuracy of the classification model when the distance between the user and the tags increases. To address this problem, we adopt the method developed in [27] to remove reflection signals caused by surrounding objects to enhance the accuracy. This not only improves recognition accuracy when the operational range increases but also improves the accuracy when the recognition model is used in a new environment. Another factor that affects the recognition accuracy of existing approaches is the speed of the gesture. When the speed of the gesture varies, the signal profile deforms, making the recognition accuracy of both ReActor and existing template-matching approaches degrade. To solve this problem, we train a classifier to first predict the speed of the gesture and feed the features to different classifiers corresponding to different speeds. Because the speed of the gesture can be predicted, this method significantly improves the recognition accuracy when the gesture's speed varies. The enhanced version of ReActor is named *ReActor+*.

We briefly summarize the contribution of this paper as follows.

- A real-time RFID-based gesture recognition approach named ReActor is proposed which can recognize 18 different gestures with an average latency smaller than 100 ms, more than 100X faster than existing works based on template matching. ReActor fuses time-domain and frequency-domain features and builds a classifier based on the fused features. It significantly speeds up the gesture recognition process by avoiding time-consuming template matching used in existing works.
- An extension of ReActor, namely ReActor+, is proposed to further improve the recognition accuracy when the user performs gestures at different speeds and when the user is distant from tags. For the former case, we train a speed classifier and feed the data to different gesture classifiers according to the predicted speed. For the latter case, we remove the signals reflected by surrounding objects to obtain clear signals related only to the gestures to improve accuracy.
- The performance of ReActor/ReActor+ is evaluated on 18 fine-grained gestures with commercial RFID devices. The results show that ReActor/ReActor+ achieves a recognition accuracy higher than 0.97, while the accuracy of existing works based on template matching [17] or convolutional neural networks (CNNs) [18] is lower than 0.93 in the same setting. Moreover, ReActor/ReActor+ significantly outperform existing works when used across different environments or when the user is distant from the tags.

The rest of this paper is organized as follows. In Section 2 we overview related work. The framework of the proposed ReActor approach and its extension ReActor+ are described in Section 3. The details of data processing, including reflection signal removal, gesture segmentation and feature extraction, are given in Section 4. We also describe how to handle the impact of varying speeds of gestures in this section. Extensive experiments are conducted with commercial RFID devices in different environments to evaluate the performance of ReActor/ReActor+, and the results are reported with comparison to related works in Section 5. Finally, we give some concluding remarks in Section 6.

2 Related Work

2.1 Contact-based Gesture Recognition

Early works on gesture recognition are mainly based on wearable sensors. uWave [28] uses a single three-axis accelerometer sensor to recognize personalized gestures with high accuracy. FEMD [29] uses the Kinect sensor to classify ten different gestures. The Magic Ring proposed in [13] recognizes different gestures by attaching a ring to the user's finger. In [14] the authors propose an approach to recognizing coarse-grained body activity of users, which requires the users to attach some RFID readers. Femo [22] recognizes the user's activities during body exercise and assesses the quality of exercise movements. ShopMiner [26] and CBid [30] monitor the customers' behaviors by attaching RFID tags to goods in the supermarket and recognizing different behavior patterns by tracing motions of tags. In [31] the authors combine Kinect-based activity recognition and RFID-based user identification to improve the quality of augmented reality applications. In [24] the authors propose an approach to detecting the user's coarse-grained gesture by attaching tags to goods, which supports online commenting of goods' quality. IDSense [32] enables smart interaction between the user and objects by developing an activity detection systems based on RFID. RF-glove [33] uses three antennas and five commercial tags affixed to the five fingers to construct a contactless smart sensing system, where each finger corresponds to a tag. It achieves fine-grained classification of eight gestures. RF-Dial [34] is a

2D human-computer interaction system that requires two antennas and two tags attached to the object. It uses translation and rotation to track object trajectories to perform gesture recognition. Recently, deep learning is also exploited to recognize user's body activities [23], [35], [36], in which the users need to attach some sensors or RFID tags.

These approaches are contact-based and require the user to wear or attach some sensors or RFID tags. In many scenarios such as elderly care, it is not practical to require the users to wear such sensors or tags. In contrast, the approach proposed in this paper is contactless and thus is convenient to use in practice.

2.2 Contactless Gesture Recognition

Compared with contact-based approaches that require attaching sensors/tags to the users, contactless gesture recognition is more convenient to use and thus has attracted much research attention in recent years. Vision-based recognition has been widely used in augmented reality games [37], [38]. In [37] the authors develop a vision-based system named RGBD that employs a combined RGB and depth descriptor to classify hand gestures. In [38] the authors use deep learning to improve the accuracy of RGBD. These vision-based gesture recognition system can operate normally under certain circumstances. When the light condition of the environment is not good, e.g., the light is too strong or too weak, the accuracy of vision-based gesture recognition systems significantly degrades. Recently, there are some works on recognizing gestures based on acoustic signals [39], [40] or millimeter-wave signals [9]. However, the operational regions of such approaches are greatly limited, making them not suitable for many practical applications that require large operational ranges.

Contactless gesture recognition based on Wi-Fi signals has attracted much research attention in recent years. Compared to vision-based approaches and approaches based on acoustic/millimeter-wave signals, Wi-Fi-based gesture recognition has a much larger operational region and can operate in environments without light. WiGest [41] detects basic primitive gestures in a device-free manner. It achieves an accuracy of 0.87 with a single AP and improves the accuracy to 0.95 with three overhearing APs. E-eyes [42] and ABLSTM [43] detects user's activity at home based on channel state information (CSI). WiFinger [44] detects fine-grained hand gestures based on CSI changes. The essential limitation of Wi-Fi-based gesture recognition is that it is difficult to distinguish between multiple users and thus cannot perform multi-user gesture recognition simultaneously.

RFID-based activity recognition can leverage the inherent identification ability of RFID to simultaneously track multiple users when performing activity recognition. However, their recognition latency is usually very high because they mainly use template matching to distinguish different gestures and their presentation space is constrained because the gesture must be between the antenna and the tag [17], [18]. For example, in GRfid [17], gesture recognition is achieved by matching the signal segment to a set of pre-stored segment templates. This not only increases the recognition latency but also degrades recognition accuracy in some cases because of the weak ability of templates to describe gesture features from different users and at different locations. Compared with them, the approach proposed in this paper significantly reduces recognition latency by two orders of magnitude and achieves even higher accuracy.

2.3 Cross-environment Sensing

Most gesture recognition approaches based on wireless signals are environment-dependent, which means that a recognition model trained in a environment A performs poorly in a new environment B. The reason is that the propagation of wireless signals are greatly affected by surrouding objects in the environment. To address this issue, recently there are a few works on transfering a trained recognition model to new environment [45]–[48]. These works focus on learning the signal mapping relationship between the environment used to train the sensing model and the environment where the model is used, based on which some syntetic data are generatd to tune the model in the new environment to improve accuracy.

In [45], the authors propose CrossSense, which is a data roaming model based on ANN to generate synthetic Wi-Fi signals using the source-domain's data to retrain the classifier in the target-domain. It leverages transfer learning to transfer the trained ANN when the model is used in a new environment, which can decrease the volume of training samples to $1/4$ of the original required samples. WiTransfer [46] uses CNN network instead of ANN to generate synthetic data for the target domain, preventing the network from overfitting. However, if the data from the source-domain and the target-domain do not follow a similar distribution, these models probably generate unreliable synthetic data. With the development of generative models such as Variational Autoencoder(VAE) and Generative Adversarial Networks(GAN), in [47] the authors propose a data roaming model based on VAE to generate reliable synthetic signals for the target-domain. Because VAE is based on Kullback-Leibler divergence, it can well measure the difference between two data distribution and thus generate high-quality data for the target-domain. In [48] the authors propose an unsupervised approach to achieve data transfer. They utilize a deep adversarial network to guide the generation of data in the target-domain by aligning the center of the distribution in the target-domain to the center of the distribution in the source-domain.

3 System Framework

In this section, we present the design of ReActor, includeing a hardware part used to collect data from tags and a software part used to process the obtained data and output the recognized gesture. We then describe the extension of ReActor, including a reflection signal removal component and a gesture speed classification component. We name the extension of ReActor as ReActor+ in the rest of the paper.

3.1 The Hardware Part

The hardware part of ReActor consists of three parts: a set of tags to sense the gesture of the user, an antenna used to communicate with the tags and collect signals backscattered from tags, and a laptop which is used to control the reader (antenna) and analyze the collected signals to recognize

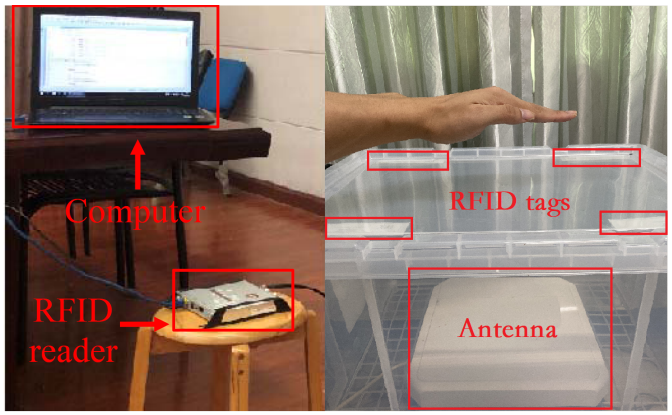


Fig. 1: The hardware part of ReActor. An RFID reader is connected to a laptop (left), which is used to control the antenna to collect and process data from tags (right).

different gestures. As shown in Fig. 1, we attach four Monza AZ-9654 passive tags to the four corners of one transparent plastic cover to sense the user's gesture. The four tags form a rectangle with a side-length of 40cm. The user performs different hand gestures above the plastic cover, and the antenna reads data backscattered from tags and sends the data to the laptop for processing. Note that different deployment strategies of tags might impact the recognition accuracy of ReActor. We will discuss this issue in Section 5.

We use an Impinj Speedway R420 reader and a circularly polarized Laird S9028PCR antenna to continuously interrogate the tags. The antenna is placed at the bottom of the plastic box. When interrogating the tags, the reader can report several types of information: received signal strength (RSS), phase value, and Doppler shifts. As pointed out in [17], the phase values are more sensitive to slight motions than the other types of information. Thus we mainly use the phase values to detect different gestures in ReActor. In our experiment, the reader can interrogate tags with a maximum throughput of about 400 readings per second. After the reader collects data from the tags, it transmits the data to the computer for data processing and gesture recognition. We implement a software in Java to control the communications between the reader and the computer.

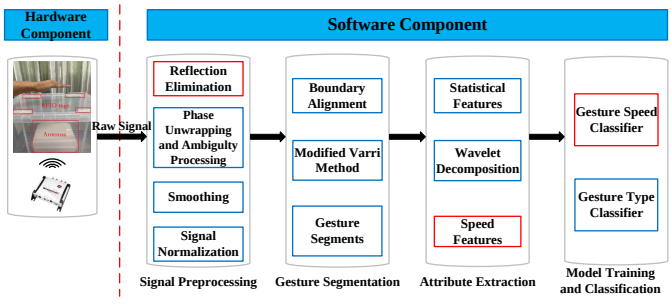


Fig. 2: The software framework of ReActor and its extension ReActor+. The components marked in blue rectangles are used in both ReActor and ReActor+, while the components marked in red rectangles are used only in ReActor+.

3.2 The Software Part

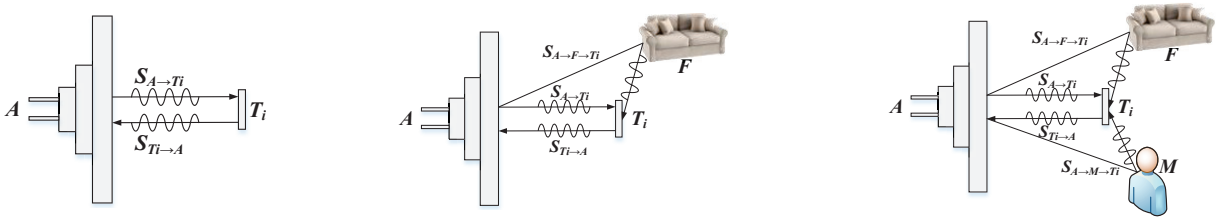
The workflow of ReActor's software part is shown in Fig. 2. It contains the following steps.

- ***Signal Preprocessing:*** The collected reflection data contain noises and should be preprocessed before they can be fed into gesture recognition algorithms. In the preprocessing step, we mainly consider how to perform phase unwrapping, phase ambiguity processing, relfection signal removal, signal smoothing, and signal normalization.
- ***Gesture Segmentation:*** After signal preprocessing, the next step is to detect the signal boundaries corresponding to different gestures and divide the signals into gesture segments. This is challenging because the boundaries of different tags might be misaligned. We use a modified Varri method [49] to segment signals from individual tags, and propose a threshold-based method to align boundaries of different tags.
- ***Attribute Extraction:*** With the obtained gesture segmentation, we extract a set of attributes that will be fed into machine learning algorithms for classifier training and gesture recognition. We extract two types of attributes to capture both coarse-grained global feature and fine-grained local feature of each gesture segment profile. We also perform feature optimization to improve recognition accuracy.
- ***Model Training and Classification:*** With the attributes obtained in the third step, we train a classifier model and use the model to recognize different gestures. We examine different classification approaches and select the one with the highest accuracy. When the user performs a gesture, we first extract attributes by using the previous steps and feed the attributes into the trained classification model to determine the gesture type.

3.3 Extension Components in ReActor+

We extend ReActor to handle two factors that might degrade the recognition accuracy in practical scenarios: the reflection signals from surrounding objects in the environment and the speed of gestures. To this end, we add three modules, which are marked in red rectangles in Fig. 2. The details of the added components will be given in Section 4.

- ***Reflection signal removal:*** Before preprocessing the signals, we first remove the reflection signals from surrounding objects in the environment to obtain signals only caused by the user's gesture by using the method proposed in [27]. This will mitigate the interference from environments and improve recognition accuracy in cross-environment scenarios.
- ***Handling different gesture speeds:*** Different users might perform the same gesture with different speeds, which significantly affects the recognition accuracy. To address this problem, we add two modules in the framework, namely the *speed feature* extraction and the *gesture speed classifier*. We train different classification models for different speeds and use the gesture speed classifier to predict the speed of the gesture before feeding it to the proper gesture classifier.



(a) Ideal case (b) Static objects (c) Dynamic objects

Fig. 3: Modeling the communications between the antenna and tags: (a) the ideal case; (b) considering reflected signals from static objects in the environment; (c) considering reflection signals from dynamic objects in the environment.

4 Gesture Segmentation and Feature Extraction

In this section, we describe in detail how to preprocess the raw signal (Section 4.1), how to divide signal profiles into segmentations corresponding to gestures (Section 4.2), how to extract attributes from the signal segments (Section 4.3), how to handle different gesture speeds (Section 4.4), and how to select the classification model (Section 4.5).

4.1 Signal Preprocessing

The data obtained by the reader contain many noises caused by different factors, e.g., phase wrapping and phase ambiguity caused by hardware imperfection of the reader and antenna, static signals noises caused by environmental factors, and background readings. We first preprocess the raw data to mitigate the effects of such noises before segmenting signals for gesture segments.

4.1.1 Resolving Phase Unwrapping and Phase Ambiguity

The raw phase readings reported by the reader might be wrapped or contain phase ambiguities [17], [50]. As shown in Fig. 4a, phase wrapping means that the phase values reported by the reader wraps when the actual phase approaches 0 or 2π. This phenomenon occurs because the reader usually restricts the phase value to $[0, 2\pi)$ [51]. Thus, when the phase $\phi(t) > 2\pi$ or $\phi(t) < 0$, the value reported by the reader is actually $\phi(t) \mod 2\pi$. Phase ambiguity means that there is an offset of π in some readings when compared with near readings, which are shown as spikes in Fig. 4a. Phase ambiguity is related to the modulation scheme used in RFID. We use the method proposed in [51] to unwrap the raw phase readings and use a median filter to rectify phase ambiguity readings. The signals after phase unwrapping and phase ambiguity resolving are shown in Fig. 4b.

4.1.2 Relfection Signal Removal

There might be multiple propagation paths for the signal to propagate from the antenna to the tag. The signals received at the reader are actually a linear superposition of all the signals propagated through different paths, containing the light-of-sight signal and the signals reflected from static or moving objects in the environment. In gesture recognition, because we want to capture the signal changes caused by the user's gestures, we should remove reflection signals from static objects and retain only the signal reflected from the user's hand.

We first consider the signal transmission between the antenna and the tag in the ideal case, as shown in Fig. 3a. Denote by S_0 the original signal emitted by the antenna A. The signal first propagates to tag T_i and is then backscattered by the tag to the reader, where it is measured. Denote by $S_{A\to T_i}$ and $S_{T_i\to A}$ the signals received by T_i and the backscattered signals at the antenna, respectively. Then we have [27], [52]

$$S_{A\to T_i} = S_0 \cdot G_{A,T} \cdot h_{A\to T_i} \cdot G_{T_i,R}, \tag{1}$$

$$S_{T_i\to A} = S_{A\to T_i} \cdot G_{T_i,T} \cdot h_{T_i\to A} \cdot G_{A,R}, \tag{2}$$

where $h_{A\to T_i}$ and $h_{T_i\to A}$ are the channel responses of the channel $A \to T_i$ and the channel $T_i \to A$, respectively. Here $G_{A,T}$, $G_{A,R}$, $G_{T_i,T}$, and $G_{T_i,R}$ are the antenna gains at the reader and the tag T_i when sending and receiving signals.

When there are static objects in the environment, the signals reflected from these objects also affect the signal received at the reader antenna. As shown in Fig. 3b, the signals reflected by the static objects (such as F in the figure) will arrive at T_i and will be backscattered by the tag together with the signal received from the reader antenna. Denote by F all the static objects in the environment. There are two channels along with F, including $C_{A\to F\to T_i}$ and $C_{A\to F\to A}$. The channel $C_{A\to F\to A}$ is called the self-reflection channel, and experiments in Tadar [27] have shown that the self-reflection channel has negligible influence on the tag's backscattered signals because current commercial readers can well handle this. Thus we only consider the reflection signals arrived at the tag, which can be expressed as

$$S_{T_i+F\to A} = (S_{A\to T_i} + S_{A\to F\to T_i}) \cdot G_{T_i,T} \cdot h_{T_i\to A} \cdot G_{A,R}. \tag{3}$$

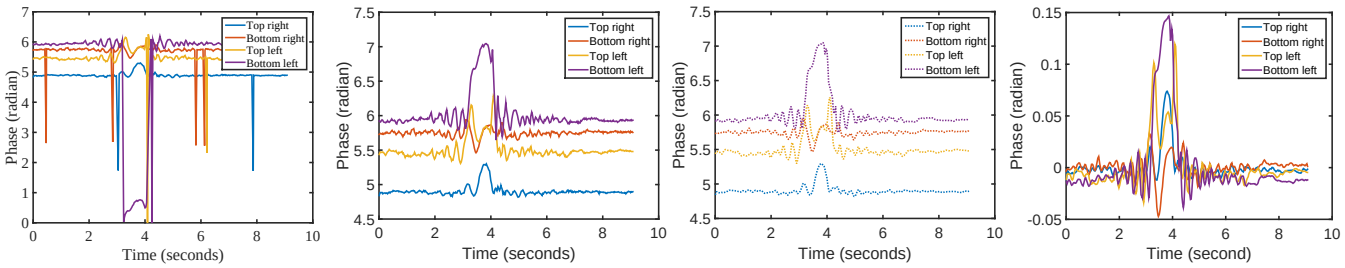
When the user performs gestures, the user's hand movement introduces new reflection component of the signal and generates new propagation paths, as shown in Fig. 3c. Denote by M the user's body part that moves to perform the gesture. There will be a new propagation channel $C_{A\to M\to T_i}$, and thus the signal received at the reader's antenna can be represented as

$$S_{T_i+F+M\to A} = (S_{A\to T_i} + S_{A\to F\to T_i} + S_{A\to M\to T_i}) \cdot G_{T_i,T} \cdot h_{T_i\to A} \cdot G_{A,R}. \tag{4}$$

To get the clear signal corresponding to the user's gesture, we subtract Eq. (3) from Eq. (4), and get

$$S_{gesture} = S_{T_i+F+M\to A} - S_{T_i+F\to A}, \tag{5}$$

where $S_{T_i+F+M\to A}$ and $S_{T_i+F\to A}$ can be estimated by using the RSS values and the phase values measured when the



(a) Raw signal (b) Phase unwrapping (c) Filtering and Smoothing (d) After normalization

Fig. 4: Signal preprocessing of ReActor: (a) the phase of the signals backscattered from four tags; (b) the data after phase unwrapping and phase ambiguity resolving; (c) the data after outlier filtering and smoothing; (d) the data after normalization to [-1,1].

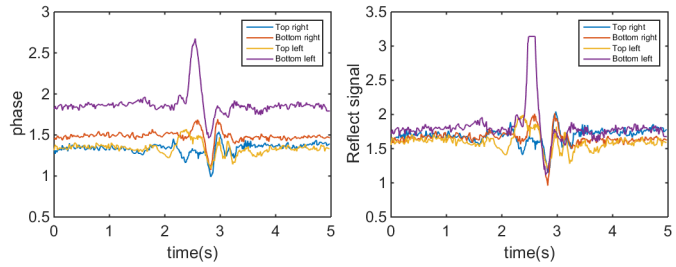
user performs the gesture and when there are no user in the environment, respectively. Take $S_{T_i+F+M\to A}$ as an example. Denote by RSS and θ the measured signal strength value and the phase value when the user performs the gesture, respectively. Then $S_{T_i+F+M\to A}$ can be estimated as

$$S_{T_i+F+M\to A} \approx \alpha e^{i\theta}, \tag{6}$$

where

$$a = 10^{\sqrt{\frac{RSS}{1000}}}. \tag{7}$$

With this method, the interference signals reflected from static objects can be effectively removed and the signals reflected from moving objects are retained, which correspond to the user's gestures. Fig. 5 shows an instance of the signal removal, where the raw phase measurements are shown in Fig. 5a and the phase values after removing reflection signals from static objects are shown in Fig. 5b.



(a) Raw Data (b) Reflector Data

Fig. 5: Reflection signal removal in ReActor+: (a) the raw phase readings obtained from four tags; (b) the phase readins after removing reflection signals from static objects.

4.1.3 Data Filtering and Smoothing

There are still some noisy readings caused by environmental factors after phase unwrapping. To smooth the data and filter out such noisy readings, we apply the Savitzky-Golay (S-G) filter [53] to the data after phase unwrapping. The S-G filter is a method based on local polynomial least square fitting in the time domain. It has been widely used in data stream smoothing and denoising because it can preserve the shape and width of the raw signal after filtering out noises. The data after applying the S-G filter are shown in Fig. 4c.

4.1.4 Signal Normalization

To make the signal changes caused by gestures more ?significant and ease gesture segmentation, we normalize the signals to mitigate the effect of background readings, e.g., those readings seem "flat" in Fig. 4c. For this purpose, we use the normalization method proposed in [54] to map the filtered data to a range of $[-1, 1]$. The signal normalization can magnify the signal changes caused by gestures and meanwhile suppress the impact of background signals by mapping them to values around zero.

The normalization process is as follows. For a given tag, we use $\{\phi_1, \ldots, \phi_n\}$ to denote its phase readings, where n is the number of total readings. For the i-th reading ϕ_i $(1 \le i \le n)$, the normalized value is calculated as

$$\widetilde{\phi_i} = \begin{cases} \frac{\phi_i - \bar{\phi}}{\phi_{max} - \bar{\phi}} & , \quad \phi_i \ge 0 \\ \frac{\phi_i - \bar{\phi}}{-\phi_{min}} & , \quad \phi_i < 0 \end{cases}, \tag{8}$$

where $\bar{\phi}$, ϕ_{max}, and ϕ_{min} are the mean value, maximum value, and minimum value of all the phase readings of this tag, respectively. Fig. 4d shows the data after normalization. Compared with Fig. 4c, it is apparent the signal changes caused by gestures are magnified and the background signals are suppressed.

4.2 Gesture Segmentation

After obtaining the normalized data, the next step is to extract signal segments related to gestures. This is a challenging task [17]. If we use a loose boundary, some background signals (e.g., those signals whose values are around zero in Fig. 4d) might be included in the resulted gesture segments. In contrast, if we use a very stringent boundary, then some signals related to the gesture might be excluded from the resulted gesture segments. In both cases, the gesture recognition accuracy might be degraded. Fortunately, our approach mainly uses statistics of the signals to perform gesture recognition and thus is more tolerant to gesture segmentation errors than DTW-based approaches [17], [18].

The gesture segmentation stage consists of three steps. First, we obtain the boundaries of signals for individual tags. Second, we combine the boundaries of different tags to obtain the boundary of the gesture. Third, there might be some exceptional cases in which the boundaries of some specific tags are apart from the boundaries of the other tags, and we should handle such exceptions to avoid segmentation errors.

4.2.1 Boundary Detection for Individual Tags

We employ a modified Varri method [49] to get boundaries of signals for individual tags. The method uses a sliding window that combines amplitude measurement and frequency measurement of the signal. Denote by L the length of the sliding window. The amplitude measurement and the frequency measurement of the i-th window is calculated as

$$\mathcal{A}_i = \sum_{k=1}^{L} |\phi_{i,k}| \tag{9}$$

and

$$\mathcal{F}_i = \sum_{k=1}^{L} |\phi_{i,k} - \phi_{i,k-1}|, \tag{10}$$

where $\phi_{i,k}$ denotes the k-th data point in the i-th sliding window. The measurement difference function $\mathcal{G}$ is defined as

$$\mathcal{G}(i) = \mathcal{C}_{\mathcal{A}}|\mathcal{A}_{i+1} - \mathcal{A}_i| + \mathcal{C}_{\mathcal{F}}|\mathcal{F}_{i+1} - \mathcal{F}_i|, \tag{11}$$

where $\mathcal{C}_{\mathcal{A}}$ and $\mathcal{C}_{\mathcal{F}}$ are two application-dependent coefficients, whose values are experimentally set as $\mathcal{C}_{\mathcal{A}} = 7$ and $\mathcal{C}_{\mathcal{F}} = 1$. The local maxima (above a predefined threshold) of the $\mathcal{G}$ function indicates the boundaries of gesture segments in the signal [49].

The length of the sliding window, namely L, affects both the calculation efficiency of $\mathcal{G}$ and the segmentation accuracy. Different from [17] that adopts a genetic algorithm to dynamically determine the value of L, in ReActor+ we use a fixed value of L. The reason that we can use a fixed L is that our approach uses statistics rather than detailed data points, and thus it can tolerant slight boundary shifts caused by different L. Moreover, by using a fixed L we can avoid searching for different L with time-consuming genetic algorithms. The value of L is experimentally set as following. Denoting by T_D the duration of a gesture and by R the mean sampling rate of tags, we set $L = T_D * R$. In our experiments, the signal fluctuation duration caused by a gesture is usually less than 1.5 seconds, and the mean sampling rate for each tag is around 40 readings per second. Thus we set $L = 60$. Experimental results show that this setting can ensure that all signal fluctuations caused by gestures can be captured while only a few background signals would be included in the gesture segmentation.

Fig. 6a shows the boundaries detected for four tags. It can be observed that for each tag the signals between corresponding boundaries capture the signal fluctuation caused by the gesture. However, because the reader interrogates different tags at different times, the boundaries of different tags are not aligned. We should fuse the boundaries of different tags to get a unified segment for the gesture.

4.2.2 Gesture Boundary Detection

After the boundaries of individual tag signals are obtained, we get the boundary of the gesture as follows. Denote by m the number of tags, and denote by $B_{l,j}$ and $B_{r,j}$ the left and right boundary of the j-th ($1 \leq j \leq m$) tag, respectively. Then the boundary of the gesture is calculated as

$$BG_l = \min\{B_{l,j}\} \text{ and } BG_r = \max\{B_{r,j}\},\ 1 \leq j \leq m. \tag{12}$$

Fig. 6b shows the obtained boundaries of the gesture. It can be observed that the fluctuated signals caused by the gesture are well included in the obtained gesture segmentation. Meanwhile, most of the background signals are excluded from the obtained gesture segmentation.

4.2.3 Handling Exceptional Cases

In some exceptional cases, the boundaries of some specific tags might be apart from the boundaries of the other tags. For example, it might be the case that one tag is obstructed by the user's arm when she performs the gesture. In such cases, the sampling rate of this tag would be very low and it cannot correctly reflect the signal changes caused by the gesture. For example, as shown in Fig. 6c, the boundaries of the *bottom right* tag are distant from the boundaries of the other tags. If we still use Eq. (12) to determine the boundary of the gesture in such cases, the obtained gesture segments might contain a lot of background signals.

We propose a threshold-based approach to handle such exceptional cases. We calculate the distances between the most left boundary and the second most left boundary among all tags. If the distance is larger than a threshold, we use the second most left boundary as the starting point of the gesture segment. Similarly, if the distance between the most right boundary and the second most right boundary is larger than a threshold, we use the second-most right boundary as the ending point of the gesture segment. Denote by

$$BG'_l = \min\{\{B_{l,j}\} \setminus \{BG_l\}\} \tag{13}$$

and

$$BG'_r = \max\{\{B_{r,j}\} \setminus \{BG_r\}\}. \tag{14}$$

If $BG_l - BG'_l > \delta_l$, we set BG'_l as the starting point of the gesture segment. Similarly, if $BG_r - BG'_r > \delta_r$, we use BG'_r as the ending point of the gesture segment. In other cases, we still use BG_l and BG_r calculated by Eq. (12) as boundaries of the gesture segment. We experimentally set $\delta_l = -1$ and $\delta_r = 1$, which indicates that the four boundaries of time offset values are less than 1s.

Fig. 6d shows the gesture segmentation obtained with our exception handling approach. It can be observed that the effect of the misaligned tag readings is effectively mitigated.

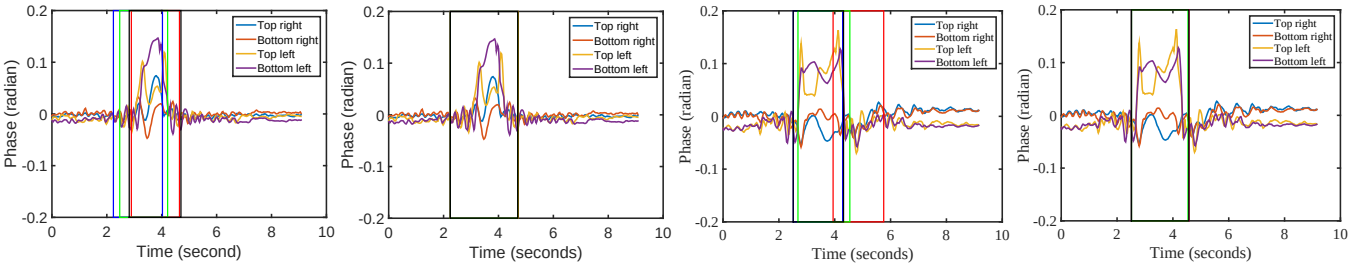
4.3 Attributes Extraction

We extract two types of attributes for each gesture segment: coarse-grained statistical attributes that characterize global features of the segment profile and fine-grained wavelet decomposition coefficient attributes that can preserve local features of the segment profile.

4.3.1 Statistical Attributes

The statistical attributes can characterize the global profile feature of gesture segments. We consider three kinds of statistical attributes:

- Attributes that reflect the central tendency of the data in the gesture segment, including the *mode*, the *median*, the *first quartile*, the *third quartile*, and the *arithmetic mean*.



(a) Normal case (b) Segment in normal case (c) Abnormal case (d) Segment in abnormal case

Fig. 6: Gesture segmentation in ReActor: (a) boundaries of individual tags in a normal case; (b) boundary of the gesture in a normal case; (c) boundaries of individual tags in an exceptional case; (d) boundary of the gesture in an exceptional case.

- Attributes that reflect the dispersion of the data, including the *max*, the *min*, the *range*, the *variance*, the *standard deviation*, and the *third-order central moment*.
- Attributes that reflect the distribution shape of data, including the *kurtosis* and the *skewness*.

For each tag in each gesture segment, we can calculate the above 13 attributes. When there are multiple tags, we concatenate attributes from different tags to form the attribute set for the gesture. We then use the collected data to train a Random Forest classifier and use it to recognize 18 different gestures shown in Fig. 12: *knock* (KN), *up* (UP), *down* (DN), *left* (LF), *right* (RG), *zoom in* (ZI), *zoom out* (ZO), *push* (PH), *pull* (PL), *circle clockwise* (CC), *circle anticlockwise* (CA), *left-right* (LR), *right-left* (RL), *up-down* (UD), *down-up* (DU), *knock twice* (KT), *enlarge* (EN), and *shrink* (SH). The normalized confusion matrix of classifying different gestures with 4 tags is shown in Fig. 7. We can observe that the recognition accuracy of 6 gestures (UP, DN, LF, DU, KT and SH) can reach 1, but the recognition accuracy of the other gestures are all lower than 0.9. The recognition accuracy of CA (0.40) and LR (0.50) are the lowest among all the gestures. The reason is that the signal profiles of these gestures have rich local features that cannot be fully characterized by the statistical attributes. The overall classification accuracy in Fig. 7 is 0.85.

4.3.2 Wavelet Decomposition Coefficient Attributes

Using mere statistical attributes, ReActor+ cannot accurately recognize gestures like CA and LR whose signal profiles have rich local features. To capture the fine-grained local features of gesture profiles, we apply wavelet decomposition to the signal profile of gestures and use the wavelet coefficients as attributes to distinguish different gestures.

***Wavelet coefficient calculation*:** We use the Daubechies wavelet (dbN) [55] as the wavelet base to decompose the signal profile of each gesture gesture. Note that similar as in extracting statistical attributes, we decompose signals related to different tags and concatenate attributes of different tags. We use discrete wavelet transformation to reduce the redundancy of wavelet transform coefficients:

$$\psi_{a,\tau}(t) = \frac{1}{\sqrt{a}}\psi(\frac{t-\tau}{a}), (a \neq 0, b \in R) \tag{15}$$

where $a = a_0^j$ ($a_0 > 1$, $j \in Z$) are discretized scale parameters and $b = ka_0^j b_0$ ($b_0 > 0$, $k \in Z$) are discretized translation parameters. Both high frequency coefficients and low frequency coefficients are used as attributes in ReActor+ to perform gesture recognition.

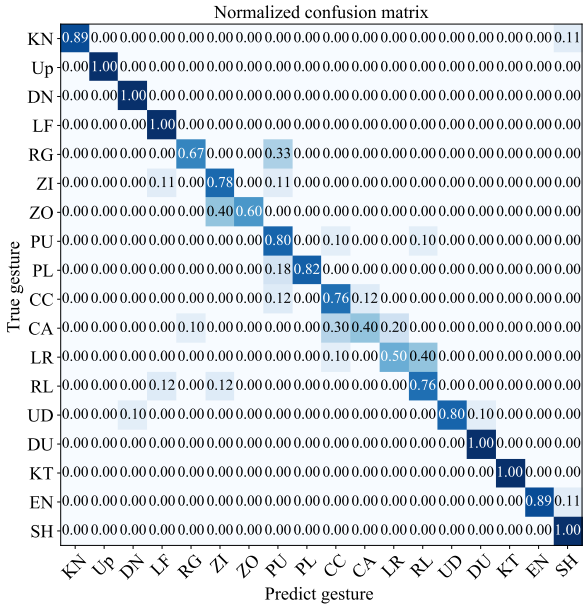


Fig. 7: Normalized confusion matrix when using only statistical features for gesture recognition.

***Data interpolation*:** The number of wavelet coefficients of a signal sequence is related to the number of data in the sequence. Due to the randomness in the communications between the reader and tags, the number of phase readings for different tags might be different. Thus, before calculating the wavelet coefficients of signal sequences of different tags, we interpolate the signal of all the tags to make in the same length. Denote the time related to the starting boundary of the gesture by T_s, and denote the time related to the ending boundary of the gesture by T_e. We create a series of time points T_i ($0 \leq i \leq 100$) as

$$T_i = T_s + i * \Delta T, \ , \Delta T = \frac{T_e - T_s}{100}. \tag{16}$$

We use linear interpolation to calculate the phase values on these time points, and use the interpolated data to calculate the wavelet coefficients.

After obtaining the wavelet coefficients, we combine both statistical attributes and wavelet coefficient attributes as input to perform gesture recognition. The level of wavelet decomposition has slight impact on the recognition accuracy. Fig. 9 shows the recognition accuracy when different

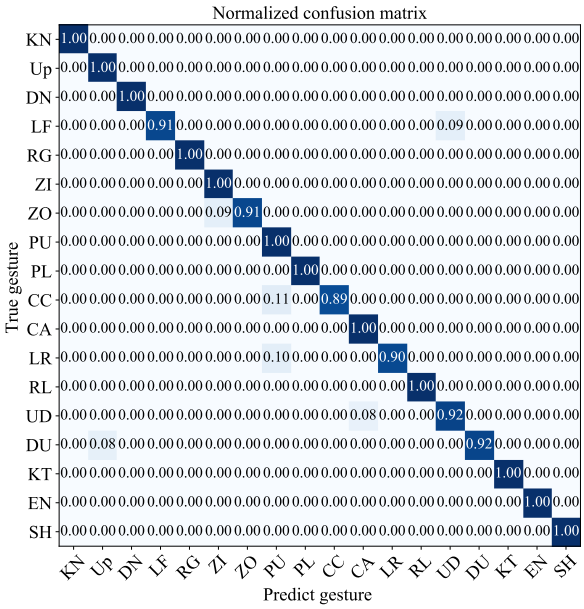


Fig. 8: Normalized confusion matrix when using both statistics attributes and wavelet decomposition coefficients for gesture recognition.

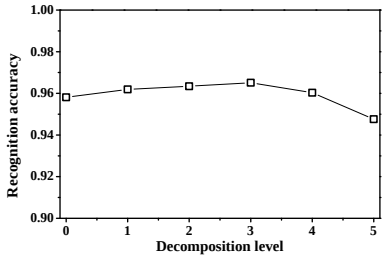


Fig. 9: Gesture recognition accuracy with different level of wavelet decomposition.

level of decomposition is used. It can be observed that the accuracy is the highest when level 3 decomposition is used, and thus we use coefficients related to level 3 decomposition in the rest of this paper. The normalized confusion matrix of using the combined attribute set to perform gesture recognition is shown in Fig. 8. We can observe that the recognition accuracy of most gestures are higher than 0.90 except CC, which is 0.89.

4.4 Handling Varing Gesture Speed

In practical scenarios, the same gesture might be performed with different speeds, either for the same user or for different users. When the user performs gestures with a speed different from the training gesture, the recognition accuray significantly degrades. We conducted some experiments to validate this speculation. In the experiment, the user performs the same gesture with three different speeds: (a) *normal* speed, for which the user performs one gesture with about 1∼2 seconds; (b) *slow* speed, for which the user performs one gesture with about 1.5X time of the normal speed; (c) *fast* speed, for which the user completes the gesture with about 0.5X time of the normal speed. TABLE 1 shows the performance of ReActor when the training speed and the test speed are different. It can be observed that the accuracy degrades significantly when the training speed does not match with the testing speed. The recognition accuracy is close to random guessing when the testing speed does not match the training speed.

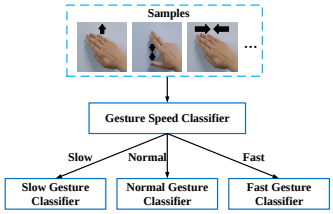
TABLE 1: Accuracy of ReActor/GRfid when using different speeds in training and testing.

	Train-Slow	Train-Nomal	Train-Fast
Test-Slow	0.913/0.811	0.212/0.367	0.146/0.144
Test-Nomal	0.247/0.383	0.938/0.833	0.312/0.294
Test-Fast	0.146/0.172	0.257/0.294	0.836/0.656

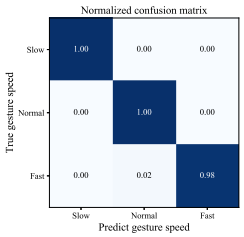
We propose a two-stage approach to address this problem. First, we train different gesture classifier for different speeds. In our case, we train three different classifiers by using samples collected at three different speeds, namely *slow*, *normal*, and *fast*. As shown in Fig. 10a, before feeding the extracted features to the gesture classifier, we first train a speed classifier to predict the speed of the gesture. Besides the statistical features (Section 4.3.1), we also use the duration of the gesture as one additional feature to train the speed classifier, which is calculated as

$$L = BG_r - BG_l, \tag{17}$$

where BG_r and BG_l are the time point of the right boundary and the left boundary of the gesture segment, respectively. We train the speed classifier with the RandomForest algorithm, and the accuracy in recognizing different speed of the gesture is higher than 0.99, as shown in Fig. 10b.



(a)



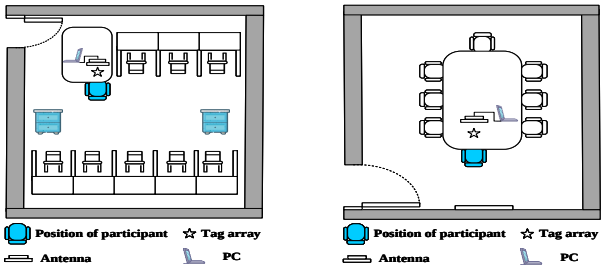
(b)

Fig. 10: (a) Diagram of handling gesture speed; (b) Normalized confusion matrix of different gesture speed recognition.

4.5 Classification Model Selection

We test 8 classification models, including one CNN-based model as used in [18] and 7 traditional machine learning approaches, namely RandomForest, 1-NN, J48, Random Tree, Naïve Bayesian, Logistic, and KStar. Note that for the CNN model, we use the raw signal as input. For other models, we use the combined feature set proposed in this paper as input. For the sake of fairness, we adopt the structure of network and use default parameter settings used in [17].

The accuracy and latency of different models are listed in TABLE 2. It can be observed that RandomForest achieves the highest accuracy (0.965) and its latency is also small. Some models, e.g. Logistic, achieves comparative accuracy as RandomForest but are much slower. The CNN-based



(a) (b)

Fig. 11: (a) Plane figure of laboratory. (b) Plane figure of meeting room.

method is slower than RandomForest and the accuracy is also lower (0.878 vs 0.965). Thus, in this paper we select the RandomForest approach as the default classification model.

TABLE 2: Performance of different classification models.

Classification Models	Accuracy	Latency
RandomForest	**0.965**	**Small**
1-NN	0.946	Small
J48	0.870	Small
Random Tree	0.819	Small
Naive Bayesian	0.813	Small
Logistic	0.954	Large
KStar	0.884	Large
CNN [18]	0.878	Small

5 Performance Evaluation

In this section, we evaluate the performance of ReActor and compare it with the state-of-the-art approaches. We implement ReActor/ReActor+ with Matlab and run the software on a personal computer with Intel(R) Core(TM) i5-7500 CPU @ 3.40 GHz and 8G memory. The reader collects phase values with the antenna transmitting power set at 20dBm. The frequency is set at 920.625MHz and the default mode of the reader is set to the *MaxThroughput* mode. When implementing GRfid for comparison, we use 10 profile templates for each gesture. We use the function implementation of DTW provided by Matlab to implement GRfid.

In the experiments, we consider 18 different gestures (as shown in Fig. 12) and evaluate the performance in two environments as shown in Fig. 11. Ten users are invited to participate in the experiments. The users perform each gestures 20 times based on their own interpretation of gestures without guidelines from the authors. For each experiment, we randomly select 70% of samples to train the model and use the rest 30% of samples to test the model. The results are averaged over 10 independent experiments.

5.1 Experiment Setting

The hardware setting is shown in Fig. 1. We paste the tags to a transparent cover of a 53cm*39cm*32cm plastic box and put the antenna at the bottom of the box. The user performs different gestures above the plastic cover, and a laptop is used to control one Impinj R420 reader to send commands to the antenna and record the RSS values and phase values received at the antennal.

When evaluating the performance of different approaches, we consider the following factors:

- *Number of tags*: We attach a different number of tags (3∼8) to the cover to investigate how the number of tags affects the accuracy of different approaches. The tags are evenly deployed on the cover. In the default setting, we attach four tags to the cover.
- *Gesture speed:* We consider three different gesture speeds, *normal*, *slow*, and *fast*. The duration of a normal gesture is about 1∼2 seconds, while the duration of a fast/slow gesture is about 0.5X shorter/longer than the normal speed case.
- *Operational distance:* We test the performance of different approaches when the distance between the user's hand and the tags changes. We consider 6 distances, from 20cm to 220cm stepped by 40cm. The default distance is 20cm.
- *Number of readings:* When there are multiple users in the environment, the number of readings for each tag would be reduced due to collisions among tags. We investigate the impact of the number of readings that can be obtained for each tag on the accuracy of different approaches.

Besides these factors, we also investigate the performance of ReActor/ReActor+ when it is used across different environments and the impact of interference caused by nearby moving persons.

5.2 Impact of Tag Number

The recognition accuracy of different approaches with different number of tags is plotted in Fig. 13. It can be observed that both ReActor and ReActor+ achieve higher accuracy when more tags are used. ReActor+ achieves higher accuracy than ReActor because it eliminates interference reflection signals from static objects in the environment, but the performance gap becomes smaller when the number of tags increases. ReActor+ always achieves a recognition accuracy higher than 0.95 in all the cases. In contrast, the recognition of ReActor is lower than 0.95 when the tag number is smaller than 4. When the number of tags increases, the accuracy of GRfid first improves and then slightly drops. It achieves slightly higher accuracy than ReActor when there are less than four tags, but performs worse than ReActor when there are more tags. The reasons might be as follows. When there are more tags, ReActor can better tolerate noises in signals because it uses a machine-learning-based algorithm to build the classifier. In contrast, GRfid uses a template-matching method to find the best matching template and it is affected more by the noise readings. To summarize, all the three approaches achieve recognition accuracy higher than 0.9 in most cases, but ReActor+ consistently outperforms ReActor and GRfid.

5.3 Recognition Latency

The time complexity of the basic DTW algorithm is $O(M * N)$, where M and N are the length of the two data series for which the DTW distance is calculated. To make fair comparisons with GRfid, besides the basic DTW, we also implemented an improved version of DTW named FastDTW

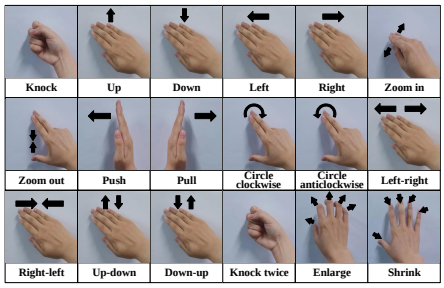


Fig. 12: The 18 gestures used in evaluation.

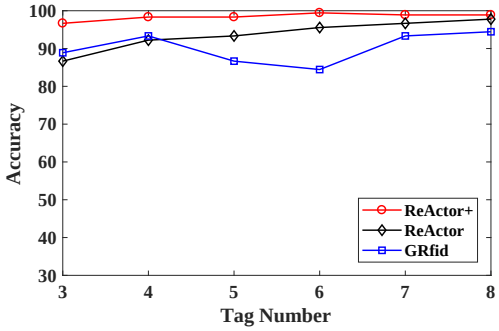


Fig. 13: Gesture recognition accuracy with different number of tags.

[56], whose time complexity is reduced to $O(max(N, M))$ and find the the optimal solution in most cases but without guarantee. Fig. 14 plots the recognition latency of ReActor/ReActor+ and GRfid using DTW and FastDTW, respectively. We select 90 gesture instances, 5 instances for gesture listed in Fig. 12. Here we define the recognition latency as the time needed to output the result after the gesture are performed and the corresponding data are collected, which include both the time spent in signal processing and the time spent in gesture matching (for GRfid) or gesture recognition (for ReActor/ReActor+). For GRfid, we construct 10 profile templates for each gesture.

As shown in Fig. 14, the recognition latency of GRfid using DTW and FastDTW is usually longer than 65 and 25 seconds respectively, while the recognition latency of ReActor is smaller than 100 ms in most cases. Compared with GRfid, ReActor reduces recognition latency by two orders of magnitude. The average recognition latency in ReActor is only 72.2 ms, while the average recognition latency of GRfid using DTW and FastDTW is 70.81 seconds and 25.94 seconds respectively. Thus, ReActor can support applications that require real-time gesture recognition. Note that time spent in removing reflection signals in ReActor+ is negligible because it uses only simple calculations.

Here we give an analysis on the runtime complexity of ReActor and GRfid. Denote by N the length of each segment profile (assume that all the segment profiles are of the same length), by K the number of templates for each gesture in GRfid, and by P the number of gestures. For GRfid, it needs to compare the testing segment profile to all the templates of all the gestures, thus its complexity is $O(P * K * N^2)$. When fastDTW [56] is used, the time complexity is $O(P * K * N)$. The runtime of ReActor is dominated by the time spent in extracting features from the raw signals.[1] Both the statistical features and the wavelet coefficients can be calculated in $O(N)$ time [57]. Thus, the time complexity of ReActor is much lower than approaches based on DTW or its variants. For example, if we consider 18 gestures and for each gesture we use 10 templates, then in GRfid each segment need to compare with 18*10=180 templates. The time spent in each comparison is on the same magnitude as the time spent in ReActor. We should clarify that there might be many parameters in RandomForest, however these parameters need to be trained once, which does not affect the latency that occurs in the testing phase.

We also investigate the latency incurred in different stages of ReActor and plot corresponding values in Fig. 15. It can be observed that the most time-consuming stages in ReActor are gesture segmentation and wavelet decomposition, both of which take about 20ms. In contrast, each DTW-based template matching in GRfid takes more than 100ms and it needs to perform 180 comparisons in our experiment setting. Thus, the execution time of ReActor is much shorter than approaches based on template matching.

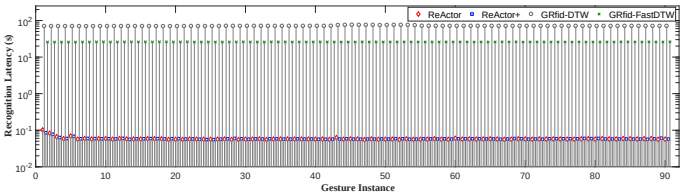


Fig. 14: Recognition latency of ReActor, ReActor+, and GRfid for 90 gesture instances, 5 instances for each gesture.

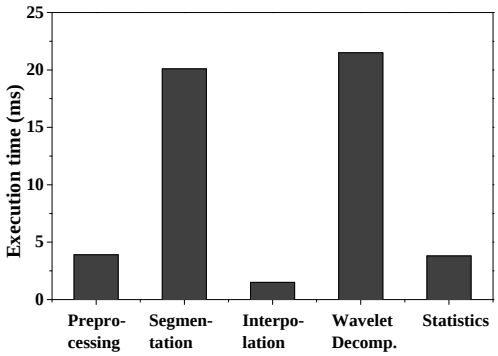


Fig. 15: Execution time in different stage of ReActor/ReActor+.

5.4 Effectiveness of Features

TABLE 3: Accuracy with different features.

	ReActor+	ReActor	CNN
Accuracy	0.947	0.928	0.878

1. The runtime complexity of RandomForest is only related to the depth of the tree. Actually, RandomForest is usually considered more time-efficient than other machine learning algorithms.

To evaluate the effectiveness of feature sets proposed in this paper, we compare the features extracted in ReActor with the features extracted with convolutional neural networks (CNN) as in [18]. The results are listed in TABLE 3. The results show that with the features extracted from the CNN model, the accuracy is 0.878, which is about 5 percent lower than the accuracy of ReActor. Because the interference from surrounding objects are mitigated, ReActor+ further improves the accuracy by 2 percent. The results indicate that the features extracted in ReActor/ReActor+ are more effective than the features extracted by CNN.

5.5 Impact of Gesture Speed

The recognition accuracy of ReActor, ReActor+, and GRfid when the gesture speed changes are shown in Fig. 16. Note that for fair comparison, we collect templates for three different speeds for GRfid and compare to all the templates at different speeds when finding the most matched template. This significantly improves the accuracy of GRfid. GRfid's accuracy when only normal speed templates are used is listed in TABLE 1.

ReActor/ReActor+ achieves higher accuracy than GRfid when the speed of the gesture is not known a prior because it uses a speed classifier to determine the speed of the gesture before feeding it to the proper classifier. We randomly select 270 instances at different speeds and use ReActor+ to classify them. Because the speed classifier can judge the speed of gestures with very high accuracy, the recognition accuracy is 0.93, 0.95, and 0.90 for gestures at slow speed, normal speed, and fast speed, respectively. In contrast, if we mix samples collected at different speeds and use the mixed samples to train a classifier, the recognition accuracy is lower than 0.8. GRfid shows high tolerance to the varying speed of the gesture because DTW can handle the profile deformation to some extent. However, after applying the two-stage approach, ReActor achieves higher accuracy than GRfid, especially for the slow gestures and fast gestures, with an improvement factor of about 5 percent.

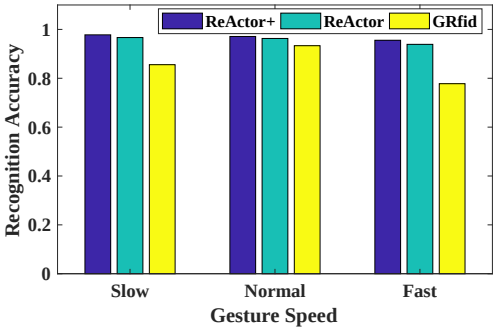


Fig. 16: Recognition accuracy with different gesture speeds.

5.6 Impact of Interference

We test the resistance of ReActor and ReActor+ to interferences from other moving people when the user performs gestures. To generate interference, we let several people walk behind the antenna during the experiment. The recognition accuracy of the three approaches is show in Fig. 17. The recognition accuracy of ReActor and ReActor+ is insensitve to interferences from moving persons: its accuracy remains higher than 0.95 even when four people move around the user during the gesture operation. ReActor+ performs slightly better than ReActor because it can remove interference signals from static objects in the environment. In contrast, the recognition accuracy of GRfid significantly degrades when the number of interfering person increases. When there are 4 moving people, the recognition accuracy of GRfid drops to 0.63. These results are consistent with the results reported in [17].

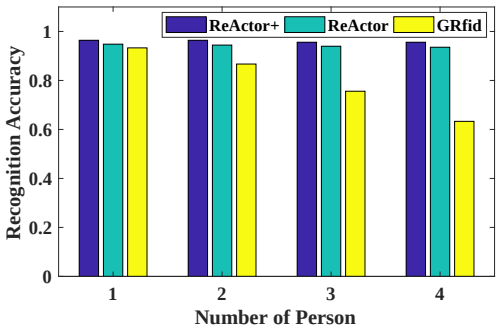


Fig. 17: Impact of moving person.

5.7 Impact of Operational Distance

The performance of ReActor and GRfid at different operational distances are plotted in Fig. 18. The operational distance is the distance between the user's hand and the plastic cover. It can be observed that from 20cm to 220cm, ReActor+'s recognition accuracy is higher than GRfid, especially at 140cm, at which distance the performance gap achieves 18 percent. The recognition accuracy of ReActor degrades only slightly as the distance increases, but the accuracy of GRfid degrades significantly. The reason might be as follows. When the distance between the user's hand and the tags increases, the pattern changes caused by the user's gesture become insignificant. In this case, it might be not easy for GRfid to find the proper matching template. In contrast, because ReActor uses statistical features, the tendency of the profile in both time-domain and frequency-domain is less impacted by the distance. When the distance increases, compared with ReActor, the accuracy of ReActor+ remains more stable.

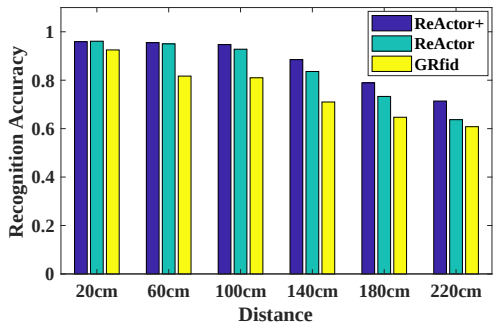


Fig. 18: Recognition accuracy of ReActor/ReActor+ and GRfid at different operational distances.

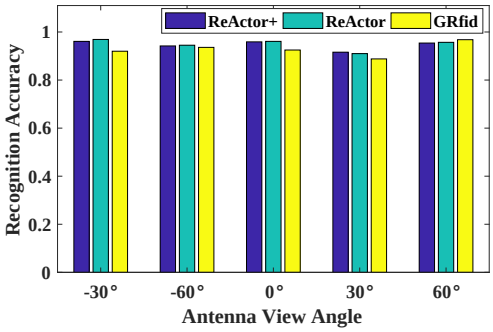


Fig. 19: Recognition accuracy of ReActor and GRfid at different antenna view angles.

5.8 Impact of Antenna View Angle

We investigate the performance of different approaches with various antenna view angles. The antenna view angle indicates the angle between the antenna and the horizontal axis. Specifically, we let users perform gestures with 5 different antenna view angles, including $-\pi/6$, $-\pi/3$, 0, $\pi/6$, $\pi/3$, and evaluate the performance of different approaches at each angle. The results are plotted in Fig. 19. It can be observed that when the angle is $\pi/6$ or $-\pi/6$, the accuracy of the three approaches slightly decreases. When the angle is $\pi/3$e or $-\pi/3$, the accuracy is nearly the same as when the user is in front of the antenna. These results show that different angles have only very limited impact on accuracy. The reason might be that we use omni-directional circularly polarized antennas in our experiments, for which the tags can be interrogated in a large region, and thus different angles will not impact the signal profiles significantly.

5.9 Impact of Data Volume

When there are multiple users, the number of RSS/phase readings of a user will be reduced because RFID readers use a time-division method to collect data from tags. We evaluated the performance of ReActor when the data volume decreases and plot the result in Fig. 20. In our default setting, the reader can interrogate tags at about 400 readings per second, e.g., 100 readings per second for each tag. The accuracy corresponding to $1/n$ in the plot means that only $1/n$ data randomly selected from all the collected data are used in recognizing the gestures, which mimics the case in which n users simultaneously perform the gesture. It can be observed that for both ReActor and GRfid, the recognition accuracy degrades when fewer data are used, but the performance of ReActor is more stable than GRfid. The accuracy of GRfid drops to below 0.85 when only $1/3$ or fewer data are used. In contrast, the accuracy of ReActor and ReActor+ remains higher than 0.9 even when only $1/5$ data are used. The reason is that when the number of data points in the gesture segmentation decreases, it is more difficult for DTW-based algorithms to find the correct template. In contrast, ReActor and ReActor+ use statistical features that are more resistant to the number of data points. It worth noting that ReActor performs slightly better than ReActor+ when fewer data are used. It might be because when fewer data are available, the reflection signals cannot be correctly removed, which slightly decreases the accuracy of ReActor+.

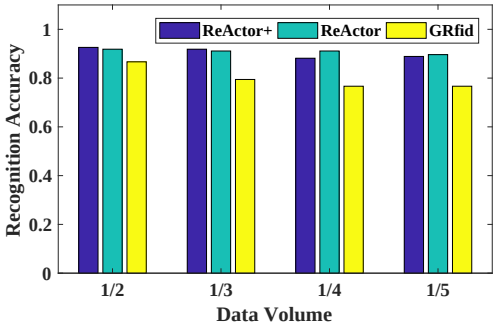


Fig. 20: Recognition accuracy of different approaches when the data volume are reduced.

5.10 Performance of Different Users

We also evaluate the accuracy of different approaches for different users. We recruit 10 users to test the accuracy for different individuals, and plot the results in Fig. 21. The accuracy for different users is different, showing up to about 10 percent difference for different users (user 4 vs. user 8). For 9 users out of the 10 users, ReActor+ significantly outperforms GRfid. Only for user 5 and user 7, GRfid achieves very slightly higher accuracy than ReActor+.

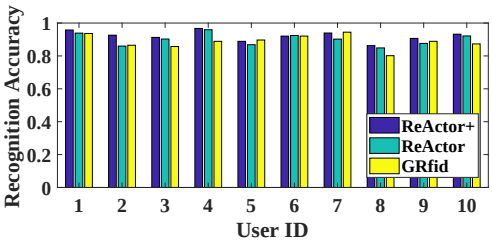


Fig. 21: Recognition accuracy of ReActor/ReActor+ and GRfid for different users.

5.11 Cross-Environment Performance

We evaluate the cross-environment performance of different approaches by training the classifier with data collected in one environment and test the trained classifier in another environment. To this end, we collect two datasets from two different rooms, which are denoted as $s1$ and $s2$ in Fig. 11, and train the model using one dataset and test the model using another dataset. The results are plotted in Fig. 22. The notation $A-B$ in the figure means we train the classifier with data collected in environment A and test it in environment B, where $A, B \in \{s1, s2\}$.

It can be observed that when the training environment and the testing environment are the same, all the three approaches show fairly high recognition accuracy. However, when the training environment and testing environment are different, the accuracy of ReActor and GRfid significantly drops to below 0.45 and 0.24, respectively. The reason is that in both approaches the noise signals caused by static objects in the environments are encoded in the classifier. In contrast,

ReActor+ removes the interference signals reflected from static objects in the environment and thus effectively mitigates the negative effects of background signals. The cross-environment accuracy of ReActor+ remains higher than 0.75 in both $s1-s2$ and $s2-s1$ cases.

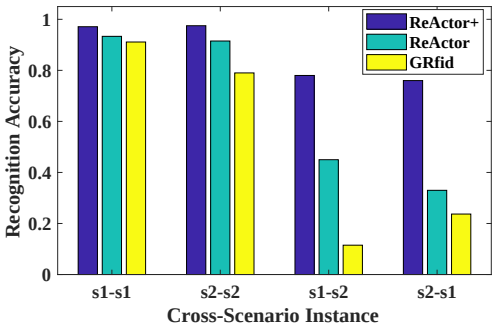


Fig. 22: Accuracy of different approaches when used across different environments.

6 Conclusion

In this paper, we propose a real-time and accurate hand gesture recognition system called ReActor. Compared with state-of-the-art approaches, ReActor reduces recognition latency by two orders of magnitude and achieves similar or even higher recognition accuracy. With the recognition latency lower than 100 ms, ReActor is able to support real-time applications. We further extend ReActor from two aspects. First, we remove reflection signals from surrounding static objects in the environment, which boosts its performance when used in cross-environment cases. Second, we propose a two-stage approach to handle the impact of varying speeds, which improves the accuracy when the speed of the gesture is not known prior. However, there are still spaces to improve ReActor, especially when there are multiple users simultaneously perform the gesture in the environment, which will make the data very sparse and cause inter-tag interference. We consider addressing these problems as our future work.

Acknowledgement

This work is partially supported by the National Natural Science Foundation of China (Grant Nos. 61772559, 62172154? 62177047, 61872310 , 62272486), the Hunan Provincial Natural Science Foundation of China under grant No. 2020JJ3016, the Open Foundation of Henan Key Laboratory of Cyberspace Situation Awareness (No. HNTS2022024), and Shenzhen Science and Technology Innovation Commission (JCYJ20200109142008673). Prof. Xiaoyan Kui and Prof. Xuan Liu are the corresponding authors of this paper.

References

[1] Shigeng Zhang, Xuan Liu, Yangyang Liu, Bo Ding, Song Guo, and Jianxin Wang. Accurate respiration monitoring for mobile users with commercial RFID devices. *IEEE Journal of Selected Areas in Communications*, 39(2):513–525, 2021.

[2] Fusang Zhang, Zhi Wang, Beihong Jin, Jie Xiong, and Daqing Zhang. Your smart speaker can "hear" your heartbeat! *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)*, 4(4), 12 2020.

[3] Ju Wang, Jianyan Li, Mohammad Hossein Mazaheri, Keiko Katsuragawa, Daniel Vogel, and Omid Abari. Sensing finger input using an RFID transmission line. In Jin Nakazawa and Polly Huang, editors, *Proceedings of the 18th ACM Conference on Embedded Networked Sensor Systems (Sensys)*, pages 531–543. ACM, 2020.

[4] Lei Xie, Chuyu Wang, Yanling Bu, Jianqiang Sun, Qingliang Cai, Jie Wu, and Sanglu Lu. Taggedar: An RFID-based approach for recognition of multiple tagged objects in augmented reality systems. *IEEE Transactions on Mobile Computing*, 18(5):1188–1202, 2019.

[5] Chenshu Wu, Feng Zhang, Beibei Wang, and KJ Ray Liu. mSense: Towards mobile material sensing with a single millimeter-wave radio. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)*, 4(3):1–20, 2020.

[6] Aditya Virmani and Muhammad Shahzad. Position and orientation agnostic gesture recognition using wifi. In *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys)*, pages 252–264. ACM, 2017.

[7] Shigeng Zhang, Zijing Ma, Kaixuan Lu, Xuan Liu, Jia Liu, Song Guo, Albert Y. Zomaya, Jian Zhang, and Jianxin Wang. Hearme: Accurate and real-time lip reading based on commercial rfid devices. *IEEE Transactions on Mobile Computing*, pages 1–14, 2022.

[8] Hari Prabhat Gupta, Haresh S Chudgar, Siddhartha Mukherjee, Tanima Dutta, and Kulwant Sharma. A continuous hand gestures recognition technique for human-machine interaction using accelerometer and gyroscope sensors. *IEEE Sensors Journal*, 16(16):6425–6432, 2016.

[9] Jaime Lien, Nicholas Gillian, M Emre Karagozler, Patrick Amihood, Carsten Schwesig, Erik Olson, Hakim Raja, and Ivan Poupyrev. Soli: Ubiquitous gesture sensing with millimeter wave radar. *ACM Transactions on Graphics (TOG)*, 35(4):142, 2016.

[10] Sikun Lin, Hao Fei Cheng, Weikai Li, Zhanpeng Huang, Pan Hui, and Christoph Peylo. Ubii: Physical world interaction through augmented reality. *IEEE Transactions on Mobile Computing*, 16(3):872–885, 2017.

[11] Hao Wang, Daqing Zhang, Yasha Wang, Junyi Ma, Yuxiang Wang, and Shengjie Li. RT-Fall: A real-time and contactless fall detection system with commodity wifi devices. *IEEE Transactions on Mobile Computing*, 16(2):511–526, 2017.

[12] Xu Zhang, Xiang Chen, Yun Li, Vuokko Lantz, Kongqiao Wang, and Jihai Yang. A framework for hand gesture recognition based on accelerometer and emg sensors. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 41(6):1064–1076, 2011.

[13] Lei Jing, Zixue Cheng, Yinghui Zhou, Junbo Wang, and Tongjun Huang. Magic ring: a self-contained gesture input device on finger. In *Proceedings of the 12th International Conference on Mobile and Ubiquitous Multimedia (MUM)*, page 39:1=39:4. ACM, 2013.

[14] Liang Wang, Tao Gu, Xianping Tao, and Jian Lu. Toward a wearable RFID system for real-time activity recognition using radio patterns. *IEEE Transactions on Mobile Computing*, 16(1):228–242, 2017.

[15] Xiaojun Chang, Zhigang Ma, Ming Lin, Yi Yang, and Alexander G Hauptmann. Feature interaction augmented sparse learning for fast kinect motion detection. *IEEE Transactions on Image Processing*, 26(8):3911–3920, 2017.

[16] Raghav H Venkatnarayan, Griffin Page, and Muhammad Shahzad. Multi-user gesture recognition using wifi. In *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services (Mobisys)*, pages 401–413. ACM, 2018.

[17] Yongpan Zou, Jiang Xiao, Jinsong Han, Kaishun Wu, Yun Li, and Lionel M Ni. GRfid: A device-free RFID-based gesture recognition system. *IEEE Transactions on Mobile Computing*, 16(2):381–393, 2017.

[18] Chuyu Wang, Jian Liu, Yingying Chen, Hongbo Liu, Lei Xie, Wei Wang, Bingbing He, and Sanglu Lu. Multi-touch in the air: Device-free finger tracking and gesture recognition via COTS RFID. In *Proceedings of IEEE Conference on Computer Communications (Infocom)*, pages 1691–1699. IEEE, 2018.

[19] Yanwen Wang and Yuanqing Zheng. Modeling RFID signal reflection for contact-free activity recognition. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)*, 2(4):193:1–193:22, 2018.

[20] Wei Wang, Alex X Liu, and Ke Sun. Device-free gesture tracking using acoustic signals. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking (Mobicom)*, pages 82–94. ACM, 2016.

[21] Yanwen Wang, Jiaxing Shen, and Yuanqing Zheng. Push the limit of acoustic gesture recognition. In *Proceedings of IEEE Conference on Computer Communications (Infocom)*, pages 566–575. IEEE, 2020.

[22] Han Ding, Jinsong Han, Longfei Shangguan, Wei Xi, Zhiping Jiang, Zheng Yang, Zimu Zhou, Panlong Yang, and Jizhong Zhao. A platform for free-weight exercise monitoring with passive tags. *IEEE Transactions on Mobile Computing*, 16(12):3279–3293, 2017.

[23] Xinyu Li, Yanyi Zhang, Ivan Marsic, Aleksandra Sarcevic, and Randall S. Burd. Deep learning for RFID-based activity recognition. In *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems (SenSys)*, pages 164–175, 2016.

[24] Longfei Shangguan, Zimu Zhou, and Kyle Jamieson. Enabling gesture-based interactions with objects. In *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys)*, pages 239–251, 2017.

[25] Shigeng Zhang, Chengwei Yang, Xiaoyan Kui, Jianxin Wang, Xuan Liu, and Song Guo. Reactor: Real-time and accurate contactless gesture recognition with RFID. In *Proceedings of the 16th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, pages 1–9. IEEE, 2019.

[26] Zimu Zhou, Longfei Shangguan, Xiaolong Zheng, Lei Yang, and Yunhao Liu. Design and implementation of an RFID-Based customer shopping behavior mining system. *IEEE/ACM Transactions on Networking*, 25(4):2405–2418, 2017.

[27] Lei Yang, Qiongzheng Lin, Xiangyang Li, Tianci Liu, and Yunhao Liu. See through walls with COTS RFID system! In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking (Mobicom)*, pages 487–499, 2015.

[28] Jiayang Liu, Lin Zhong, Jehan Wickramasuriya, and Venu Vasudevan. uwave: Accelerometer-based personalized gesture recognition and its applications. *Pervasive and Mobile Computing*, 5(6):657–675, 2009.

[29] Zhou Ren, Junsong Yuan, Jingjing Meng, and Zhengyou Zhang. Robust part-based hand gesture recognition using kinect sensor. *IEEE Transactions on Multimedia*, 15(5):1110–1120, 2013.

[30] Jinsong Han, Han Ding, Chen Qian, Wei Xi, Zhi Wang, Zhiping Jiang, Longfei Shangguan, and Jizhong Zhao. Cbid: A customer behavior identification system using passive tags. *IEEE/ACM Transactions on Networking*, 24(5):2885–2898, 2016.

[31] Lei Xie, Jianqiang Sun, Qingliang Cai, Chuyu Wang, Jie Wu, and Sanglu Lu. Tell me what I see: Recognize RFID tagged objects in augmented reality systems. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing (Ubicomp)*, pages 916–927. ACM, 2016.

[32] Hanchuan Li, Can Ye, and Alanson P Sample. Idsense: A human object interaction detection system based on passive UHF RFID. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI)*, pages 2555–2564. ACM, 2015.

[33] Lei xie, Chuyu Wang, Alex X.Liu, Jianqiang Sun, and Sanglu Lu. Multi-touch in the air: Concurrent micromovement recognition using RF signals. *IEEE/ACM Transactions on Networking*, (1):231–244, 2017.

[34] Yanling Bu, Lei xie, Chuyu Wang, lei Yang, jia Liu, and Sanglu Lu. RF-Dial: An RFID-based 2d human-computer interaction via tag array. In *Proceedings of IEEE Conference on Computer Communications (Infocom)*, pages 837–845. IEEE, 2018.

[35] Jindong Wang, Yiqiang Chen, Shuji Hao, Xiaohui Peng, and Lisha Hu. Deep learning for sensor-based activity recognition: A survey. *Pattern Recognition Letters*, 2018.

[36] Yinggang Yu, Dong Wang, Run Zhao, and Qian Zhang. RFID based real-time recognition of ongoing gesture with adversarial learning. In *Proceedings of the 17th ACM Conference on Embedded Network Sensor Systems (SenSys)*, pages 298–310. ACM, 2019.

[37] Eshed Ohn-Bar and Mohan Manubhai Trivedi. Hand gesture recognition in real time for automotive interfaces: A multimodal vision-based approach and evaluations. *IEEE transactions on Intelligent Transportation Systems*, 15(6):2368–2377, 2014.

[38] Jordi Sanchez-Riera, Kathiravan Srinivasan, Kai-Lung Hua, Wen-Huang Cheng, M Anwar Hossain, and Mohammed F Alhamid. Robust rgb-d hand tracking using deep learning priors. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(9):2289–2301, 2018.

[39] Cheng Zhang, Qiuyue Xue, Anandghan Waghmare, Sumeet Jain, Yiming Pu, Sinan Hersek, Kent Lyons, Kenneth A Cunefare, Omer T Inan, and Gregory D Abowd. Soundtrak: Continuous 3d tracking of a finger using active acoustics. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)*, 1(2):30, 2017.

[40] Linfei Ge, Qian Zhang, Jin Zhang, and Qianyi Huang. Acoustic strength-based motion tracking. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)*, 4(4):1–19, 2020.

[41] Heba Abdelnasser, Moustafa Youssef, and Khaled A Harras. Wigest: A ubiquitous wifi-based gesture recognition system. In *Proceedings of IEEE Conference on Computer Communications (Infocom)*, pages 1472–1480. IEEE, 2015.

[42] Yan Wang, Jian Liu, Yingying Chen, Marco Gruteser, Jie Yang, and Hongbo Liu. E-eyes: device-free location-oriented activity identification using fine-grained wifi signatures. In *Proceedings of the 20th annual international conference on Mobile computing and networking (Mobicom)*, pages 617–628. ACM, 2014.

[43] Zhenghua Chen, Le Zhang, Chaoyang Jiang, Zhiguang Cao, and Wei Cui. Wifi CSI based passive human activity recognition using attention based BLSTM. *IEEE Transactions on Mobile Computing*, 18(11):2714–2724, 2019.

[44] Sheng Tan and Jie Yang. Wifinger: leveraging commodity wifi for fine-grained finger gesture recognition. In *Proceedings of the 17th ACM International Symposium on Mobile Ad Hoc Networking and Computing (Mobihoc)*, pages 201–210. ACM, 2016.

[45] Jie Zhang, Zhanyong Tang, Meng Li, Dingyi Fang, Petteri Nurmi, and Zheng Wang. Crosssense: Towards cross-site and large-scale wifi sensing. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking (Mobicom)*, pages 305–320, 2018.

[46] Yuanrun Fang, Biyun Sheng, Haiyan Wang, and Fu Xiao. Witransfer: A cross-scene transfer activity recognition system using wifi. In *Proceedings of the ACM Turing Celebration Conference-China*, pages 59–63, 2020.

[47] Unsoo Ha, Junshan Leng, Alaa Khaddaj, and Fadel Adib. Food and liquid sensing in practical environments using RFIDs. In *Proceedings of the 17th {USENIX} Symposium on Networked Systems Design and Implementation (NSDI)*, pages 1083–1100, 2020.

[48] Jie Wang, Yunong Zhao, Xiaorui Ma, Qinghua Gao, Miao Pan, and Hongyu Wang. Cross-scenario device-free activity recognition based on deep adversarial networks. *IEEE Transactions on Vehicular Technology*, 69(5):5416–5425, 2020.

[49] Hamed Azami, Karim Mohammadi, and Behzad Bozorgtabar. An improved signal segmentation using moving average and savitzky-golay filter. *Journal of Signal and Information Processing*, 3(01):39, 2012.

[50] Lei Yang, Yekui Chen, Xiang-Yang Li, Chaowei Xiao, Mo Li, and Yunhao Liu. Tagoram: Real-time tracking of mobile RFID tags to high precision using COTS devices. In *Proceedings of the 22th Annual International Conference on Mobile Computing and Networking (Mobicom)*, pages 237–248. ACM, 2014.

[51] Chao Zuo, Lei Huang, Minliang Zhang, Qian Chen, and Anand Asundi. Temporal phase unwrapping algorithms for fringe projection profilometry: A comparative review. *Optics and Lasers in Engineering*, 85:84–103, 2016.

[52] Daniel M Dobkin. *The rf in RFID: uhf RFID in practice*. Newnes, 2012.

[53] Ronald W. Schafer. What is a savitzky-golay filter?[lecture notes]. *IEEE Signal Processing Magazine*, 28(4):111–117, 2011.

[54] Yiman Wu and Liang Li. Sample normalization methods in quantitative metabolomics. *Journal of Chromatography A*, 1430:80–95, 2016.

[55] Hafeez Ullah Amin, Aamir Saeed Malik, Rana Fayyaz Ahmad, Nasreen Badruddin, Nidal Kamel, Muhammad Hussain, and Weng-Tink Chooi. Feature extraction and classification for eeg signals using wavelet transform and machine learning techniques. *Australasian physical & engineering sciences in medicine*, 38(1):139–149, 2015.

[56] Stan Salvador and Philip Chan. Toward accurate dynamic time warping in linear time and space. *Intelligent Data Analysis*, 11(5):561–580, 2007.

[57] Arrate Muñoz, Raphaël Ertlé, and Michael Unser. Continuous wavelet transform with arbitrary scales and o (n) complexity. *Signal Processing*, 82(5):749–757, 2002.

Shigeng Zhang Shigeng Zhang received the BSc, MSc, and DEng degrees, all in Computer Science, from Nanjing University, China, in 2004, 2007, and 2010, respectively. He is currently a Professor in School of Computer Science and Engineering at Central South University, China. His research interests include Internet of Things, mobile computing, RFID systems, and IoT security. He has published more than 70 technique papers in top international journals and conferences including Ubicomp, Infocom, Mobihoc, ICNP, TMC, TC, TPDS, TOSN, and JSAC. He is on the editorial board of International Journal of Distributed Sensor Networks, and was a program committee member of many international conferences including ICC, ICPADS, MASS, UIC and ISPA. He is a member of IEEE and ACM.

Zijing Ma Zijing Ma received the BSc degree in computer science and technology from South China Agricultural University in 2020. He is currently working towards his MSc degree in computer science and technology from Central South University, China. His research interests include wireless sensing, RFID, and the Internet of Things.

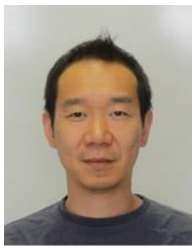
Chengwei Yang Chengwei Yang received the MSc degree in computer science and technology from Central South University in 2020. He is currently working at Research Institute of China Telecom Co., Ltd. His research interests include Cloud Computing, RFID, and the Internet of Things.

Xiaoyan Kui Xiaoyan Kui received her Ph.D. degree in Computer Science from Central South University, China, in 2012. She is currently a professor with the Department of Computer Science and Technology at Central South University. Her research interests include data visualization, medical big data, and medical image processing.

Xuan Liu Xuan Liu is currently a Professor in the College of Computer Science and Electronic Engineering at Hunan University. She received the BSc degree in information and computing mathematics from XiangTan University in 2005, the MSc degree in Computer Science from National University of Defense Technology in 2008, and the PhD degree in the Hong Kong Polytechnic University in 2015, respectively. Her research interests include Multi-agent reinforcement learning, RFID systems and Internet of Things. She has published more than 40 technique papers in top international journals including JSAC/ToN/TMC/TC/TPDS and top conferences including Infocom/ICNP/Mobihoc/Ubicomp.

Weiping Wang Weiping Wang received the Ph.D. degree in computer science from Central South University, China, in 2004. Currently, she is a professor in the School of Information Science and Engineering, Central South University. Her current research interests include network coding and network security. She has published more than 60 papers in refereed journals and conference proceedings.

Jianxin Wang Jianxin Wang received the BEng and MEng degrees in computer engineering from Central South University, China, in 1992 and 1996, respectively, and the PhD degree in computer science from Central South University, China, in2001. He is the Dean of and a professor in School of Computer Science and Engineering, Central South University, Changsha, Hunan, P.R. China. His current research interests include algorithm analysis and optimization, parameterized algorithm, Bioinformatics and computer network. He is a senior member of the IEEE.



Song Guo Song Guo is a Full Professor at Department of Computing, The Hong Kong Polytechnic University. He also holds a Changjiang Chair Professorship awarded by the Ministry of Education of China. Prof. Guo is a Fellow of the Canadian Academy of Engineering, Member of Academia Europaea, and Fellow of the IEEE (Computer Society). His research interests are mainly in federated learning, edge AI, mobile computing, and distributed systems. He published many papers in top venues with wide impact in these areas and was recognized as a Highly Cited Researcher (Clarivate Web of Science). He is the recipient of over a dozen Best Paper Awards from IEEE/ACM conferences, journals, and technical committees. Prof. Guo is the Editor-in-Chief of IEEE Open Journal of the Computer Society. He was an IEEE ComSoc Distinguished Lecturer and a member of IEEE ComSoc Board of Governors. He has served for IEEE Computer Society on Fellow Evaluation Committee, Transactions Operations Committee, Steering Committee of IEEE Transactions on Cloud Computing, Editor-in-Chief Search Committee, and been named on editorial board of a number of prestigious international journals like IEEE TC, IEEE TPDS, IEEE TCC, IEEE TETC, ACM CSUR, etc. He has also served as chairs of organizing and technical committees of many international conferences.